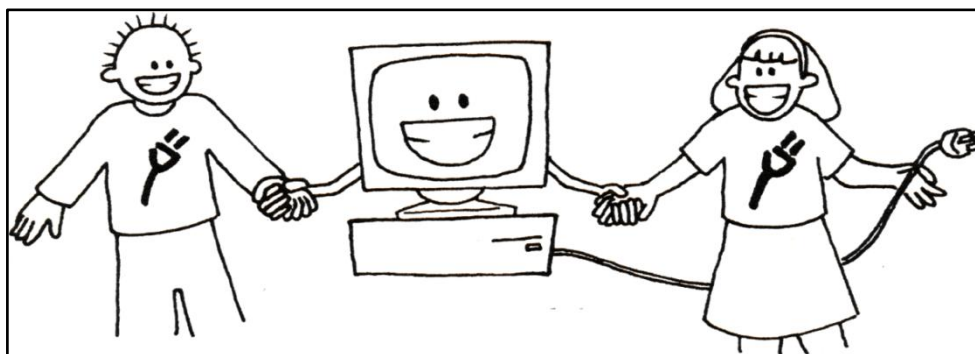


Informatika

BE KOMPIUTERIO



Turininga informatikos mokymosi medžiaga pradinukams ir vyresniems

Parašė Tim Bell, Ian H. Witten ir Mike Fellows

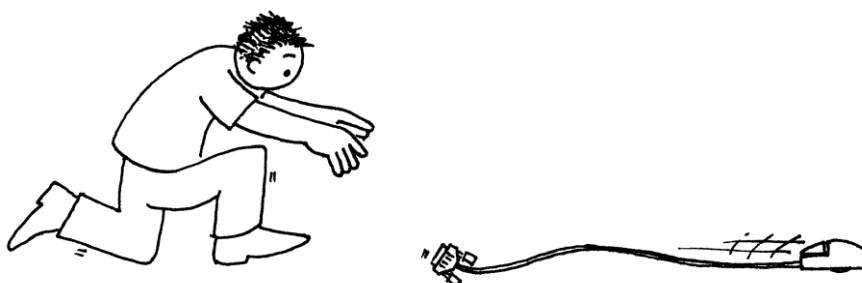
Darbui klasėje pritaikė Robyn Adams ir Jane McKenzie

Iliustravo Matt Powell

2015 m. atnaujino Sam Jarman

Į lietuvių kalbą išvertė Valentina Dagienė ir Eglė Jasutė, 2015 m.

Redagavo Audra Ivanauskienė



<http://csunplugged.org/>



Kūrybinių bendrijų licencija
© Computer Science Unplugged (csunplugged.org), 2015

Turiny

Įvadas	4
Padėka	6
I dalis Duomenys <i>Informacijos atvaizdavimas</i>	7
Duomenys. Pradžią pradžia	8
Taškų skaičiavimas. Dvejetainiai skaičiai	9
Spalvinimas skaičiais. Paveikslai kompiuteryje	25
Pakartok tai dar kartą! Teksto suspaudimas	34
Kortelių keitimo magija. Klaidų radimas ir taisymas	43
Dvidešimt spėjimų. Informacijos teorija	50
II dalis Kaip veikia kompiuteris? <i>Algoritmai</i>	56
Kaip veikia kompiuteris?	57
Laivų mūšis. Paieškos algoritmai	58
Lengviausias ir sunkiausias. Rikiavimo algoritmai	77
Dar greičiau! Tinklinis rikiavimas	84
Purvinas miestas. Minimalūs jungiantieji medžiai	90
Apelsinų žaidimas. Maršruto parinkimas ir tinklo aklavietė	96
Akmeninės plokštelės. Tinklo komunikacijų protokolas	99
III dalis Kaip pasakyti kompiuteriui, ką jis turi daryti? <i>Algoritmų užrašymas</i>	106
Kaip pasakyti kompiuteriui, ką jis turi daryti?	107
Lobio ieškojimas. Baigtiniai automatai	108
Nurodymų tvarka. Programavimo kalbos	124
IV dalis Ko kompiuteris negali? <i>Sunkiai sprendžiami uždaviniai</i> ...	129
Sunkiai sprendžiami uždaviniai	130
Vargšas kartografas. Grafo spalvinimas	131



Turistų miestas. Dominuojančios sekos	143
Ledo kelias. Šteinerio medžiai.....	150
V dalis Informacijos šifravimas <i>Kriptografija</i>	159
Informacijos šifravimas	160
Dalijimasis paslaptimis. Informacijos slėpimo protokolai	163
Monetos metimas. Kriptografijos protokolai.....	167
Vaikai šnipai. Viešojo rakto kriptografija	178
VI dalis Žmogiškasis kompiuterio veidas <i>Bendravimas su kompiuteriu</i>	190
Žmogiškasis kompiuterio veidas	191
Šokolado fabrikas. Vartotojo sąsajos projektavimas	193
Dialogas su kompiuteriu. Tiuringo testas	204



Ivadas

Kompiuteriai mus supa visur. Visiems reikia mokytis jais naudotis, o dauguma iš mūsų be jų neapsieiname nė dienos. Kaipgi veikia kompiuteriai? Kaip jie „mąsto“? Kaip žmonės sukuria greitai veikiančias ir lengvai naudojamą kompiuterių programas? Informatika – nepaprastai įdomus dalykas, nagrinėjantis šiuos klausimus. Šioje knygoje aprašomos įdomios ir smagios įvairaus amžiaus mokiniams skirtos užduotys, vadinamos veiklomis, supažindinama su kompiuterio veikimo pagrindais. Svarbiausia, kad visa tai pateikiama net nesiliečiant prie paties kompiuterio!

Ši knyga pravers ne tik informacinių technologijų būreliuose, bet ir įprastose pamokose. Net ir ne informatikai gali smagiai mokytis kompiuterio pagrindų drauge su mokiniais. Knygoje lengva ir suprantama kalba aprašoma daug veiklų, siejamų su informatikos pagrindais. Pateikiami visų užduočių atsakymai, kiekviena veikla baigiama skyreliu „Apie ką visa tai?“, kuriame aiškinama atliktos veiklos teorija.

Daugelis veiklų grindžiamos matematika, pavyzdžiui, dvejetainiai skaičiai, žemėlapiai ir grafai, modeliai ar struktūros, rikiavimo užduotys ir, žinoma, kriptografija. Kitos veiklos labiau siejamos su informacinėmis technologijomis, kompiuterių veikimo pagrindais. Pateikiamas prasmingas kontekstas skatina mokinius aktyviai bendrauti, kartu spręsti problemas, atlikti kūrybinį darbą ir mąstyti. Šios veiklos ypač lavina mokinių informatinį mąstymą, kurio siekia modernios mokyklų programos.

Daug papildomos ir laisvai prieinamos medžiagos, įskaitant animaciją ir paveikslus, pateikiama projekto *Unplugged* svetainėje csunplugged.org. 2015 m. knyga atnaujinta, naujai perdaryta ir svetainė, joje pateikta daug atvirosios medžiagos, sukurta patogesnė prieiga, labiau siejant su informatika ir informatiniu mąstymu mokykloje aiškiau išdėstyta ugdymo programos struktūra.

Šią knygą parašė trys informatikos dėstytojai ir du mokytojai remdamiesi didele patirtimi mokykloje ir atsižvelgdami į šimtus pastabų bei pasiūlymų, gautų per porą dešimtmečių iš šių medžiagų naudojančių mokytojų. Įsitikinta, kad dauguma svarbių informatikos konceptų gali būti mokoma be kompiuterio, iš tiesų, kartais kompiuteris pernelyg atitraukia mokinių dėmesį nuo mokymosi. Dažnai informatikos pamokos pradedamos programavimo mokymu, tačiau tai motyvuoja ne visus mokinius, todėl tokia pradžia gali tapti reikšminga kliūtimi norint sudominti mokinius šiuolaikiniais informatikos principais. Taigi išjunkite kompiuterius ir pasiruoškite mokytis tikrosios informatikos!

Šią knygą galima laisvai parsisiųsti ir ja naudotis – tokią galimybę suteikia Google Inc. parama. Knyga platinama pagal kūrybinių bendrijų licenciją nekomerciniais tikslais (*Creative Commons Attribution-NonCommercial-ShareAlike*). Tai reiškia, kad ją galima laisvai dalytis: ją kopijuoti, platinti ar persiųsti. Taip pat leidžiama knygą atnaujinti. Visa tai galima daryti laikantis šių sąlygų: būtina nurodyti knygos autorius, nenaudoti jos komerciniais tikslais, pakeista ar pertvarkyta arba šios medžiagos pagrindu sukurta nauja knyga turi būti platinama pagal panašią arba tą



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

pačią licenciją. Daugiau informacijos apie šią licenciją galite rasti internete, ieškokite CC BY-NC-SA 3.0.

Siūlome šią medžiagą naudoti klasėje ar užklasinėje veikloje: galite išsispausdinti knygos kopiją ir dauginti darbo lapus mokiniams. Klauskite autorių ir teikite jiems siūlymų (žr. csunplugged.org).

Knyga išversta į daugelį kalbų. Apie vertimus į kitas kalbas galite sužinoti svetainėje csunplugged.org.



Padėka

Daug mokinių ir mokytojų padėjo tobulinti šią knygą. Pirmieji išbandžiusieji čia aprašytas veiklas buvo mokiniai ir jų mokytojai iš Kanados Britų Kolumbijos provincijos sostinės Viktorijos *South Park* mokyklos, Naujosios Zelandijos Kraistčerčo *Shirley, Ilam* ir *Westburn* pradinių mokyklų. Nuoširdžiai dėkojame mokytojams Lindai Picciotto, Karen Able, Bryonui Porteousui, Paului Cathro, Tracy Harrold, Simone Tanoa, Lorraine Woodfield ir Lynn Atkinson, kad įsileido į savo pamokas ir teikė vertingų siūlymų, kaip tobulinti veiklas. Gwenda Bensemman specialiai mums parengė keletą veiklų pamokų ir pasiūlė, kaip ir ką keisti. Richardas Lyndersas ir Sumant Murugesh taip pat padėjo išbandyti veiklas pamokose. Dalį kriptografijos veiklų sukūrė Kenas Noblitzas. Kai kurias veiklas praktiškai atliko Viktorijos „Mathmania“ grupė, padedama Kathy Beveridge. Pradinius paveikslus kūrė Malcolmas Robinsonas ir Gail Williams. Gavome patarimų iš Hanso Knutsono. Prie šio projekto daug prisidėjo Mattas Powellas. Esame dėkingi Braino Masono mokslo ir technikos koncernui (angl. *Brian Mason Scientific and Technical Trust*) už dosnų šios knygos pradinės versijos finansavimą.

Ypač dėkojame Paului ir Ruth Ellen Howardams, kurie testavo daugelį šios knygos veiklų ir pateikė daugybę vertingų pasiūlymų. Peteris Hendersonas, Bruce'as McKenzie, Joan Mitchell, Nancy Walker–Mitchell, Gwen Stark, Tony Smithas, Timas A. H. Bellas¹, Mike'as Hallettas ir Haroldas Thimbleby taip pat pasidalijo savo mintimis.

Esame dėkingi savo šeimų nariams: Bruce'ui, Franui, Grantui, Judith ir Pam už paramą ir Andrew, Annai, Hannah'i, Maxui, Michaelui² ir Nikki, kurie buvo šio darbo įkvėpėjai ir pirmieji vaikai, testuodavę naują veiklą.

Ypač dėkojame *Google Inc.* už projekto *Unplugged* finansavimą ir galimybę laisvai platinti šią knygą.

Laukiame atsiliepimų ir siūlymų apie veiklas. Autoriams galite rašyti svetainėje csunplugged.org.

Į lietuvių kalbą šią knygą išvertė Vilniaus universiteto Matematikos ir informatikos instituto mokslininkės prof. dr. Valentina Dagienė ir dr. Eglė Jasutė. Redagavo Audra Ivanauskienė.

¹ Nesusijęs su pirmuoju autoriumi.

² Teksto suspaudimo veiklos autorius.



I dalis

Duomenys

Informacijos atvaizdavimas



Duomenys. Pradžių pradžia

Kaip informacija laikoma kompiuteryje?

Žodis „kompiuteris“ kilo iš lotynų kalbos žodžio „computare“, reiškiančio „skaičiuoti“ arba „sudėti“. Šiuolaikiniai kompiuteriai gali daug daugiau, nei tik suskaičiuoti. Juose gali būti laikoma, kaupiama visa biblioteka, jie gali padėti rašyti, ieškoti informacijos, groti muziką ar rodyti vaizdo įrašus. Kaipgi juose laikoma visa ši informacija? Neįtikėtina, bet kompiuteris tam naudoja tik du ženklus: nulį ir vienetą!

Kuo skiriasi duomenys ir informacija?

Duomenys yra žaliava – skaičiai, kuriais operuoja kompiuteris. Kompiuteris duomenis paverčia informacija (žodžiais, skaičiais ir paveikslais), kurią gali žmogus suprasti.

Kaip skaičiai, raidės, žodžiai ir paveikslai paverčiami nuliais ir vienetais?

Šiame skyriuje supažindinama su dvejetainiais skaičiais, aiškinama, kaip kompiuteriai vaizduoja paveikslus, kaip veikia fakso aparatai, kaip geriausia laikyti didelius informacijos kiekius, kaip išvengti klaidų ir kaip matuojama kompiuteryje laikoma informacija.



1 veikla

Taškų skaičiavimas. Dvejetainiai skaičiai

Santrauka

Duomenys kompiuteryje laikomi ir perduodami skaitmenų 0 ir 1 sekomis. Kaip pavaizduoti žodžius ir skaičius vartojant tik šiuos du ženklus?

Ryšiai su ugdymo programomis

- ✓ Matematika: skaičiai, skaičių vaizdavimas dvejetainėje sistemoje
- ✓ Matematika: algebra, begalinė seka, sekos taisyklės, dvejetainiai laipsniai

Gebėjimai

- ✓ Skaičiavimo
- ✓ Atitikčių nustatymo
- ✓ Sekos sudarymo

Amžius

- ✓ Nuo 6 metų

Priemonės

- ✓ Penkios dvejetainių skaičių kortelės. Rekomenduojama padaryti abipusių A4 formato kortelių: šypsenėlės vienoje pusėje, taškai – kitoje pusėje.

Kiekvienam mokiniui reikia:

- ✓ 5 kortelių
- ✓ Darbo lapo su dvejetainiais laipsniais

Kitų veiklų priemonės kiekvienam mokiniui:

- ✓ Darbo lapas „Dvejetainiai skaičiai“
- ✓ Darbo lapas „Slapta žinutė“
- ✓ Darbo lapas „Elektroninis paštas ir modemai“
- ✓ Darbo lapas „Didesnių už 31 skaičių vaizdavimas“
- ✓ Darbo lapas „Daugiau apie dvejetainius skaičius“



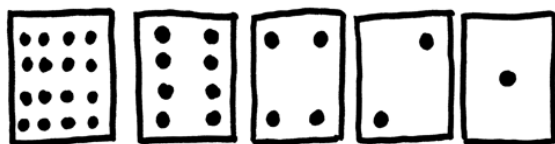
Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Dvejetainiai skaičiai

Įvadas

Veikla pristatoma visai grupei. Šiai veiklai reikia penkių A4 formato kortelių, kurių vienoje pusėje surašyti taškai, o kita pusė – tuščia. Parenkami penki mokiniai, jie atsistoja prieš klasę ir laiko šias korteles. Kortelės rodomos tokia tvarka:



Diskusija

Iš pradžių kortelės rodomos tuščiaja puse. Atverčiama pirmoji kortelė iš dešinės stebinčių mokinių atžvilgiu. Mokinių prašoma spėti, kiek taškų yra kortelėje, esančioje atverstosios kortelės dešinėje. Atspėjus atverčiama antroji kortelė. Mokiniai turi pastebėti ryšį tarp gretimų kortelių taškų skaičiaus. (Kiekvienoje tolesnėje kortelėje yra dvigubai daugiau taškų, nei buvo prieš tai atverstoje.)

Kiek taškų turėtų būti šeštoje kortelėje? (32) Septintoje?..(64)

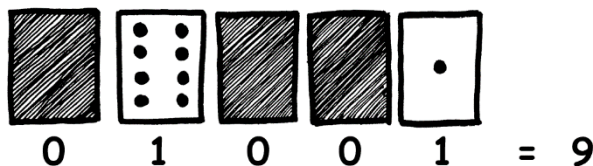
Stovintys mokiniai korteles vėl atverčia tuščiomis pusėmis į priekį. Prašoma kortelėmis pavaizduoti įvairius skaičius: 6 (atverčiamos 4 ir 2 taškų kortelės), 15 (atverčiamos 8, 4, 2 ir 1 taško kortelės). Taisyklė: nereikalingos kortelės turi būti užverstos, o reikalingos – atverstos.

Koks galimas mažiausias taškų skaičius kortelėje? (Iš tikrųjų 0.)

Toliau verčiant korteles vaizduojami skaičiai iš eilės nuo nulio.

Grupės nariai gali prieiti arčiau pažiūrėti, kaip verčiamos kortelės rodant įvairius skaičius. Galima keisti korteles laikančius mokinius.

Užversta taškų kortelė atitinka 0. Atversta kortelė atitinka 1. Taip gaunama dvejetainė skaičiaus išraiška.



Mokinių prašoma parodyti 01001. Koks dešimtainis skaičius atitinka šį dvejetainį skaičių? (9) Koks dvejetainis skaičius atitinka dešimtainį skaičių 17? (10001)

Tokiu būdu skirtingi skaičiai vaizduojami keletą kartų, kol mokiniai perpranta sistemą. Siūloma atlikti ir kitokias dvejetainių skaičių veiklas.



Darbo lapas *Dvejetainiai skaičiai*

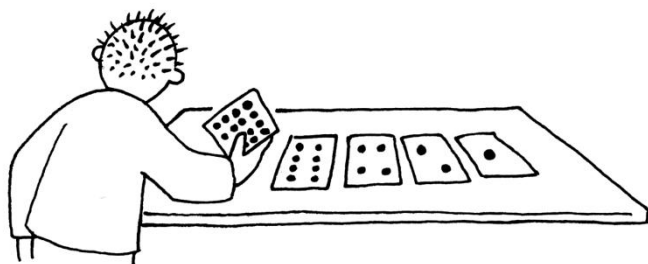
Mokytis skaičiuoti

Jūs tikrai mokate skaičiuoti, juk taip? O dabar išbandykite naują skaičiavimo būdą!

Ar žinote, kad kompiuteriai vartoja tik 0 ir 1? Viskas, ką girdite ir matote kompiuteryje (žodžiai, paveikslai, vaizdo įrašai ir net garsai), užrašyta tik dviejų skaitmenų sekomis! Tolesnių veiklų tikslas – išmokyti siųsti koduotas žinutes draugams taip, kaip daro kompiuteris.

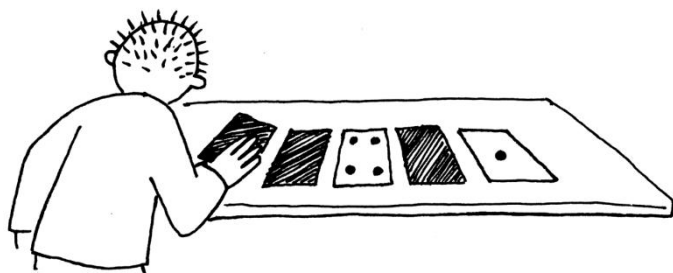
Įvadas

Sukarpykite taškų korteles (tolesnis lapas).



Išdėliokite korteles iš eilės nuo 1, 2, 4, ... atversdami tuščią pusę į viršų.

Atverskite korteles, kurių taškų suma būtų 5. Kortelių tvarkos nekeiskite!

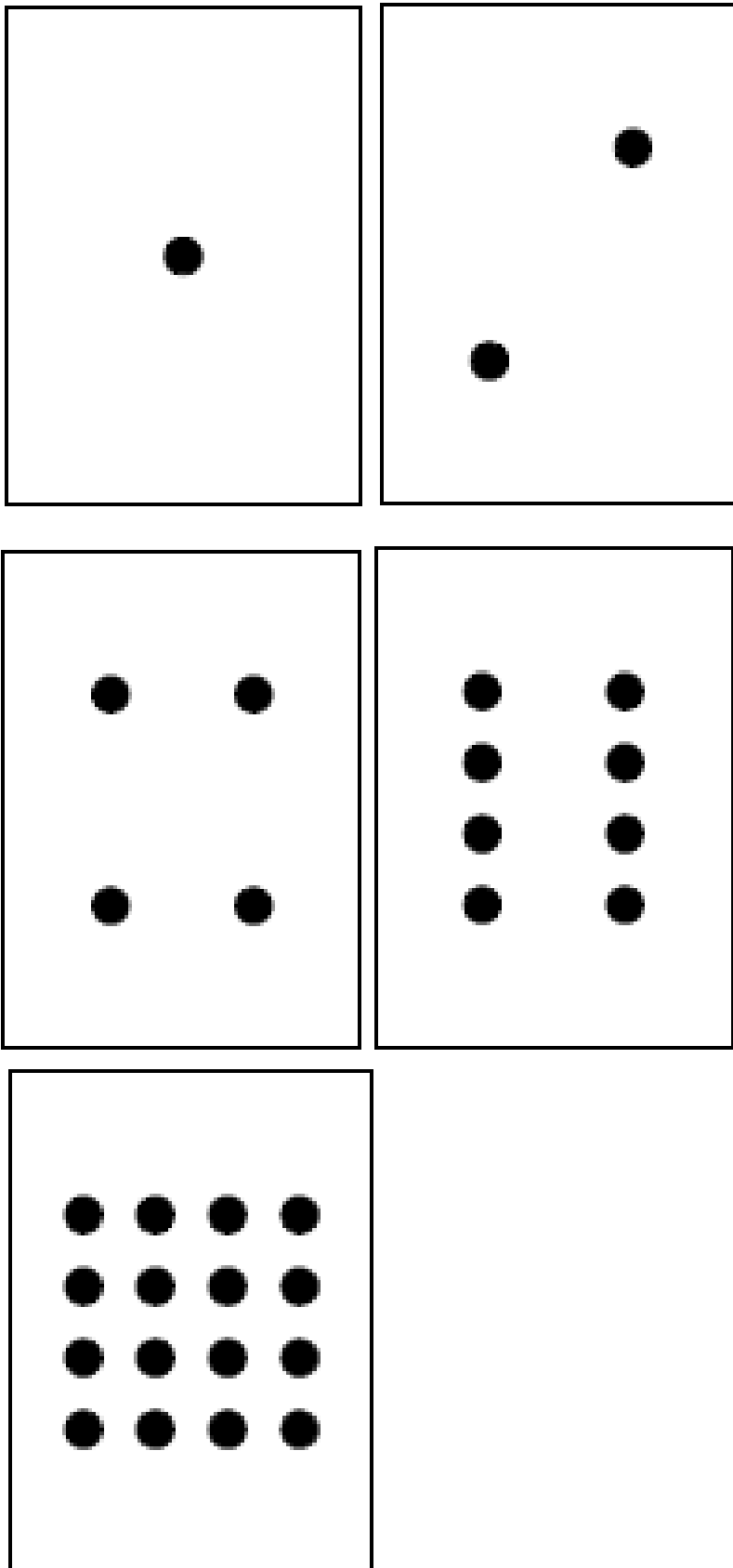


Atverskite korteles, kurių taškų suma būtų 3, 12, 19. Ar galima daugiau nei vienu būdu atversti korteles, kad gautumėte tą patį skaičių? Kokį didžiausią taškų skaičių galite atversti? Kokį mažiausią skaičių galite atversti? Ar tarp mažiausio ir didžiausio taškų skaičiaus yra skaičius, kurio negalima atversti kortelėmis?

Gudručiams. Pabandykite kortelėmis pavaizduoti skaičius iš eilės 1, 2, 3, 4, ... Gal galite sugalvoti logišką ir praktišką kortelių atvertimo metodą, kad verčiant korteles skaičiai didėtų vienetu?



Dvejetainiai skaičiai





Q



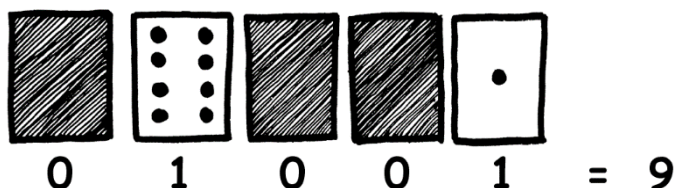






Darbo lapas *Dvejetainiai skaičiai*

Dvejetainė sistema turi tik du skaitmenis: **nulį** ir **vienetą**. Jei kortelė užversta, tai reiškia **0**, jei atversta – **1**, kaip parodyta žemiau:



Kokį dešimtainį skaičių atitinka **10101**? Kokį dešimtainį skaičių atitinka **11111**?

Kurią mėnesio dieną gimėte? Užrašykite tą dieną dvejetainė sistema. Nustatykite, kuris dvejetainis skaičius yra draugo gimimo dienos skaičius.

Užrašykite, kuriuos dvejetainius skaičius atitinka šie ženklai:

$$\begin{array}{c} \boxed{\times} \boxed{\checkmark} \boxed{\times} \boxed{\times} \boxed{\checkmark} = \\ (\checkmark=1, \times=0) \end{array}$$

$$\begin{array}{c} \uparrow \downarrow \uparrow = \\ (\uparrow=1, \downarrow=0) \end{array}$$

$$\begin{array}{c} \bigcirc \bigcirc \bigcirc \bigcirc \bigcirc = \\ (\odot=1, \bigcirc=0) \end{array}$$

$$\begin{array}{c} \text{☐} \text{☐} \text{☐} \text{☐} = \\ (\text{☐}=1, \text{☐}=0) \end{array}$$

$$\begin{array}{c} \text{☺} \text{☹} = \\ (\text{☺}=1, \text{☹}=0) \end{array}$$

$$\begin{array}{c} \text{👍} \text{👎} \text{👍} \text{👎} = \\ (\text{👍}=1, \text{👎}=0) \end{array}$$

$$\begin{array}{c} + + \times + = \\ (+=1, \times=0) \end{array}$$

$$\begin{array}{c} \curvearrowright \curvearrowright \curvearrowright \curvearrowright \curvearrowright = \\ (\curvearrowright=1, \curvearrowleft=0) \end{array}$$

$$\begin{array}{c} \blacktriangle \blacktriangledown \blacktriangle \blacktriangledown \blacktriangledown = \\ (\blacktriangle=1, \blacktriangledown=0) \end{array}$$

$$\begin{array}{c} \spadesuit \spadesuit \spadesuit \spadesuit \spadesuit = \\ (\spadesuit=1, \clubsuit=0) \end{array}$$

Gudručiams. Juostelėmis, kurių ilgis 1, 2, 4, 8 ir 16 matavimo vienetų, parodykite, kaip suduriamos bet kokio ilgio iki 31 matavimo vieneto juostelės. Be to, galite nustebinti draugus, parodydami, kad turėdami tik penkis svarelius galite pasverti gana sunkius daiktus!

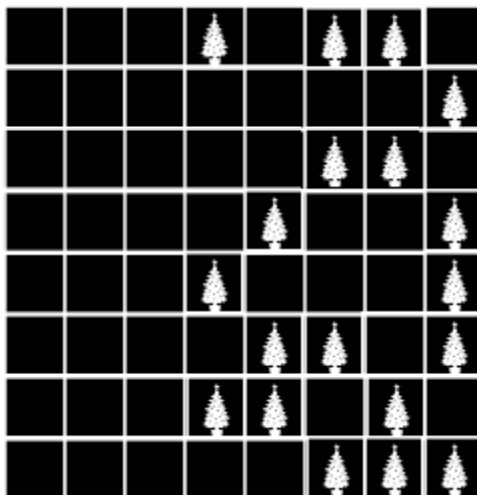
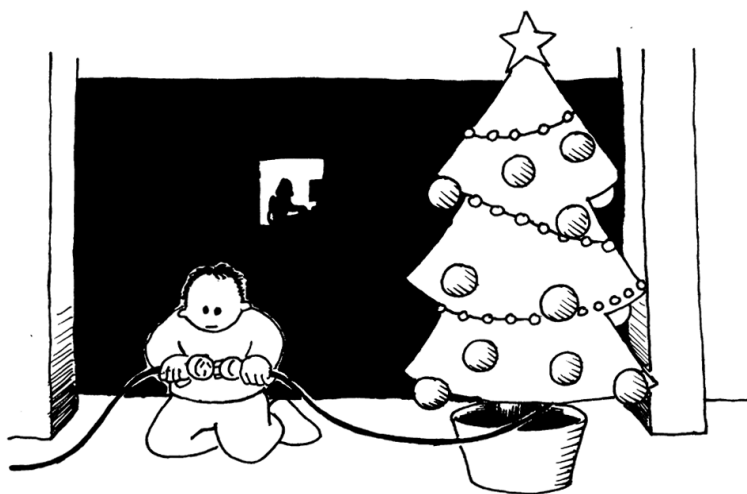


Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Darbo lapas *Slaptų žinučių siuntimas*

Prieš pat Kalėdas Tomą atsitiktinai užrakino vienoje parduotuvėje viršutiniame pastato aukšte. Aišku, jis norėjo kuo greičiau grįžti namo su dovanomis. Taigi ką daryti? Tomas bandė skambinti, rėkti, tačiau niekas jo negirdėjo. Pro langą pamatė žmogų, dirbantį kompiuteriu pastate kitoje gatvės pusėje. Kaip atkreipti jo dėmesį? Tomas apsidairė ieškodamas, kuo galėtų pasinaudoti. Jam šovė mintis perduoti žinutę Kalėdų eglutės lemputėmis, todėl šias lemputes sujungė taip, kad galėtų kiekvieną išjungti ir vėl įjungti. Tomas nusprendė perduoti savo žinutę dvejetainiu kodu, kurį turėtų suprasti žmogus, dirbantis kompiuteriu. Ką užkodavo Tomas? Ar jūs galėtumėte taip padaryti?

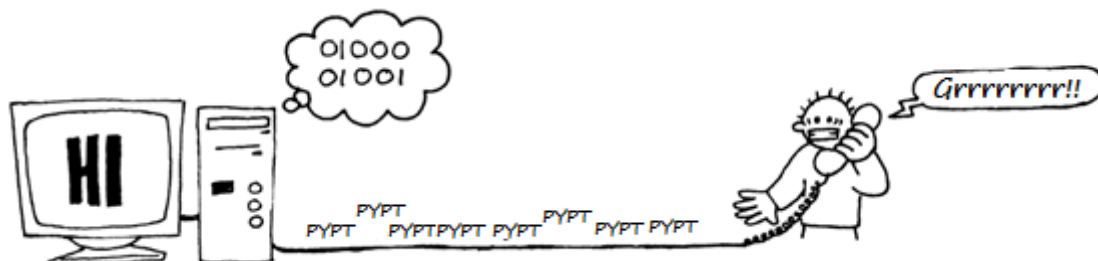


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	ą	b	c	č	d	e	ę	ė	f	g	h	i	į	y	j
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
k	l	m	n	o	p	r	s	š	t	u	ų	ū	v	z	ž

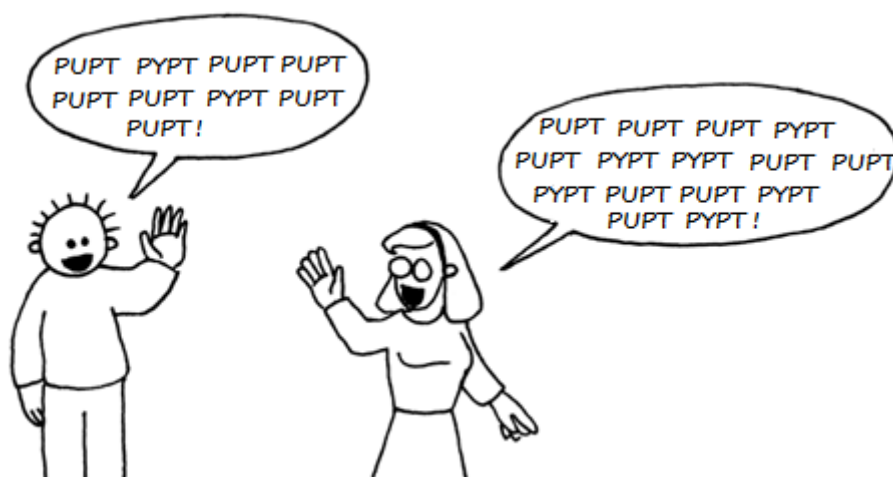


Darbo lapas *Elektroninis paštas ir modemai*

Kompiuteriai, prijungti prie interneto per modemą, informaciją siunčia dvejetainiu kodu naudodami pyptelėjimus. Aukšto tono pyptelėjimas reiškia 1, žemo tono – 0. Pyptelėjimai sklinda labai greitai. Taip greitai, kad girdime ištisą cypimo garsą. Jei niekada jo negirdėjote, tai įsiklausykite į prijungto prie interneto modemo skleidžiamą garsą.



Tuo pačiu dvejetainiu kodu, kurį Tomas taikė slaptų žinučių siuntimo užduotyje, parašykite el. laišką. Darykite atidžiai, kad nekiltų sunkumų skaitančiajam – jums nereikia būti tokiems greitiems kaip modemui ☺



Darbo lapas *Didesnių už 31 skaičių vaizdavimas*

Paimkite anksčiau naudotas dvejetaines korteles su taškais. Kiek taškų reikėtų nupiešti septintoje iš eilės kortelėje? Kiek aštuntoje? Pagal kokią taisyklę galima suskaičiuoti, kiek taškų reikia nupiešti kiekvienoje naujoje kortelėje? Jau įsitikinote, kad skaičiams nuo 1 iki 31 pavaizduoti užtenka penkių kortelių.

Įsižiūrėkite į seką. Ar galite nusakyti sekos narių ryšį?

1, 2, 4, 8, 16...

Sudėkite: $1 + 2 + 4 = ?$ Kiek gaunate?

Dabar sudėkite: $1 + 2 + 4 + 8 = ?$

Kiek gaunate sudėję visus skaičius nuo pirmosios iki penktosios kortelės?

Ar teko kada girdėti apie skaičiavimą „pirštais“? Išmėginkite šį būdą, kai skaičiai didesni kaip 10. Naudodami dvejetainę sistemą, įsivaizduokite, kad vienas vienos rankos pirštas yra kortelė. Šitaip galite gauti 32 skaičius. Nepamirškite, kad 0 – taip pat skaičius!

Pabandykite skaičiuoti iš eilės. Pakeltas pirštas reiškia 1, nuleistas – 0.

Abiejų rankų pirštais galite suskaičiuoti nuo 0 iki 1023. Tai 1024 skaičiai!

Jei vienos rankos pirštais galite pavaizduoti 32 skaičius, o abiejų rankų – $32 \times = 1024$ skaičius, kiek skaičių galėtų pavaizduoti Keturrankis pabaisiukas?



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Darbo lapas *Dar daugiau apie dvejetainius skaičius*

1. Kaip manote, kas nutiktų, jei dvejetainio skaičiaus pabaigoje prirašytumėte nulį? Jei nulį prirašytumėte dešimtainio skaičiaus pabaigoje, gautumėte 10 kartų didesnį skaičių už pradinį. Pavyzdžiui, prie 9 prirašę nulį, gautumėte 90, kuris yra 10 kartų didesnis už 9.

Taigi kas nutiktų, jei nulį prirašytumėte dvejetainio skaičiaus pabaigoje? Pabandykite:

1001 → 10010
(9) (?)

Sugalvokite hipotezę, pritaikykite ją keliems pavyzdžiams. Suformuluokite taisyklę. Kaip manote, kodėl taip atsitinka?

2. Ankstesnėse užduotyse naudojome korteles su taškais. Kiekviena kortelė atitinka bitą kompiuteryje. Žodis „bitas“ kilo iš anglų k. frazės „binary digit“ (liet. *dvejetainis skaitmuo*). Lietuvių kalbos abėcėlės raidėms (32 didžiosioms ir antra tiek mažosioms) koduoti pakaktų 6 kortelių arba, kitaip sakant, bitų. Tačiau kompiuteris turi atpažinti ne tik raides, bet skirti didžiąsias ir mažąsias raides, skaitmenis, aritmetinių operacijų ženklus, skyrybos ženklus, specialiuosius ženklus ir pan.

Pažvelgę į kompiuterio klaviatūrą, matome daug ženklų. Kaip manote, kiek bitų kompiuteriui reikia visiems šiems ženklams užkoduoti?

Šiuolaikiniuose kompiuteriuose teksto ženklai dažniausiai koduojami 8 bitais. Dėl to ir kompiuterio atmintis skirstoma į 8 bitų grupes, vadinamas baitais. Vieno baito pakanka kelių kalbų abėcėlėms ir dar daugeliui kitų reikalingų ženklų užkoduoti. Daugiau paskaityti apie ženklų kodavimą galite V. Dagienės, G. Grigo ir T. Jevsikovo knygoje „Programinės įrangos lokalizavimas“ (<http://www.mii.lt/files/lokalizavimas.pdf>)

Suskaičiuokite, kiek ženklų telpa į vieną baitą.



Apie ką visa tai?

Šiuolaikiniai kompiuteriai pagrįsti dvejetainė sistema informacijai kaupti ir apdoroti. Sistema vadinama dvejetainė, nes turi tik du skaitmenis. Dar sakoma: sistema, kurios pagrindas 2 (mūsų įprasta dešimtainė sistema gali būti vadinama sistema, kurios pagrindas 10). Nulis ir vienetas vadinami bitais Bitas kompiuterio atmintyje reiškia tranzistorių, kuris tam tikru momentu yra įjungtas arba išjungtas, atitinkamai įtampa yra arba jos nėra.



Perduodant duomenis telefonu arba radijo bangomis, vietoj nulių ir vienetų naudojami žemi ir aukšti tonai. Magnetinėse laikmenose bitai reiškiami magnetinio lauko kryptimis: šiaurė → pietūs arba pietūs → šiaurė.



CD, CD-ROM ir DVD laiko bitus optiškai – paviršiaus dalys atspindi šviesą arba jos neatspindi.



Kai skiriamos dvi reikšmės, daug paprasčiau sukurti prietaisus informacijai apdoroti. Jei būtų naudojama dešimtainė sistema, kompaktinis diskas turėtų turėti 10 būdų skaitmenims nuo 0 iki 9 išreikšti. Sukurti tokį prietaisą būtų brangu ir sudėtinga. Nors sakoma, kad kompiuteris duomenis vaizduoja nuliais ir vienetais, bet iš tiesų jame yra tik aukštoji ir žemoji įtampa arba šiaurės ir pietų magnetinio lauko kryptis ar pan. Visa tai žymima 0 ir 1, kad būtų lengviau suprantama žmonėms. Šiais bitais kompiuteryje atvaizduojami dokumentai, paveikslai, dainos, skaičiai, programos ir programėlės.

Vienu bitu atvaizduojama labai mažai informacijos, todėl bitai jungiami į sekas po 8, kurios gali išreikšti skaičius nuo 0 iki 255. Aštuonių bitų grupė vadinama baitu. Kompiuterio sparta priklauso nuo to, kiek bitų vienu metu jis gali apdoroti. Pavyzdžiui, 32 bitų kompiuteris vienos operacijos metu gali apdoroti 32 bitų skaičius, o 16 bitų kompiuteris pirmiausia turėtų 32 bitų skaičių padalyti į dalis. Tai, žinoma, lėtina kompiuterio darbą. Taigi skaičiams, tekstui ir kitokiai informacijai laikyti ir apdoroti kompiuteris naudoja tik bitus ir baitus.

Kituose skyriuose supažindinama, kaip laikoma ir apdorojama kito pobūdžio informacija dvejetainiais skaitmenimis.



Sprendimai ir užuominos

Dvejetainiai skaičiai

3 pavaizduoti reikia kortelių su 2 ir 1 tašku.

12 pavaizduoti reikia kortelių su 8 ir 4 taškais.

19 pavaizduoti reikia kortelių su 16, 2 ir 1 tašku.

Tik vieninteliu būdu galima pavaizduoti kiekvieną iš šių skaičių.

Didžiausias skaičius, kurį galima pavaizduoti, yra 31. Mažiausias yra 0. Galima pavaizduoti visus skaičius nuo 0 iki 31. Kiekvieną iš jų galima pavaizduoti tik vienu būdu.

Gudručiams. Didinant skaičių vienetu, iš eilės apverčiamos taškais į viršų visos kortelės iš dešinės į kairę.

Dvejetainiai skaičiai

10101 = 21, 11111 = 31

Slaptų žinučių siuntimas

Užkoduota žinutė: PADĖKITE

Didesnių už 31 skaičių vaizdavimas

Jei sudedami visi skaičiai nuo pirmojo, tai jų suma visada bus vienetu mažesnė už kitą sekos skaičių.

Keturrankis pabaisiukas gali pavaizduoti $1024 \times 1024 = 1\,048\,576$ skaičių — nuo 0 iki 1 048 575!

Daugiau apie dvejetainius skaičius

Parašius nulį dvejetainio skaičiaus dešinėje, šis skaičius padidėja dvigubai. Kiekvienas vienetas dvejetainiame skaičiuje padvigubėja, todėl ir galutinis skaičius padvigubėja. (Dešimtainės sistemos skaičiaus dešinėje prirašius nulį, šis skaičius padidėja 10 kartų.)



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

2 veikla

Spalvinimas skaičiais. Paveikslai kompiuteryje

Santrauka

Piešiniai, paveikslai, nuotraukos ir kitokie vaizdai kompiuteryje koduojami bitais. Čia aiškinama, kaip tai daroma.

Ryšiai su ugdymo programomis

- ✓ Matematika: geometrija – formos ir erdvės
- ✓ Informatika: įvairių duomenų atvaizdavimas skaičiais; atminties taupymas – pasikartojančių duomenų taupesnis kodavimas

Gebėjimai

- ✓ Skaičiavimo
- ✓ Vaizdavimo

Amžius

- ✓ Nuo 7 metų

Priemonės

- ✓ Pateiktis „Spalvinimas skaičiais“

Kiekvienam mokiniui reikia:

- ✓ Darbo lapo „Vaikiškas faksas“
- ✓ Darbo lapo „Nupiešk savo paveikslą“



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

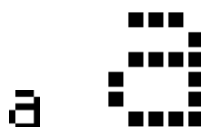
Spalvinimas skaičiais

Įvadas

Diskusiniai klausimai

1. Ką daro fakso aparatai?
2. Kuriose iš šių situacijų kompiuteriai turi įrašyti paveikslus? (Dirbant piešimo programa, žaidžiant interaktyviuosius žaidimus ar rengiant skaidres pateikčių programa.)
3. Kaip kompiuteriai gali koduoti paveikslus vien skaičiais?
(Norėdami geriau pasirengti šiai veiklai ir supažindinti su ja mokinius, galite suorganizuoti faksogramos siuntimą ar gavimą.)

Išsamus paaiškinimas



Kompiuterio ekranas yra tarsi tinklėlis, sudarytas iš mažų taškelių, vadinamų pikseliais (angl. *pixels*).

Kiekvienas nespaltvoto paveikslo pikselis yra arba juodas, arba baltas.

Aukščiau pateikta raidė „a“ padidinta tam, kad įsivaizduotume pikselius. Kompiuteriui įrašant paveikslą reikalinga tik informacija, kurie taškeliai yra juodi, o kurie – balti.

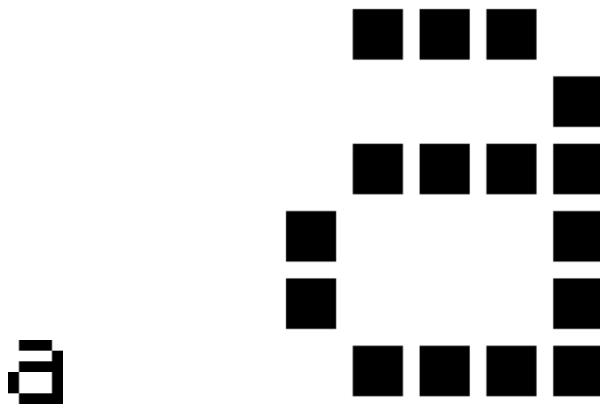
	■	■	■		1, 3, 1
				■	4, 1
	■	■	■	■	1, 4
■				■	0, 1, 3, 1
■				■	0, 1, 3, 1
	■	■	■	■	1, 4

Šis piešinys rodo, kaip galima koduoti paveikslus skaičiais. Pirmoje eilutėje yra vienas baltas pikselis, trys juodi ir vėl vienas baltas, todėl pirmoji eilutė užrašoma skaičių 1, 3, 1 seka.

Pirmasis skaičius visada nusako baltųjų pikselių skaičių. Jei pirmasis pikselis yra juodas, tada skaičių seka turi prasidėti nuliu. Darbo lape pateikiama keletas paveikslų, kuriuos siūloma mokiniams užkoduoti aprašytu metodu.



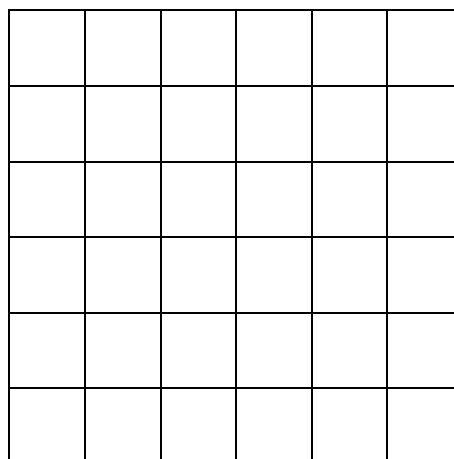
Spalvinimas skaičiais



▲ Raidė „a“, matoma kompiuterio ekrane, ir ta pati padidinta raidė, kai matomi ją sudarantys pikseliai.

	■	■	■		1, 3, 1
				■	4, 1
	■	■	■	■	1, 4
■				■	0, 1, 3, 1
■				■	0, 1, 3, 1
	■	■	■	■	1, 4

▲ Tas pats paveikslas, užkoduotas skaičiais.

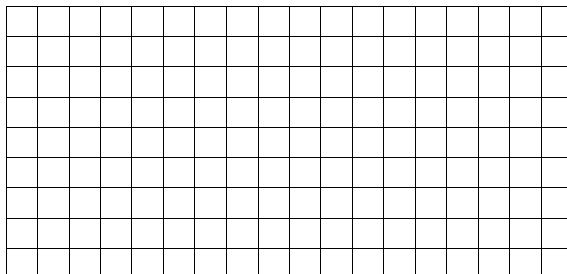


▲ Tuščias tinklėlis (naudojamas mokyti)

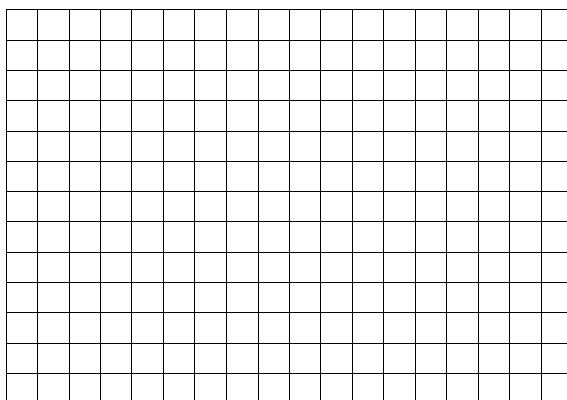


Darbo lapas *Vaikiškas faksas*

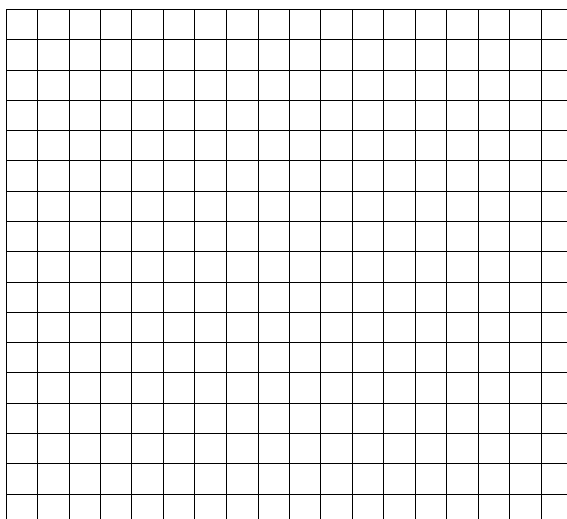
Pirmasis paveikslas yra paprasčiausias, paskutinis – sudėtingiausias. Atliekant užduotį, lengva padaryti klaidų, tad patariama piešti pieštuku ir po ranka turėti trintuką!



4, 11
4, 9, 2, 1
4, 9, 2, 1
4, 11
4, 9
4, 9
5, 7
0, 17
1, 15



6, 5, 2, 3
4, 2, 5, 2, 3, 1
3, 1, 9, 1, 2, 1
3, 1, 9, 1, 1, 1
2, 1, 11, 1
2, 1, 10, 2
2, 1, 9, 1, 1, 1
2, 1, 8, 1, 2, 1
2, 1, 7, 1, 3, 1
1, 1, 1, 1, 4, 2, 3, 1
0, 1, 2, 1, 2, 2, 5, 1
0, 1, 3, 2, 5, 2
1, 3, 2, 5

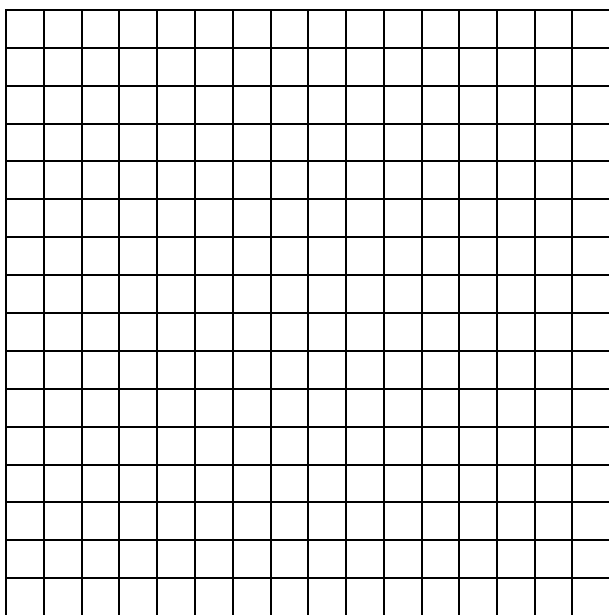


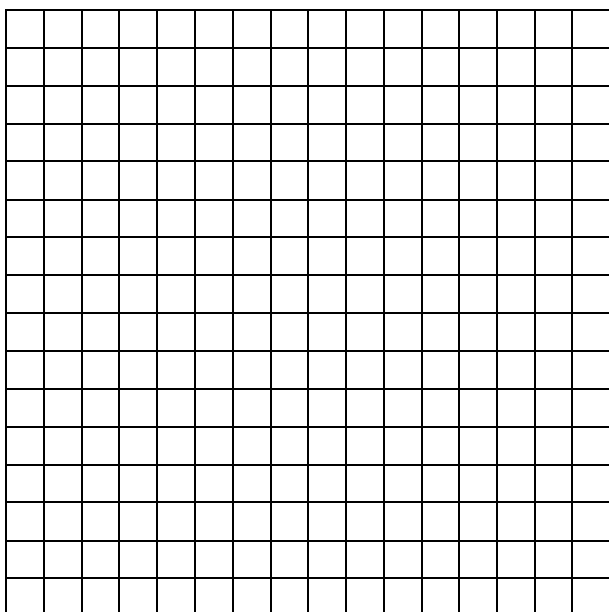
6, 2, 2, 2
5, 1, 2, 2, 2, 1
6, 6
4, 2, 6, 2
3, 1, 10, 1
2, 1, 12, 1
2, 1, 3, 1, 4, 1, 3, 1
1, 2, 12, 2
0, 1, 16, 1
0, 1, 6, 1, 2, 1, 6, 1
0, 1, 7, 2, 7, 1
1, 1, 14, 1
2, 1, 12, 1
2, 1, 5, 2, 5, 1
3, 1, 10, 1
4, 2, 6, 2
6, 6



Darbo lapas *Nupiešk savo paveikslą*

Dabar, kai jau žinome, kaip skaičiais galime koduoti paveikslus, kodėl nepabandžius patiems užkoduoti paveikslo, o paskui duoti jį kam nors iškoduoti? Sugalvokite paveikslą ir nupieškite jį viršutiniame tinklelyje. Kai baigsite, užrašykite kodą skaičiais šalia apatinio tinklelio ant dešinėje esančių linijų. Perkirpkite lapą per punktyrinę liniją ir duokite apatinę lapo dalį su tuščiu tinkleliu draugui nuspalvinti. (Pastaba: nebūtina panaudoti viso tinklelio, jei piešinys nedidelis, galite palikti keletą tuščių eilučių apačioje.)

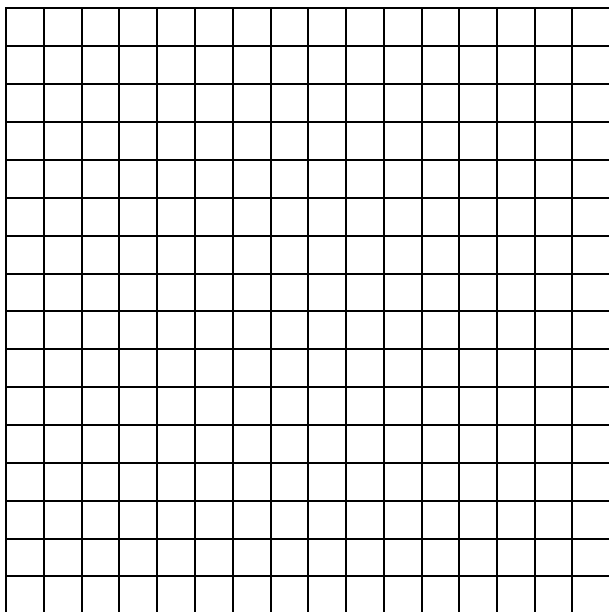


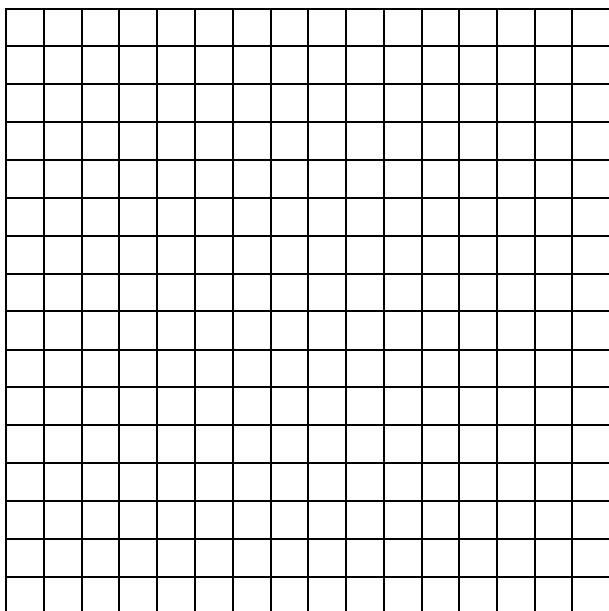




Darbo lapas *Nupiešk savo paveikslą*

Gudručiams. Jei norite nupiešti spalvotą paveikslą, spalvoms turite suteikti kodus (pavyzdžiui, galima sutarti, kad 0 žymi juodą spalvą, 1 – raudoną, 2 – žalią ir t. t.). Kaip ir anksčiau pikselius atvaizduosime dviem skaičiais: pirmasis nusakys vienodų pikselių (langelių) skaičių, antrasis – reikiamą spalvą. Nupieškite spalvotą paveikslą ir duokite jį draugui iškoduoti. Nepamirškite pasakyti, kuris skaičius atitinka kurią spalvą.







Užuominos ir galimi pakeitimai

1. Pabandykite piešti užsidėję kopijavimo popierių ant tinklelio viršaus taip, kad galutinis paveikslas būtų matomas be tinklelio. Tada paveikslas bus matomas aiškiau.
2. Užuoť spalvinę tinklelio langelius, mokiniai ant didesnio tinklelio gali klijuoti ar dėti lipnius kvadratinis lapelius ar kartono kvadratėlius.

Diskusija

Paprastai pikselių skaičius eilutėje yra ribojamas ir užrašomas dvejetainiu skaičiumi. Kaip užrašytumėte dvylikos juodų pikselių eilę skaičiais tik iki 7? (Vienas iš būdų užkoduoti dvylikos juodų pikselių eilę yra pirmiausia nurodyti 7 juodus pikselius, tada nulį baltųjų ir galų gale likusius penkis juodus pikselius.)



Apie ką visa tai?

Fakso aparatai yra paprasčiausi kompiuteriai, kurie nuskenuoja nespaltotą popieriaus lapą į maždaug 1000×2000 pikselių tinklėlį. Tada tą nuskenuotą tinklėlį fakso aparatas per modemą siunčia į kitą fakso aparatą, kuris išspausdina pikselius ant popieriaus lapo. Dažnai faksu siunčiamuose paveiksluose yra didelių baltų (pavyzdžiui, paraščių) ir juodų pikselių blokų (pavyzdžiui, linijų). Spalvoti paveikslai dažniausiai turi labai daug pasikartojančių dalių. Įvairiais suspaudimo metodais programuotojai siekia sutaupyti atminties tokiems paveikslams laikyti. Metodus, aprašomas šiame skyriuje, vadinamas *vienodų eilėje kodavimu*–. Tai labai efektyvus būdas paveikslui suspausti. Jei paveikslų nesuspaustume, jų siuntimas užtruktų kur kas ilgiau, be to, jiems laikyti reiktų daugiau atminties. Tada siųsti paveikslus faksu ar įdėti juos į tinklalapį būtų beveik neįmanoma. Pavyzdžiui, faksu siunčiamas suspaustas paveikslas dažniausiai yra septynis kartus mažesnis, palyginti su jo tikruoju dydžiu. Tokio pat nesuspausto paveikslo siuntimas užtruktų septynis kartus ilgiau.

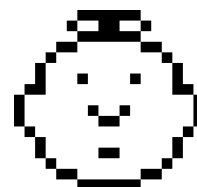
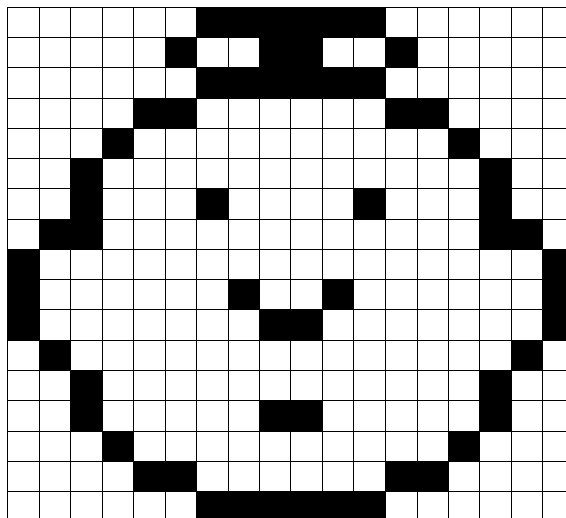
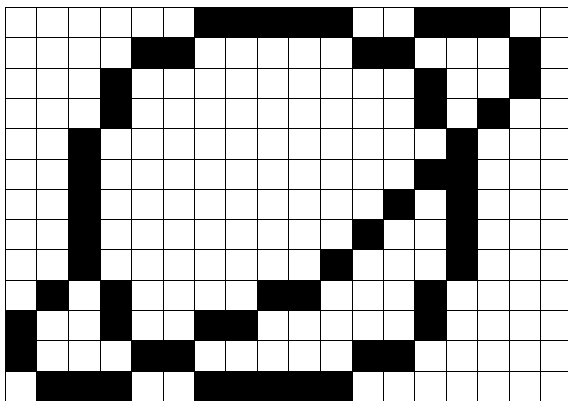
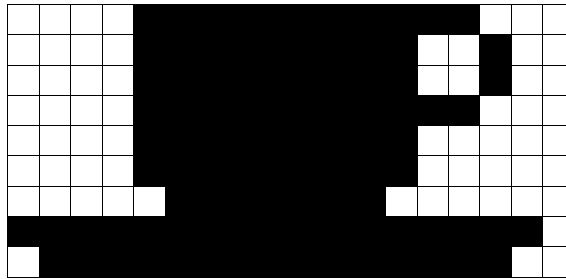
Nuotraukos dažniausiai suspaudžiamos iki dešimties ar net iki šimto kartų (panašiais metodais, pavyzdžiui, JPEG, GIF ir PNG). Tokių nuotraukų diske galima laikyti daugiau, o jų peržiūra tinklalapyje trunka tik dalį sekundės.

Paprastai programuotojas nusprendžia, kuris iš suspaudimo metodų labiausiai tinka paveikslams siųsti.



Sprendimai ir užuominos

Atsakymai. Darbo lapas „Vaikiškas faksas“



3 veikla

Pakartok tai dar kartą! Teksto suspaudimas

Santrauka

Kompiuterių atmintis, kurioje laikoma informacija, yra ribota, todėl ta informacija turi būti vaizduojama kuo efektyviau. Šiuo tikslu informaciją reikia suglaudinti. Užkodavus (t. y. suglaudinus) norimus saugoti duomenis ir juos išskleidus, kai prireikia, kompiuteris jų gali laikyti daugiau arba tokius suspaustus duomenis kur kas sparčiau persiųsti internetu.

Ryšiai su ugdymo programomis

- ✓ Lietuvių kalba: atpažinti pasikartojimus žodžiuose ir tekste
- ✓ Informatika: atminties taupymas nesaugant pasikartojančių duomenų

Gebėjimai

- ✓ Teksto kopijavimo

Amžius

- ✓ Nuo 9 metų

Priemonės

- ✓ Pateiktis „Pakartok tai dar kartą!“

Kiekvienam mokiniui būtinas:

- ✓ Darbo lapas „Pakartok tai dar kartą!“
- ✓ Darbo lapas „Gudručiams“
- ✓ Darbo lapas „Trumpas ir saldus“
- ✓ Darbo lapas „Tikriems gudručiams“



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

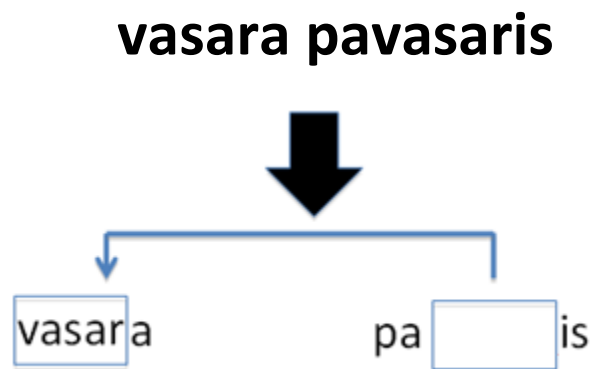
Pakartok tai dar kartą!

Įvadas

Kompiuteriai laiko ir persiunčia daugybę duomenų. Kad jiems nereikėtų labai daug atminties arba kad pernelyg ilgai neužtruktų informacijos siuntimas, kompiuteriai suspaudžia tekstą panašiai, kaip rodoma toliau.

Diskusija ir pavyzdžiai

Parodoma skaidrė „Saulė“. Ieškoma raidžių pasikartojimų eiliuotame tekste, pateiktame skaidrėje. Ar galima rasti pasikartojančių raidžių po dvi ar daugiau grupelių? O gal kartojasi net visas žodis ar frazė? (Pakeiskite juos kvadratėliais, kaip parodyta diagramoje žemiau.)



Saulė

Vasara pavasaris

Vasara pavasaris

Saulė šypsosi

Vasara pavasaris

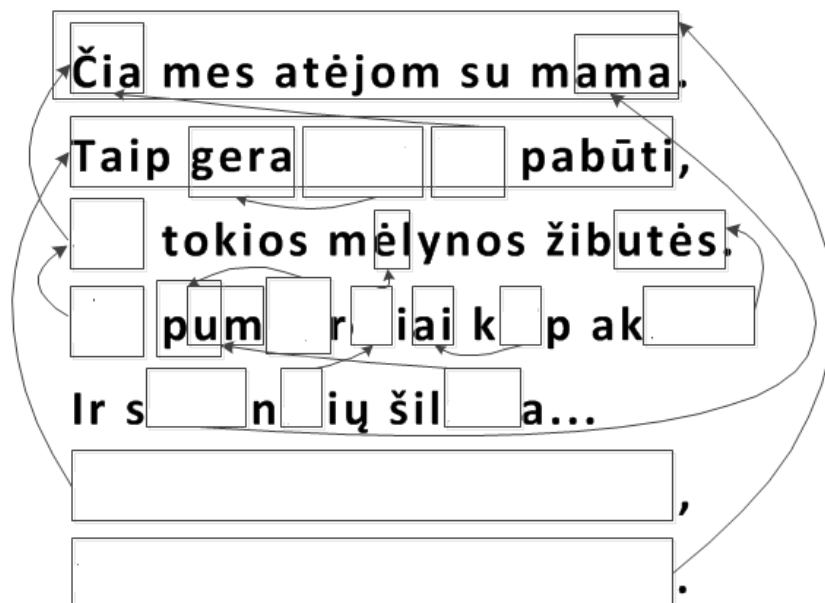
Vasara pavasaris

Į laukelį kviečia



Darbo lapas *Pakartok tai dar kartą!*

Šiame eilėraštyje trūksta daug raidžių ir žodžių. Ar galite užpildyti tuščius stačiakampius raidėmis ir žodžiais, kad eilėraštis būtų užbaigtas? Trūkstamas raides ir žodžius galite atsekti pagal rodykles.



Dabar patys pasirinkite kokį norite eilėraščių ar ketureilį ir suspauskite jo tekstą. Atkreipkite dėmesį, kad rodyklės turi rodyti į ankstesnes teksto dalis. Užkoduotas eilėraštis iškoduojamas tokia tvarka, kokia įprastai skaitome tekstą: iš kairės į dešinę ir iš viršaus į apačią.

Iššūkis: atkreipkite dėmesį, kiek nedaug pirminio teksto žodžių reikia palikti!

Užuomina: stenkitės neapkrauti teksto rodyklėmis. Rašydami eilėraščių palikite daugiau laisvos vietos aplink žodžius ir tarp raidžių, kad galėtumėte piešti stačiakampius ir rodykles.

Suspaustą tekstą lengviau užkoduoti, kai pasirašote atskirai sau visą eilėraščių ir tada nupiešiate kvadratėlius ir rodykles.

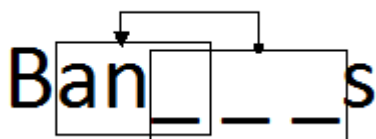


Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Darbo lapas *Gudručiams*

Kaip išspręstumėte šį galvosūkį?



Kartais trūkstamos teksto dalys rodo pačios į save. Šiuo atveju žodis gali būti iš koduotas teisingai, jeigu kopijuojame raides iš kairės į dešinę. Tada kiekviena raidė gali būti kopijuojama prieš tai, kai įrašoma. Toks kopijavimo būdas labai naudingas kompiuteriams, kai yra daug pasikartojimų.

Pabandykite nupiešti patys.

Kompiuteryje stačiakampiai ir rodyklės vaizduojami skaičiais. Pavyzdžiui,

Pasas

gali būti užrašytas kaip **Pas(2,2)**. Pirmasis skaičius **2** rodo antrą vietą nuo galo – kopijavimo vietos pradžią

Pas—

Antrasis skaičius **2** reiškia, kad kopijuosime dvi iš eilės einančias raides.

Pasa—

Pasas



Kadangi žodžiui užkoduoti naudojami du skaičiai, tai verta suspausti dviejų ar daugiau raidžių grupes, kitaip nebus sutaupoma vietos. Iš tiesų failo dydis gali padidėti, jeigu dviem skaičiais koduosime vieną raidę.

Sugalvokite žodžių ir suspauskite juos taip, kaip darytų kompiuteris. Ar jūsų draugai galėtų juos iš koduoti?



Darbo lapas *Trumpas ir saldus*

Kiek žodžių reikia čia iš tiesų?

Tarkime, jūs esate kompiuteris, kuris nori sutalpinti savo diske kuo daugiau informacijos. Išbraukite visas grupes po dvi ar daugiau pasikartojančių raidžių. Jos nebereikalingos, nes keičiamos rodyklėmis. Jūsų tikslas – išbraukti kuo daugiau raidžių.

Apvali, saldi gražuolė...

Šmurkšt, po suolu jau nupuolė.

„Kurgi bėgi? Grįžk į puodą...”

Bet, pakėlus savo kuodą,

Bulvė lekia per laukus:

„Nebegrišiu pas vaikus.”



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Darbo lapas *Tikriems gudručiams*

Ar pasiruošę *tikrai* „kietai“ suspausti?

Žemiau pateikta istorija buvo apdorota kompiuterio programa, kuri šioje istorijoje aptiko mažiausiai 1000 galimų išbraukti raidžių. Kiek galite rasti jūs? Prisiminkite, tik grupės po dvi ar daugiau pasikartojančių raidžių gali būti išbrauktos. Sėkmės!

„Vieną kartą, labai labai seniai, trys maži paršiukai nusprendė pasistatyti sau namus. Pirmasis paršiukas nebuvo labai protingas ir nusprendė pasistatyti namą iš šiaudų, nes taip būsią pigiau. Antrasis paršiukas irgi nebuvo labai protingas, todėl nusprendė pasistatyti namą iš šakų, siekdamas ekologiško įvaizdžio, nes tai buvo madinga net tais laikais. Trečiasis paršiukas buvo protingesnis negu jo broliai ir nusipirko krūvą plytų iš gretimo miesto, kad pasistatytų tvirtą ir patogų namelį.

Tuo pat po įkurtuvių vakarėlio pirmasis paršiukas suposi kėdėje ir skaitė knygą, kai kažkas pabeldė į duris. Tai buvo didelis piktas vilkas.

– Paršiuk, paršiuk, įleisk mane! – prašė vilkas.

– Ne, nė per nago juodumą! – sužviegė pirmasis paršiukas.

– Tada aš supyksiu, giliai įkvėpsiu ir nupūsiu tavo namą! – užriaumojo vilkas ir, labai supykęs, giliai įkvėpė, stipriai papūtė ir suniokojo namą. Išsigandęs pirmasis paršiukas iš visų jėgų nubėgo pas savo brolių į iš šakų pastatytą namą. Vos tik pasijuto saugus, vilkas vėl buvo šalia slenksčio.

– Paršiuk, paršiuk, įleisk mane! – prašė vilkas.

– Ne, nė per nago juodumą! – sužviegė antrasis paršiukas.

– Tada aš supyksiu, giliai įkvėpsiu ir nupūsiu tavo namą! – užriaumojo vilkas ir, labai supykęs, giliai įkvėpė, stipriai papūtė ir suniokojo šakų namą. Abu išsigandę paršiukai nubėgo pas trečiąjį brolių į plytų namą, bet vilkas lipo jiems ant kulnų ir tuo pat jau buvo prie durų slenksčio.

– Paršiuk, paršiuk, įleisk mane! – prašė vilkas.

– Ne, nė per nago juodumą! – sužviegė trečiasis paršiukas.

– Tada aš supyksiu, giliai įkvėpsiu ir nupūsiu tavo namą! – užriaumojo vilkas ir, labai supykęs, giliai įkvėpė, stipriai papūtė... Bet namas buvo pastatytas iš plytų! Vilkas bepūsdamas užduso, o namas liko stovėti. Tada vilkui šovė nauja mintis. Kaminas! Jis užlipo šakotu qžuolu ant stogo, tačiau pamatė, kad namas – be kamino, nes trečiasis paršiukas rūpinosi aplinka ir įrengė elektrinį šildymą. Labai įniršęs vilkas paslydo ir nukrito nuo stogo. Jis susilaužė kairę koją ir prarado savo orumą. Paršiukai juokėsi, kai vilkas nušlubavo atgal. Jie suprato, kaip protinga gyventi mieste, kur vieninteliai vilkai yra zoologijos sode. Paršiukai persikraustė į miestą ir ten gyveno ilgai ir laimingai.“



Apie ką visa tai?

Kompiuterio atminties talpa didėja kasdien – per paskutinius 25-ius metus standartinių kompiuterių ji padidėjo net milijoną kartų, bet matyt tai dar ne pabaiga. Kompiuteriuose gali būti laikomos knygos ar net ištisa biblioteka, muzikos įrašai ir filmai, jei tik yra laisvos vietos. Didelės apimties dokumentai internete yra labai didelė problema, nes ilgai užtrunka juos atsisiųsti. Dabar stengiamasi sumažinti kompiuterius – net telefonai ir rankiniai laikrodžiai gali laikyti labai daug informacijos.

Vienas šios problemos sprendimų štai toks: užuot pirkę kompiuterius, turinčius daugiau atminties, ar įsigiję spartesnį interneto ryšį, galime duomenis taip *suspausti*, kad jie užimtų mažiau vietos. Šis duomenų kodavimo ir iškodavimo procesas dažniausiai atliekamas kompiuteriu automatiškai. Mes to proceso beveik nepastebime. Nors iš tiesų kompiuteris atlieka daugiau veiksmų, mes matome tik tai, kad diske telpa daugiau informacijos ar sparčiau atsiunčiamas tinklalapis.

Sugalvota daug duomenų suspaudimo metodų. Atliekant šią veiklą, buvo naudojamas metodas, kai nuorodomis rodoma į prieš tai buvusių pasikartojančius simbolius tekste. Šis metodas vadinamas tiesiog „zip“, jį 1970 m. sugalvojo Izraelio mokslininkai. Metodas tinka bet kuriai kalbai, juo lengvai suspaudžiami ir perpus sumažinami duomenys. Todėl šis metodas dažnai naudojamas asmeniniuose kompiuteriuose bei sparčiuosiuose modemuose, juo suspaudžiami GIF ir PNG paveikslėliai, sumažinamas telefono linija siunčiamų duomenų kiekis, tad jų siuntimas kur kas spartesnis.



Sprendimai ir užuominos

Pakartok tai dar kartą³!

**Čia mes atėjom su mama.
Taip gera gera čia pabūti,
Čia tokios mėlynos žibutės.
Čia pumpurėliai kaip akutės
Ir samanėlių šiluma...
Taip gera gera čia pabūti,
Čia mes atėjom su mama.**

³ Parenkant eiles veiklai talkino Vilniaus universiteto studentė Indrė Tamošauskaitė



4 veikla

Kortelių keitimo magija. Klaidų radimas ir taisymas

Santrauka

Saugant duomenis laikmenoje ar siunčiant iš vieno kompiuterio į kitą, įprasta manyti, kad jie nesikeičia. Tačiau kartais įvyksta kas nors nenumatyta ir duomenys pasikeičia. Šioje veikloje magišku triuku parodoma, kaip nustatyti, kada duomenys yra sugadinti, ir kaip tai ištaisyti.

Ryšiai su ugdymo programomis

- ✓ Matematika: skaičiavimas ir įvertinimas
- ✓ Matematika: pavyzdžių aiškinimas ir ryšių nustatymas, nežinomos reikšmės paieška
- ✓ Matematika: koordinatės
- ✓ Technologijos: duomenų pagrindimas

Gebėjimai

- ✓ Skaičiavimo
- ✓ Lyginių ir nelyginių skaičių atpažinimo

Amžius

- ✓ Nuo 7 metų

Priemonės

- ✓ 36 kortelės (geriausia magnetinės), kurių viena pusė nuspalsvinta
- ✓ Stalas (arba metalinė lenta) demonstracijai

Kiekvienai mokinių porai reikės:

- ✓ 36 vienodų kortelių, kurių viena pusė nuspalsvinta

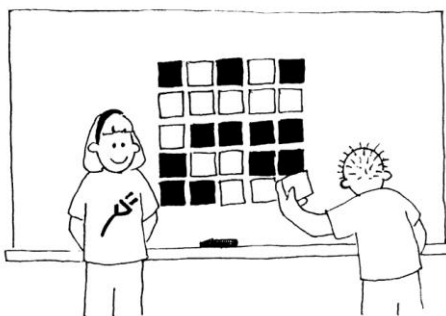


Magiškas triukas

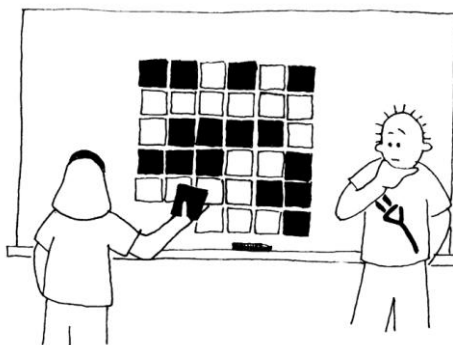
Demonstracija

Reikia 36 vienodų kortelių, kurių abi pusės yra skirtingų spalvų. (Galima iškirpti iš spalvoto popieriaus lapo, kurio viena pusė yra spalvota, o kita – balta.) Demonstracijai patogiu naudoti korteles su magnetais iš abiejų pusių, tik reikia, kad abi jų pusės dar būtų ir skirtingų spalvų. (Dažniausiai tokios kortelės turi magnetus iš vienos pusės. Dvi korteles galima suklijuoti puse, kurioje nėra magneto, tada ant vienos iš „magnetinių“ naujos kortelės pusių nupiešti baltą tašką ar tą pusę nuspalvinti.)

1. Mokinio paprašoma išdėlioti korteles kvadratu 5×5 atsitiktine puse.



Mokytojas išdėlioja dar vieną eilutę ir stulpelį, kad užduotis taptų šiek tiek sunkesnė.



Šios papildomos kortelės ir yra šio uždavinio – triuko pagrindas. Papildomas korteles reikia pridėti taip, kad kiekvienoje eilutėje ir stulpelyje būtų lyginis nuspalvintų kortelių skaičius.

2. Tegul mokinys vieną kortelę apverčia, kai mokytojas nežiūri. Apverstos kortelės eilutėje ir stulpelyje bus nelyginis nuspalvintų kortelių skaičius. Taip nesunkiai nustatoma, kuri kortelė buvo apversta. Ar mokiniai galėtų atspėti, kaip atliekamas šis triukas?

Mokiniai mokomi šio triuko:

1. Mokiniai susiskirsto poromis ir išdėlioja savo korteles ant stalo kvadratu 5×5 .
2. Po kiek nuspalvintų kortelių yra kiekvienoje eilutėje ir stulpelyje? Ar jų skaičius yra lyginis, ar nelyginis? Primenama, kad nulis yra lyginis skaičius.



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

3. Pridedama po vieną papildomą kortelę prie kiekvienos eilutės ir įsitikinama, kad kiekvienoje eilutėje nuspaltvintų kortelių yra nelyginis skaičius. Papildoma kortelė vadinama lyginumo kortele.
4. Taip pat pridedama dar viena kortelių eilutė taip, kad kiekviename stulpelyje būtų po nelyginį skaičių spalvotų kortelių.
5. Dabar viena kortelė apverčiama. Ką galima pastebėti apverstos kortelės stulpelyje ir eilutėje? (Nuspaltvintų kortelių skaičius – nelyginis.) Lyginumo kortelė parodo, kur buvo pakeista.
6. Kortelės kaitaliojamos toliau – šis triukas atliekamas iš naujo.

Papildomos veiklos:

1. Šį triuką galima išbandyti ir su kitais objektais. Pavyzdžiui, su loginio žaidimo kortelėmis (balta – juoda), monetomis (herbas ir skaičius) arba kortelėmis su 0 ir 1 skirtingose pusėse (siejama su dvejetainė sistema).
2. Kas nutiktų, jei būtų apverstos dvi ar daugiau kortelių? (Negalima visada tiksliai pasakyti, kurios dvi kortelės buvo apverstos, tačiau visada galima pasakyti, kad kai kas buvo pakeista. Galima aptarti, kas įvyksta, jei apverčiamos viena ar daugiau kortelių porų. Įmanoma, kad apvertus 4 korteles lyginumo kortelės neparodys klaidos.)
3. Galima išbandyti triuką su daugiau kortelių, pavyzdžiui, išdėstyti iš jų didelį kvadratą 9×9 . (Šis triukas gali būti daromas su bet kiek kortelių, be to, nebūtina išdėstyti kvadratu.)
4. Kitas įdomus pratimas – nagrinėti apatinę dešiniąją kortelę. Jei ši lyginimo kortelė bus teisinga stulpeliui virš jos, tai ar ši kortelė bus teisinga ir eilutei iš kairės? (Taip, visada, jei naudojamas nelyginis lyginumas.)
5. Šiame pratime lyginumas yra nelyginis – nelyginis nuspaltvintų kortelių skaičius eilutėse ir stulpeliuose. Ar galima tą patį padaryti su lyginiu lyginiu? (Tai įmanoma. Tačiau apatinės dešinėsios lyginumo kortelės pratimas tinka tik tada, kai yra lyginis (arba nelyginis) ir eilučių, ir stulpelių skaičius išdėstyme. (Pavyzdžiui, kortelių išdėstymams 5×9 arba 4×6 pratimą galima atlikti, o išdėstymui 4×7 pratimas netinka.)



Realaus gyvenimo pavyzdys gudručiams!

Remiantis panašia metodika, tikrinami knygų ir kitų prekių brūkšniniai kodai. Apžiūrėkite knygos ant užpakalinio viršelio išspausdintą 10 arba 13 skaitmenų kodą. Paskutinis skaitmuo yra kontrolinis skaitmuo, panašiai kaip anksčiau žaidžiant lyginumo kortelė.

Vadinasi, jei knygai suteiktas *tarptautinis standartinis knygos numeris* ISBN (angl. *International Standard Book Number*), tai galima patikrinti, ar nepadaryta klaida. Paprastai tikrinama kontrolinė suma.

Toliau aptariama, kaip tikrinama kontrolinė 10 skaitmenų suma. Pirmasis skaitmuo dauginamas iš 10, antrasis – iš 9, trečiasis – iš 8 ir t. t., devintasis dauginamas iš 2. Visos gautos sandaugos sudedamos. Pavyzdžiui, kodo ISBN 0–13–911991–4 suma yra

$$(0 \times 10) + (1 \times 9) + (3 \times 8) + (9 \times 7) + (1 \times 6) + (1 \times 5) + (9 \times 4) + (9 \times 3) + (1 \times 2) = 172.$$

Gautas skaičius padalijamas iš 11. Kokia liekana?

$$172 \div 11 = 15 \text{ liekana } 7$$

Jei liekana yra 0, tai kontrolinė suma yra 0. Kitu atveju iš 11 atimama liekana ir gaunama kontrolinė suma

$$11 - 7 = 4$$

Ar skaitmuo 4 yra ISBN knygos kodo paskutinis skaitmuo? Taip.

Tai reiškia, kad numeruojant knygas naudojamas ISBN.

Jei paskutinis skaitmuo būtų ne 4, tai būtų aišku, kad padaryta klaida.

Galima apskaičiuoti 10 skaitmenų kodo kontrolinę sumą, kuriai reikia daugiau kaip vieno skaitmens (iš 11 atėmus liekaną gaunama 10). Kai taip atsitinka, kontrolinio skaičiaus vietoje rašomas ženklas X.



„Rokiškio sūrio“ dėžės brūkšninis kodas

Kitas kontrolinio skaitmens pavyzdys yra bakalėjos gaminio brūkšninis kodas. Jis nustatomas kita formule (tokia pat formule tikrinami ir 13 skaitmenų knygų kodai). Jei brūkšninis kodas neteisingai suprantamas, tai paskutinis skaitmuo skiriasi nuo jo apskaičiuotos reikšmės. Kai taip nutinka, kasininkas dar kartą nuskaito kodą. Kontrolinis skaitmuo naudojamas banko sąskaitų, saugumo, mokesčių, traukinių numeriuose ir kituose vietose, kur žmonėms reikia įsitikinti, kad naudojamas numeris yra teisingas.



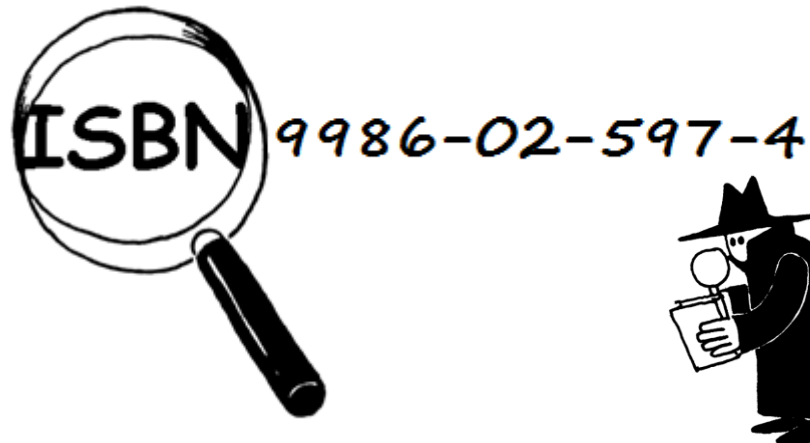
Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Patikrink šią knygą!

Detektyvai

Knygų tikrinimo paslaugos



Randame ir patikriname ISBN kontrolines sumas už nedidelį mokestį.

Prisijunk prie mūsų. Savo klasėje arba bibliotekoje suraskite knygų ir patikrinkite jų ISBN kodus.

Ar jų kontrolinės sumos teisingos?

Kartais būna klaidų.

Dažnai pasitaikančios klaidos:

- ✗ pakeista skaitmens reikšmė;
- ✗ du skaitmenys sukeisti vietomis;
- ✗ skaitmuo įterptas į numerį;
- ✗ skaitmuo praleistas numeryje.

Ar galėtumėte surasti knygą su X kontrolinei sumai 10? Neturėtų būti labai sunku, nes kiekviena 11 knyga jį turi.

Kokios klaidos nebūtų galima nustatyti? Ar galima pakeisti skaitmenį ir gauti teisingą kontrolinę sumą? Kas nutiks, jei du skaitmenys bus sukeisti vietomis (dažna skaitmenų rinkimo klaida)?



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Apie ką visa tai?

Tarkime, reikia įnešti 10 € į banko sąskaitą. Banko kasininkas įveda sumą ir siunčia duomenis į centrinį kompiuterį. Tačiau atsiranda trikdžių ir į centrinį kompiuterį vietoj kodo 10 € nusiunčiamas kodas 1000 €. Nors klientui tai džiaugsmas, bet bankui – tai didelė problema!

Svarbu nustatyti siunčiamų duomenų klaidas. Priimantis duomenis kompiuteris turi patikrinti, ar jie siunčiami nebuvo sugadinti kokių elektros trikdžių. Kartais klaidingai perduodamų duomenų siuntimas pakartojamas, tačiau kai kada tai neįmanoma, pavyzdžiui, kai diską sugadina magnetinė ar elektrinė spinduliuotė, karštis ar jis pažeidžiamas fiziškai arba kai duomenys gaunami iš kosmoso, labai ilgai reikia laukti retransliacijos, jei įvyksta klaida. (Signalas eina pusę valandos nuo Jupiterio iki Žemės, o jis yra arčiausiai Žemės.)

Reikia mokėti atpažinti sugadintus duomenis (*klaidos radimas*) ir atkurti originalius duomenis (*klaidos ištaisymas*).

„Magiško triuko“ žaidimo metodika naudojama ir kompiuteriuose. Bitai surašomi į įsivaizduojamus stulpelius ir eilutes, į kiekvieną eilutę ir stulpelį prirašomi papildomi lyginumo bitai. Tokiu būdu galima ne tik nustatyti klaidą, bet ir pasakyti, kur ji yra. Pažeistas bitas atkuriamas. Klaida ištaisoma.

Kompiuteriuose naudojamos daug sudėtingesnės klaidų kontroliavimo sistemos, kurios gali nustatyti ir ištaisyti sudėtingesnes klaidas. Dalis kompiuterio standžiojo disko yra skirta klaidoms taisyti, todėl diskas gali patikimai veikti net tada, jei dalis jo yra pažeista. Tam naudojama klaidų kontrolė pagrįsta lyginiu.



Sprendimai ir užuominos

ISBN 10 skaitmenų kodo kontrolinės sumos klaida yra nenustatoma, kai vienas skaitmuo padidėja, o kitas sumažėja. Tada sumos yra vienodos, o kodai skirtingi. Naudojant aprašytą skaičiavimo metodą, vargu ar tai kada nutiks. 13 skaitmenų ISBN sistemoje yra kitokio tipo nenustatomų klaidų: trys iš eilės einantys skaitmenys apverčiami. Tačiau dauguma dažniausių klaidų (vienas neteisingai surinktas skaitmuo arba vietomis sukeisti du skaitmenys) bus aptikta.



5 veikla

Dvidešimt spėjimų. Informacijos teorija

Santrauka

Kiek informacijos yra 1000 puslapių knygoje? Kur informacijos daugiau: 1000 puslapių telefonų knygoje, 1000 tuščių puslapių ar J. R. R. Tolentino (J. R. R. Tolkien) knygoje „Žiedų valdovas“? Jei galima išmatuoti informaciją, galima ir įvertinti, kiek vietos reikia šiai informacijai laikyti. Pavyzdžiui, ar galite perskaityti žemiau esantį sakinį?

Š s skns ntr bls.

Tikriausiai galite, nes balsės šiame sakinyje neturi daug informacijos. Ši veikla supažindina su informacijos turinio matavimo būdu.

Ryšiai su ugdymo programomis

- ✓ Matematika: skaičiai, skaičių palyginimas, rikiavimas
- ✓ Matematika: algebra, modeliai ir sekos
- ✓ Lietuvių kalba: rašyba, teksto elementų atpažinimas

Gebėjimai

- ✓ Skaičių lyginimo ir skaičių intervalų naudojimo
- ✓ Dedukcijos
- ✓ Klausinėjimo

Amžius

- ✓ Nuo 10 metų

Priemonės

- ✓ Pirmai veiklai atlikti papildomos priemonės nereikalingos

Papildomai veiklai atlikti kiekvienam mokiniui reikia:

- ✓ Darbo lapo „Sprendimų medžiai“



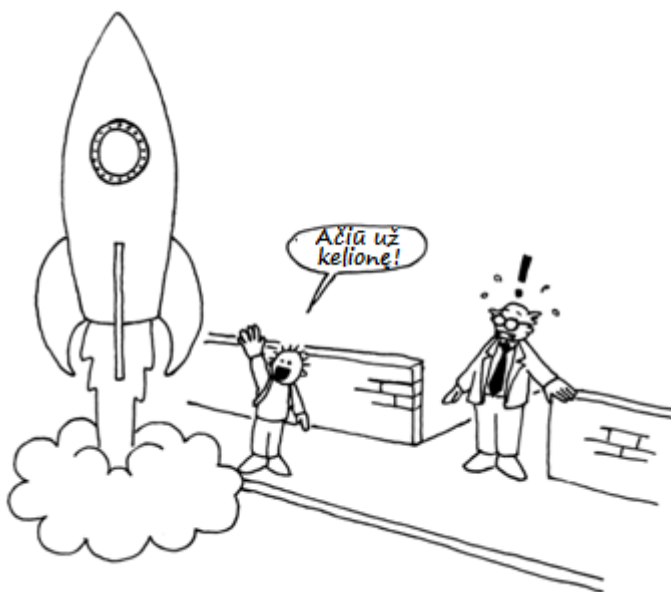
Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Dvidešimt spėjimų

Diskusija

1. Su mokiniais diskutuojama, kaip jie supranta, kas yra informacija.
2. Kaip galima pamatuoti, kiek informacijos yra knygoje? Ar puslapių, žodžių skaičius yra svarbus? Ar viena knyga gali turėti daugiau informacijos už kitą? Ar nuobodi knyga turi mažiau informacijos negu įdomi? Ar 400 puslapių knygoje, kurioje parašyta tik „cha, cha, cha...“, yra daugiau (mažiau) informacijos nei, pavyzdžiui, telefonų knygoje?
3. Paaiškinama, kad informacija informatikoje matuojama pagal pranešimo ar knygos netikėtumą. Sakykime, jūsų draugas, kuris kiekvieną dieną eina į mokyklą pėsčiomis, sako: „Aš atėjau į mokyklą pėsčiomis“. Jo pasakymas nesuteikia jums informacijos, nes jis nėra netikėtas. Tačiau, jei jis pasakytų: „Aš atskridau į mokyklą sraigtasparniu“, tai būtų netikėta ir jūs informacijos gautumėte daugiau.
4. Kaip įvertinti informacijos netikėtumą?
5. Vienas iš būdų yra parodyti, kaip sunku atspėti informaciją. Jei jūsų draugas pasakytų: „Atspėk, kaip aš šiandien atvykau į mokyklą“, tai pirmasis spėjimas tikriausiai būtų, kad jis atėjo pėsčiomis. Tačiau reiktų dar kelių spėjimų, jei jis atskrido sraigtasparniu, ar dar daugiau spėjimų, jei atskrido erdvėlaiviu.
6. Informacijos kiekis, kurį turi žinia, priklauso nuo to, kaip lengva ar sunku ją atspėti. Žemiau aprašytas žaidimas padės tai išsiaiškinti.



Dvidešimt klausimų

Tikriausiai dauguma žino 20 klausimų žaidimą: pasirinktam mokiniui kiti mokiniai užduoda klausimus, į kuriuos galima atsakyti tik „taip“ arba „ne“, tol, kol atspėja to mokinio sugalvotą atsakymą.

Pasiūlymai:

Sugalvojamas:

- ✓ skaičius nuo 1 iki 100;
- ✓ skaičius nuo 1 iki 1000;
- ✓ skaičius nuo 1 iki 1000000;
- ✓ sveikasis skaičius;
- ✓ 6 skaičių seka pagal kokią nors taisyklę (pvz., lyginiai skaičiai, skaičių kvadratai ir pan.). Sekos nariai spėliojami iš eilės nuo pirmojo iki paskutinio (pvz., 2, 4, 6, 8, 10).

Suskaiciuojami užduoti klausimai. Gautas skaičius yra informacijos matavimo vertė.

Diskusija

Kokie būdai buvo naudojami? Kuris buvo geriausias?

Atkreipiamas dėmesys, kad skaičiui nuo 1 iki 100 atspėti užtenka 7 klausimų. Pavyzdžiui:

Ar skaičius mažesnis už 50? Taip.
Ar skaičius mažesnis už 25? Ne.
Ar skaičius mažesnis už 37? Ne.
Ar skaičius mažesnis už 43? Taip.
Ar skaičius mažesnis už 40? Ne.
Ar skaičius mažesnis už 41? Ne.
Tai 42! Taip!

Įdomu, jei intervalas padidinamas iki 1000, prireikia tik trijų papildomų klausimų. Kiekvieną kartą intervalą padidinus dukart, reikia vieno papildomo klausimo.

Mokiniams galima pasiūlyti sužaisti stalo žaidimą – „Kodo laužymas“ (angl. „Mastermind“).

Gudručiams

Kiek informacijos turi žinia?

Pateiktas spėjimo būdas tinkamas ne tik skaičiams atspėti. Galima spėti, kuri raidė labiau tikėtina žodyje arba sakinyje.

Siūloma pabandyti spėjimo žaidimą, paėmus trumpą 4–6 žodžių sakinį. Raidė turi būti spėjama iš eilės nuo pirmos iki paskutinės pagal abėcėlę. Vienas mokinyss galėtų surašyti raides abėcėlės tvarka. Galima įrašyti užduodamus klausimus. Klausimai gali būti tik tokie, kurių atsakymas „taip“ arba „ne“. Pavyzdžiui, „Ar tai raidė a?“, „Ar tai balsė?“, „Ar raidė yra prieš m abėcėlėje?“ Tarpas tarp žodžių taip pat laikomas raide ir turi būti spėjamas. Perklausomas įrašas ir nustatoma, kurią žinios dalį buvo lengviau atspėti.



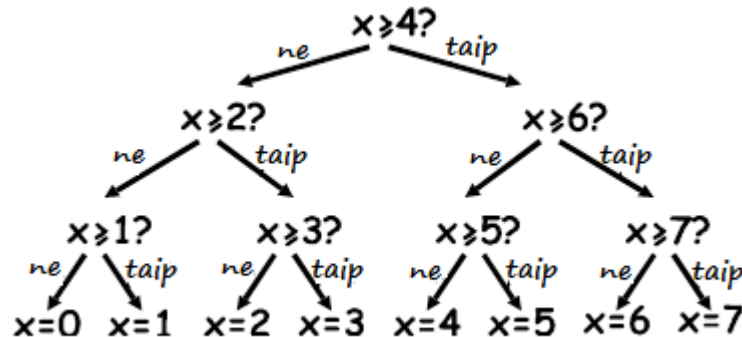
Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Darbo lapas *Pasirinkčių medžiai*

Išsiaiškinome klausinėjimo strategiją, todėl esame pasiruošę perduoti žinią neklausinėdami.

Pateikta diagrama „Pasirinkčių medis“ vaizduoja skaičiaus nuo 0 iki 7 spėjimą.



Nurodykite atsakymų „taip“ ir „ne“ kelią, kai spėjate skaičių 5.

Kiek pasirinkčių „taip“ ir „ne“ reikia norint atspėti bet kurį skaičių?

Atkreipiame dėmesį į kai ką įdomaus. Aukščiau pavyzdyje po skaičiais 0, 1, 2, 3, ... paskutinėje eilėje užrašykite jų reikšmes dvejetainė sistema (žr. 1 veiklą).

Pažvelkite giliau į medį. Pakeiskite medyje pasirinktį „ne“ į 0, o „taip“ – į 1. Ką matote?

Žaidžiant skaičių spėjimo žaidimą, klausimai parenkami taip, kad pasirinkčių seka vaizduotų spėjamą skaičių dvejetainė sistema.

Nubraižykite pasirinkčių medį, kuris vaizduotų skaičiaus nuo 0 iki 15 spėjimą.

Gudručiams. Kokį medį naudotumėte, spėdami kieno nors amžių?
Kokiu medžiu pavaizduotumėte sakinio raidžių spėjimo medį?



Apie ką visa tai?

Žymus amerikiečių matematikas (buvęs fokusininkas, taip pat važinėjęs vienračiu) Klodas Šenonas (Claude Shannon) atliko daug šio žaidimo eksperimentų. Jis išmatavo informacijos kiekį bitais – kiekvienas atsakymas „taip“ arba „ne“ yra vienas bitas. K. Šenonas atrado, kad informacijos kiekis žinioje priklauso nuo to, kas žinoma. Kartais gali būti užduodamas klausimas, kuris labai sumažina toliau užduodamų klausimų skaičių. Tokiu atveju informacijos kiekis yra mažesnis. Pavyzdžiui, informacijos, kokia puse atsivers viena metama moneta, kiekis, yra vienas bitas: herbas arba skaičius. Bet jei moneta netaisyklinga (deformuota) ir herbas atsivers devynis kartus iš dešimties, tai informacijos kiekis mažesnis kaip vienas bitas. Kaip nustatyti, ar monetos atsivertusioji pusė išsiaiškinama užduodant vieną klausimą, į kurį atsakoma tik „taip“ ir „ne“? Tiesiog galima klausti: „Ar metus monetą du kartus abu kartus atsivers herbas?“ Į 80 % tokių klausimų, jei metama netaisyklinga moneta, atsakymas būtų „taip“ ir tik 20 % klausimų atsakymas būtų „ne“. Vidutiniškai užduodamas mažiausiai vienas klausimas apie vieną monetos metimą.

K. Šenonas įvedė informacijos turinio matavimo terminą – „entropija“. Entropija priklauso ne tik nuo galimų atsakymų skaičiaus, bet ir nuo įvykio tikimybės. Neįtikėtinam arba netikėtam įvykiui išsiaiškinti užduodama daugiau klausimų, nes tokiu būdu gaunama daugiau informacijos.

Informatikams žinios entropija yra labai svarbi. Negalima suspausti žinios, kad užimtų mažiau vietos nei jos entropija, o geriausia pakavimo programa yra kaip spėjimo žaidimas. Kadangi kompiuterio programa „spėja“, o atsakymai koduojami bitais, klausimų perskaičiavimas gali būti atliktas vėliau, atkuriant informaciją. Geriausia pakavimo programa gali sumažinti tekstinio failo dydį – tai sutaupo gana daug kompiuterio atminties!

Spėjimo metodas taikomas ir kuriant kompiuterio programų vartotojo sąsają, kai spėjama, kokius ženklus vartotojas ruošiasi toliau rinkti. Tai gali būti naudinga neįgaliesiems, kuriems sunku rinkti ženklus klaviatūra. Kompiuteris spėja, ką žmogus ruošiasi rinkti, tad šiam belieka sutikti arba atmesti siūlymus. Kompiuteriui užtenka tik vieno atsakymo „taip“ arba „ne“ kiekvienam ženklui ir tai gali būti didelė pagalba sutrikusios koordinacijos ar judėjimo žmonėms, kuriems sunku naudotis kompiuterio klaviatūra ar pele. Ši teksto rinkimo sistema naudojama ir kai kuriuose liečiamuosiuose telefonuose.



Sprendimai ir užuominos

Vieno „taip“ arba „ne“ klausimo atsakymas atitinka lygiai vieną bitą ir nepriklauso nuo klausimo sudėtingumo. Pavyzdžiui, gali būti užduodamas paprastas klausimas: „Ar skaičius yra didesnis už 50?“ arba sudėtingesnis: „Ar skaičius yra nuo 20 iki 60?“, atsakymas vis tiek yra vienas bitas.

Jei skaičiaus spėjimo žaidime klausimai pasirenkami tam tikra tvarka, atsakymų seka yra spėjamo skaičiaus reikšmė dvejetainėje sistemoje. Trys dvejetainėje sistemoje yra 110, atsakymų apie šį skaičių seka skaičiaus spėjimo žaidime yra „ne, taip, taip“ arba „0, 1, 1“, jei „ne“ žymimas 0, o „taip“ – 1.

Raidės spėjimas sakinyje gali priklausyti nuo prieš ją esančios raidės.



II dalis

Kaip veikia kompiuteris?

Algoritmai



Kaip veikia kompiuteris?

Kompiuteris gali būti puikus žmogaus pagalbininkas, kai reikia kaupti, laikyti, saugoti, apdoroti, perduoti informaciją ar jos ieškoti. Kompiuteriu apdorojamą informaciją įprasta vadinti duomenimis. Veiksmai su jais užrašomi algoritmais. Kompiuteriu apdorojama daug informacijos. Siekiant kuo didesnės spartos, reikia sukurti tokius algoritmus, kurie padėtų iš informacijos gausos kuo greičiau rasti tam tikrą informaciją ir ją perduoti tinklu.

Algoritmas – tai aiškūs ir tikslūs nurodymai, kaip ir kokius veiksmus reikia atlikti siekiant užsibrėžto tikslo arba išspręsti suformuluotą uždavinį. Informatikoje algoritmo sąvoka yra esminė. Algoritmu kompiuteriui nurodoma, kaip spręsti problemą. Vieni algoritmai yra spartesni už kitus, todėl jais išspręsti problemas galima daug greičiau nei anksčiau. Pavyzdžiui, surasti skaičiaus π (pi) milijoną skaitmenų po kablelio, visus tinklalapius, kuriuose minimas tam tikras žodis, talpiausią į konteinerį sudėjimo būdą, didžiausią pirminį skaičių ir kt.

Algoritmo sąvoka atsirado labai seniai, daugiau nei prieš tūkstantį metų. Pats žodis „algoritmas“ kilo iš IX a. arabų matematiko ir astronomo Mohamedo ibn Musos al Chorezmio lotyniškais rašmenimis parašyto vardo *Algorithmi*. Šis mokslininkas sudarė keturių aritmetikos veiksmų dešimtainiais skaičiais taisyklės, kurias Europoje imta vadinti „algorizmais“. Vėliau šis žodis buvo pakeistas „algoritmu“, jam suteikta platesnė prasmė – juo pradėtos vadinti įvairių skaičiavimų (ne tik aritmetinių) taisyklės.



6 veikla

Laivų mūšis. Paieškos algoritmai

Santrauka

Dažnai kompiuteris turi surasti tam tikrą informaciją iš didelio jos kiekio. Tuo tikslu turi būti sukurtas greitas ir efektyvus būdas. Čia pateikiami trys skirtingi paieškos metodai: *nuoseklioji paieška*, *dvejetainė paieška* ir *maiša* (angl. *hashing*).

Ryšiai su ugdymo programomis

- ✓ Matematika: skaičiai ir skaičiavimai, skaičių palyginimas
- ✓ Matematika: sąryšiai ir funkcijos, koordinatės
- ✓ Informatika: algoritmai

Gebėjimai

- ✓ Loginio argumentavimo

Amžius

- ✓ Nuo 9 metų

Priemonės

Kiekvienam mokiniui reikės:

- ✓ Darbo lapų laivų mūšio žaidimams
 - 1A, 1B pirmajam žaidimui
 - 2A, 2B antrajam žaidimui
 - 3A, 3B trečiajam žaidimui
- ✓ Papildomi žaidimų lapai: 1A', 1B', 2A', 2B', 3A', 3B'



Laivų mūšis

Įvadas

1. Pasirenkama 15 mokinių, kurie sustoja viena eile prieš klasę. Kiekvienam mokiniui duodama kortelė su skaičiumi (atsitiktine tvarka). Skaičiai nerodomi likusiems mokiniams. Pasakoma, į kokį intervalą patenka turimi skaičiai.
2. Kitiems mokiniams duodama po dėžutę su keturiais ar penkiais saldainiais. Mokiniai turi surasti duotą skaičių. Jie gali „sumokėti“ saldainiais ir pažiūrėti į tam tikrą kortelę. Jei jie suranda teisingą skaičių anksčiau, nei išdalija savo saldainius, tai likusius saldainius gali pasilikti.
3. Kai žaidimas baigiamas, jei reikia, jį galima sužaisti dar kartą.
4. Kortelės išmaišomos ir vėl išdalijamos mokiniams. Dabar mokiniai sustoja į eilę jų kortelėse esančių skaičių didėjimo tvarka. Kartojama paieška, aprašyta antrajame punkte.

Jei skaičiai surikiuoti, mokiniai turėtų suvokti geriausią veikimo būdą – už vieną saldainį gali atsikratyti pusės mokinių, pasirinkę atversti vidurinio mokinio kortelę. Kartodami šį veiksmą jie gali išsiaiškinti ieškomą skaičių „sumokėję“ tris saldainius. Šio metodo efektyvumas akivaizdus.

Veikla

Mokiniai gali suvokti, kaip kompiuteris ieško informacijos, žaisdami laivų mūšio žaidimą. Žaidžiant šį žaidimą mokinių paprašoma apgalvoti metodus, kuriais galima tiksliau nustatyti laivo vietą.



Laivų mūšis. Nuoseklioji paieška

Instrukcija

1. Mokiniai susiskirsto poromis. Vienas paima 1A lapą, kitas – 1B. Savo lapų mokiniai vienas kitam nerodo!
2. Abu mokiniai apveda vieną laivą viršutinėje savo lapo kortelėje ir pasako jo numerį vienas kitam.
3. Vienas po kito abu mokiniai spėja, kur yra apibrauktas porininko laivas: sako bet kurio laivo raidę, o porininkas turi pasakyti to laivo numerį.
4. Mokiniai turi nustatyti, kiek reikia šūvių norint pataikyti į porininko pasirinktą laivą. Šūvių skaičius atitinka žaidimo taškus.

(Papildomi lapai 1A' ir 1B' skirti mokiniams, norintiems žaisti dar kartą arba iškart atspėjusiems porininko laivo vietą. Lapai 2A', 2B' ir 3A', 3B' skirti kitiems žaidimams.)

Diskusija

1. Kokie rezultatai?
2. Koks galimas mažiausias ir didžiausias šūvių skaičius? (Mokiniai turėtų nurodyti atitinkamai 1 ir 26. Daroma prielaida, kad mokiniai nešaus į tą patį laivą du kartus.)



Laivų mūšis. Dvejtainė paieška

Instrukcija

1. Mokiniai susiskirsto poromis. Vienas paima 1A lapą, kitas – 1B. Savo lapų mokiniai vienas kitam nerodo!
2. Abu mokiniai apveda vieną laivą viršutinėje savo lapo kortelėje ir pasako jo numerį vienas kitam.
3. Vienas po kito abu mokiniai spėja, kur yra apibrauktas porininko laivas: sako bet kurio laivo raidę, o porininkas turi pasakyti to laivo numerį.
4. Mokiniai turi nustatyti, kiek reikia šūvių norint pataikyti į partnerio pasirinktą laivą. Šūvių skaičius atitinka žaidimo taškus.

(Papildomi lapai 2A' ir 2B' skirti mokiniams, norintiems žaisti dar kartą arba iškart atspėjusiems porininko laivo vietą. Lapai 3A', 3B' skirti kitam žaidimui.)

Diskusija

1. Kokie rezultatai?
2. Kokiu metodu buvo surinkta mažiausiai taškų?
3. Į kurį laivą šaunama pirmiausia? (Šūvis į vidurinį sutrumpina paiešką perpus.) Į kurį laivą šaunama antrą kartą? (Geriausia šauti į vidurinį laivą ieškomame intervale.)
4. Jei buvo veikiamas šiuo metodu, kiek prireikė šūvių pataikyti į laivą? (Daugiausia 5 šūviai.)

Šis metodas vadinamas *dvejtaine paieška*, nes intervalai visada dalijami į dvi dalis.



Laivų mūšis. Maiša

Instrukcija

1. Mokiniai susiskirsto poromis. Vienas paima 3A lapą, kitas – 3B. Abu vienas kitam pasako savo pasirinkto laivo numerį.
2. Žaidžiant šį žaidimą reikia nustatyti, kuriame iš dešimties stulpelių (nuo 0 iki 9) yra laivas. Tuo tikslu sudedami laivo numerio skaitmenys. Gautos sumos vienetų skaitmuo parodo, kuriame stulpelyje yra laivas. Pavyzdžiui, laivo numeris yra 2345, tai $2+3+4+5$ yra 14. Vienetų skaitmuo yra 4, taigi laivas yra ketvirtajame stulpelyje. Nustačius laivo stulpelį belieka atspėti laivą. Šis metodas vadinamas *maiša*, nes skaitmenys yra „sugrūsti“ drauge.
3. Dabar galima pabandyti žaisti šiuo metodu. Žaidimą galima pakartoti pasirenkant laivą iš kito stulpelio.

(Gali būti naudojami tik papildomi lapai 3A' ir 3B', nes laivų išdėstymas stulpeliuose turi būti vienodas abiejuose lapuose.)

Diskusija

1. Kokie rezultatai?
2. Kokiu metodu buvo surinkta mažiausiai taškų?
3. Kuriuos laivus buvo lengviausia nušauti? (Tuos, kurie yra vieninteliai stulpelyje.) Kuriuos sunkiausia? (Tuos, kurių stulpeliuose yra daugiausia laivų.)
4. Kuris iš paieškos metodų greičiausias? Kodėl?
5. Kuo vienas paieškos metodas geresnis už kitą? (Antrasis metodas yra greitesnis už pirmąjį, bet pasirinkus pirmąjį nereikia išrikiuoti laivų. Trečiasis metodas paprastai yra greičiausias, bet netinkamai pasirinkus pirmąjį laivą, paieška šiuo metodu gali būti labai lėta.



Papildomos veiklos

1. Mokiniai gali sugalvoti žaidimų patys. Žaisdami antrąjį žaidimą jie turėjo skaičius išdėlioti didėjančiai. Mokiniai galėtų pamąstyti, kaip būtų galima pasunkinti trečią laivų mūšio žaidimą. (Sunkiausia žaisti, kai visi laivai yra vienoje eilėje.) Kaip būtų galima pakeisti trečią žaidimą? (Reikėtų sudėti po vienodą skaičių laivų kiekviename stulpelyje.)
2. Kas nutiktų, jei ieškomo laivo lape nebūtų? (Atlikdami nuosekliają paiešką tu įsitikintume padarę 26 žingsnius. Jei paieška dvejetainė, reikėtų tik 5 žingsnių. Jei naudojamosi *maiša*, žingsnių skaičius priklauso nuo laivų skaičiaus kiekviename stulpelyje.)
3. Kiek dvejetainės paieškos žingsnių prireiktų laivo vietai nustatyti, jei būtų 100 laivų (6 žingsnių), 1000 laivų (apie 9) ar milijonas laivų (apie 19)? (Žingsnių skaičius didėja labai lėtai palyginti su laivų skaičiumi. Kai laivų skaičius padidėja dvigubai, reikalingas papildomas žingsnis, todėl žingsnių skaičius yra proporcingas laivų skaičiaus logaritmui.)



Mano laivai													Šūvių skaičius:												
9058		7169		3214		5891		4917		2767		4715		674		8088		1790		8949		13		3014	
A		B		C		D		E		F		G		H		I		J		K		L		M	
8311		7621		3542		9264		450		8562		4191		4932		9462		8423		5063		6221		2244	
N		O		P		Q		R		S		T		U		V		W		X		Y		Z	

Priešininko laivai													Šūvių skaičius:												
A	B	C	D	E	F	G	H	I	J	K	L	M													
N	O	P	Q	R	S	T	U	V	W	X	Y	Z													

C



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Mano laivai				Šūvių skaičius:								
1630	9263	4127	405	4429	7113	3176	4015	7976	88	3465	1571	8625
A	B	C	D	E	F	G	H	I	J	K	L	M
2587	7187	5258	8020	1919	141	4414	3056	9118	717	7021	3076	3336
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Priešininko laivai													Šūvių skaičius:												
A B C D E F G H I J K L M																									
N O P Q R S T U V W X Y Z																									

1B



Mano laivai													Šūvių skaičius:																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
163	445	622	1410	1704	2169	2680	2713	2734	3972	4208	4871	5031																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	</

Priešininko laivai													Šūvių skaičius:												
A B C D E F G H I J K L M																									
N O P Q R S T U V W X Y Z																									

Mano laivai													Šūvių skaičius:												
33	183	730	911	1927	1943	2200	2215	3451	3519	4055	5548	5655													
A	B	C	D	E	F	G	H	I	J	K	L	M													
5785	5897	5905	6118	6296	6625	6771	6831	7151	7806	8077	9024	9328													
N	O	P	Q	R	S	T	U	V	W	X	Y	Z													

Priešininko laivai													Šūvių skaičius:												
A B C D E F G H I J K L M																									
N O P Q R S T U V W X Y Z																									

2B

Mano laivai				Šūvių skaičius:					
0	1	2	3	4	5	6	7	8	9
A 9047	C 3080		E 5125	H 8051	L 7116	O 6000	R 9891		W 1062
B 1829	D 9994		F 1480	I 1481	M 8944	P 7432	S 1989	V 4392	X 2106
			G 8212	J 4712	N 4128	Q 4110	T 2050		Y 5842
				K 6422			U 8199		Z 7057

Priešininko laivai				Šūvių skaičius:					
0	1	2	3	4	5	6	7	8	9
A	E	H	K	L		O	R	V	Y
B	F	I		M		P	S	W	Z
C	G	J		N		Q	T	X	
D							U		

3A



Šūvių skaičius:										Mano laivai													
0	A 9308	B 1478	C 8417	D 9434	1	E 6519	F 2469	G 5105	2	H 1524	I 8112	J 2000	3	K 4135	4	L 9050	M 1265	N 5711	5	6	7	8	9

Priešininkų laivai										Šūvių skaičius:										
0		1		2		3		4		5		6		7		8		9		
A		C						E		H		L		O		R		W		
B		D						F		I		M		P		S		X		
						G			K		N		Q		T		V		Y	
																			Z	

3B



Mano laivai												Šūvių skaičius:													
6123	1519	9024	5164	2038	2142	7156	9974	9375	7104	1004	1023	5108	A	B	C	D	E	F	G	H	I	J	K	L	M
1884	3541	5251	4840	3289	3654	2480	5602	8965	4053	2405	2304	1959	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Priešininko laivai													Šūvių skaičius:																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
A	B	C	D	E	F	G	H	I	J	K	L	M																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									</

1A'

Mano laivai		Šūvių skaičius:											
													
2387	9003	3951	5695	1284	4761	7118	1196	1741	3791	3405	3132	6682	
A	B	C	D	E	F	G	H	I	J	K	L	M	
													
9493	9864	7359	1250	7036	2916	7562	9299	8910	6713	5173	8617	4222	
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	

Priešininko laivai													Šūvių skaičius:												
A B C D E F G H I J K L M																									
N O P Q R S T U V W X Y Z																									

1B'

Mano laivai													Šūvių skaičius:												
	28		326		943		1321		1896		2346		2430		2929		3106		3417		4128		4717		4915
A	B	C	D	E	F	G	H	I	J	K	L	M													
	5123		5615		6100		7015		7120		7695		7812		8103		8719		9020		9608		9713		9911
N	O	P	Q	R	S	T	U	V	W	X	Y	Z													

Priešininko laivai													Šūvių skaičius:																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												

2A'

Mano laivai													Šūvių skaičius:												
	56		194		306		1024		1510		1807		2500		2812		3011		3902		4178		5902		5915
A	B	C	D	E	F	G	H	I	J	K	L	M													
	6102		6526		6818		7020		7155		7913		8016		8230		8599		8902		9090		9526		9812
N	O	P	Q	R	S	T	U	V	W	X	Y	Z													

Priešininko laivai													Šūvių skaičius																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
A	B	C	D	E	F	G	H	I	J	K	L	M																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		

2B'



Mano laivai										Šūvių skaičius:									
0	1	2	3	4	5	6	7	8	9										
A 1982	C 6113		E 9121	H 5009	L 1248	O 2004	R 9369	V 3285	W 9172										
B 7841	D 1055		F 1011	I 2651	M 1716	P 5173	S 1321		X 2052										
			G 2984	J 1751	N 2148	Q 2806	T 3004		Y 6012										
				K 4848			U 7190		Z 7525										

Priešininko laivai										Šūvių skaičius:									
0	1	2	3	4	5	6	7	8	9										
A	E	H		L		O	R	V	Y										
B	F	I		M		P	S	W	Z										
C	G	J	K	N		Q	T	X											
D							U												



Šūvių skaičius:									
Mano laivai									
0	1	2	3	4	5	6	7	8	9
A 8615 B 7003 C 1991 D 6211	E 1361 F 7644 G 5600	H 7726 I 9003 J 5557	K 3000	L 1814 M 2002 N 8844		O 9656 P 4002 Q 1221	R 6993 S 3121 T 4300 U 1907	V 8208 W 9423 X 4176	Y 2917 Z 4122

Šūvių skaičius:									
Priešininko laivai									
0	1	2	3	4	5	6	7	8	9
A B	C D		E F G	H I J K	L M N	O P Q	R S T U	V	W X Y Z

3B'

Apie ką visa tai?

Kompiuteriuose laikoma labai daug informacijos, ji turi būti greitai apdorojama. Daugiausia paieškos sunkumų kyla dėl būtinos didelės spartos: interneto paieškos sistemomis paieška atliekama tarp milijardo tinklalapių per sekundės dalį. Duomuo (pvz., žodis, kodas ar vardas), pagal kurį ieškoma informacijos, vadinamas reikšminiu žodžiu.

Kompiuteris apdoroja informaciją labai greitai. Galima pamanyti, kad jis paiešką pradeda nuo pirmojo elemento ir tikrina visus elementus iš eilės, kol randa ieškomą. Tokia *nuoseklioji paieška* buvo atliekama viename iš žaidimų, tačiau šis paieškos metodas yra labai lėtas – net ir kompiuteriui. Pavyzdžiui, prekybos centre yra 10000 skirtingų produktų, išdėliotų skirtingose lentynose. Kai nuskaitomas prekės kodas, kompiuteris turi patikrinti 10000 kodų, kol randa reikiamą prekę ir jos kainą. Jei vieną prekę kompiuteris tikrintų vieną tūkstantąją sekundės, tai užtruktų 10 sekundžių. Kiek užtruktų surasti visas kieno nors perkamas prekes!

Dvejtainė paieška yra greitesnis metodas. Šiuo metodu skaičiai rikiuojami iš eilės didėjančiai. Tikrinant vidurinį sąrašo elementą nustatoma, kurioje pusėje yra ieškomas skaičius. Šis veiksmas kartojamas, kol surandamas ieškomas skaičius. Prisiminus pavyzdį apie prekybos centrą surasti prekės kainą užtruktų dvi šimtasias sekundes, o tai sunkiai pastebima.

Trečiasis duomenų paieškos metodas vadinamas *maiša*. Ši paieška remiasi raktu, sudaromu taip, kad reikiama informacija būtų surasta tiksliai iškart. Pavyzdžiui, jei paieškos raktas yra telefono numeris, galima sudėti visus skaitmenis, imti gautos sumos dalybos iš 11 liekaną. Šiuo atveju maišos raktas šiek tiek panašus į svarstyklės, kurios buvo aprašytos 4 veiklos skyriuje – nedidelė duomenų dalis, priklausoma nuo kitų apdorotų duomenų. Maža tikimybė, kad keli maišos raktai atsidurs toje pačioje vietoje ir tada kompiuteris turės atlikti paiešką pagal kiekvieną iš jų, kol suras ieškomą skaičių.

Programuotojai naudoja maišos metodą, kai ieškant duomenų eilė negali būti suardoma ar lėta paieška nėra tikėtina.



7 veikla

Lengviausias ir sunkiausias. Rikiavimo algoritmai

Santrauka

Dažnai kompiuteriui reikia išrikiuoti sąrašą. Pavyzdžiui, išrikiuoti vardus pagal abėcėlę, elektroninius laiškus pagal datą ar kt. Išrikiuotame sąraše galima greičiau pamatyti ir surasti ieškomą informaciją. Išrikiavus mokinių pažymius, aiškiai matomi patys žemiausi ir patys aukščiausi įvertinimai.

Kai kurie rikiavimo metodai yra neefektyvūs net ir atliekant juos kompiuteriu. Žinoma, yra atrasti keli greitesni rikiavimo metodai. Šiame skyriuje mokiniai sužinos apie skirtingus rikiavimo metodus, taip pat kaip pasirinktas „gudrus“ rikiavimo metodas paspartina rikiavimą.

Ryšiai su ugdymo programomis

- ✓ Matematika: matai ir matavimai, praktinis svorio matavimas
- ✓ Informatika: algoritmai

Gebėjimai

- ✓ Svarstyklių naudojimo
- ✓ Rikiavimo
- ✓ Lyginimo

Amžius

- ✓ Nuo 8 metų

Priemonės

Kiekvienai mokinių grupei reikės:

- ✓ 8 vienodo dydžio, bet skirtingo svorio pakuočių (gali būti pakuotės nuo pieno, nepermatomi plastikiniai buteliukai nuo gėrimo ar skardinės dėžutės, pripildytos smėlio ar vandens)
- ✓ Svirtinių svarstyklių
- ✓ Darbo lapo „Svorių rikiavimas“
- ✓ Darbo lapo „Skaldyk ir valdyk!“



Lengviausias ir sunkiausias

Diskusija

Kompiuteriui dažnai reikia išrikiuoti duomenis. Kas nutiktų, jei duomenys būtų neišrikiuoti?

Paprastai kompiuteris gali palyginti dvi reikšmes vienu metu. Atlikdami užduotis mokiniai supras, kaip kompiuteris tai daro.

Veikla

1. Mokiniai suskirstomi į grupes.
2. Kiekvienai grupei duodamas Darbo lapas „Svorių rikiavimas“, sveriamų pakuočių rinkiniai ir svirtinės svarstyklės.
3. Mokiniai atlieka šią veiklą ir tada aptaria rezultatus.



Darbo lapas *Svorių rikiavimas*

Tikslas: surasti geriausią nežinomų svorių rikiavimo metodą.

Reikės: smėlio arba vandens, 8 vienodų pakuočių, svirtinių svarstyklių.

Ką daryti:

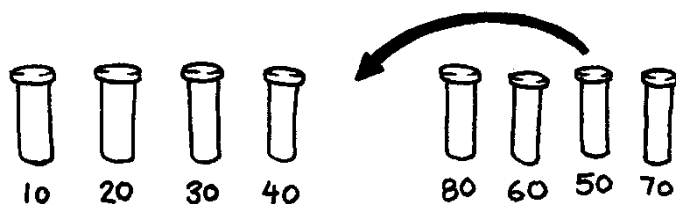
1. Į kiekvieną pakelį įpilkite skirtingą kiekį smėlio arba vandens. Sandariai uždarykite.
2. Sumaišykite pakuotes, kad nežinotumėte, kiek juose smėlio ar vandens.
3. Suraskite lengviausią pakuotę. Kokiu paprasčiausiu būdu tai galima padaryti?

Patarimai: Naudokite svirtines svarstykles. Vienu metu galite jomis palyginti tik du svorius.

4. Atsitiktinai pasirinkite 3 pripildytas pakuotes. Išrikiuokite jas nuo lengviausios iki sunkiausios pasverdami svirtinėmis svarstyklėmis. Kaip tai atlikote? Kiek kartų mažiausiai reikėjo sverti? Kodėl?
5. Išrikiuokite visas 8 pripildytas pakuotes nuo lengviausios iki sunkiausios.
6. Tada patikrinkite, ar gerai išrikiavote, lygindami dviejų gretimų pakuočių svorį svirtinėmis svarstyklėmis.

Išrenkamojo rikiavimo algoritmas

Vienas iš kompiuterio naudojamų rikiavimo algoritmų yra *išrenkamojo rikiavimo algoritmas*. Pirma, surandamas mažiausias svoris ir atidedamas į šoną. Antra, surandamas mažiausias svoris iš likusiųjų ir atidedamas į šoną. Tai kartojama, kol visi svoriai išrikiuojami.



Suskaičiuokite, kiek kartų lyginote svorius.

Gudručiams: Parodykite, kaip matematiškai galima suskaičiuoti, kiek kartų reikia lyginti 8 objektus norint juos išrikiuoti pagal svorį. Kiek kartų norint išrikiuoti reikia lyginti 9 objektus? 20?



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Darbo lapas *Skaldyk ir valdyk!*

Greitojo rikiavimo algoritmas

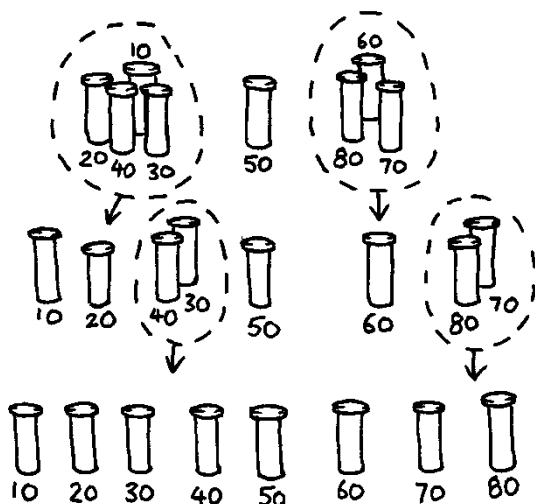
Greitojo rikiavimo algoritmas yra daug greitesnis ir paprastesnis už *išrenkamojo rikiavimo algoritmą*, todėl yra labai dažnai naudojamas.

Atsitiktinai pasirinkite vieną iš objektų ir padėkite jį ant svarstyklių lėkštelės.

Visus likusius objektus palyginkite su atsitiktinai pasirinktu objektu. Lengvesnius už jį padėkite kairėje svarstyklių pusėje, o sunkesnius – dešinėje. Objektą, su kuriuo lyginate kitus objektus, padėkite viduryje tarp grupių. (Baigus sverti visus objektus gali būti, kad vienoje pusėje bus daug daugiau objektų nei kitoje.)

Pasirinkite vieną iš atrinktų objektų grupių ir pakartokite tą patį: atsitiktinai iš grupės pasirinkite objektą ir palyginkite visus likusius objektus su juo. Tą patį padarykite ir su kita objektų grupe. Objektą, su kuriuo lyginate kitų objektų svorį, padėkite viduryje tarp sudarytų naujų dviejų grupių.

Kartokite šiuos veiksmus, kol grupėse liks tik po vieną objektą. Kadangi visose objektų grupėse liko tik po vieną objektą, vadinasi, visi objektai yra išrikiuoti nuo lengviausio iki sunkiausio.



Kiek kartų reikėjo lyginti objektus naudojant šį algoritmą?

Greitasis rikiavimas yra daug efektyvesnis už *išrenkamojo rikiavimo algoritmą*, jei iš pradžių pavyksta pasirinkti patį lengviausią arba patį sunkiausią objektą. Jei pasirenkamas vidutinio svorio objektas, užtenka 14 kartų palyginti visus objektus ir taip juos išrikiuoti nuo lengviausio iki sunkiausio. Padaryti tą patį *išrenkamojo rikiavimo algoritmu* prireiktų 28 lyginimo kartų.

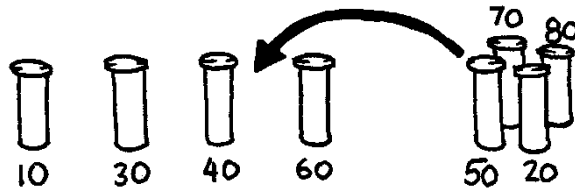
Gudručiams: Kiek kartų greitojo rikiavimo algoritmu reikėtų lyginti objektus, jei visada atsitiktinai pasirinktumėte lengviausią objektą?



Kiti rikiavimo metodai

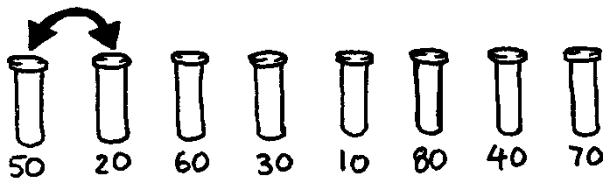
Sugalvota daug rikiavimo metodų. Pabandykite išrikiuoti savo objektus įvairiais metodais.

Pagrindinis *įterpiamojo rikiavimo algoritmo* principas – kiekvienas elementas imamas iš eilės ir įterpiamas į jam skirtą vietą jau surikiuotoje objektų grupėje. Kiekvieną kartą įterpus objektą, neišrikiuotoje grupėje objektų mažėja, o išrikiuotoje – daugėja.

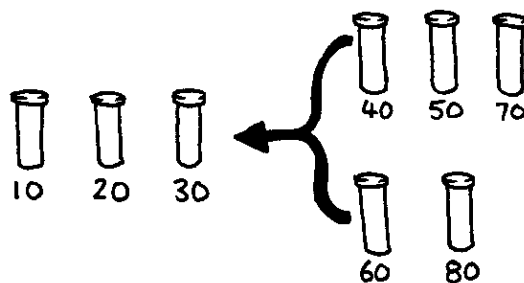


Burbulinio rikiavimo algoritmo principas – nuosekliai iš eilės patikrinti gretimų objektų poras, prireikus objektus sukeisti, t. y. perkelti mažesnį arčiau pradžios. Šitaip per pirmą žingsnį lengviausias objektas perkeliamas į pirmą poziciją, vėliau tas pats principas taikomas likusiems objektams ir t. t.

Šio algoritmo veikimo principas primena virimą, kai oro burbulai kyla į paviršių, dėl to jis ir vadinamas burbulinio rikiavimo metodu. Nors nėra labai efektyvus, šis metodas mėgstamas dėl savo paprastumo.



Sąlajinio rikiavimo algoritmas – vienas iš „skaldyk ir valdyk“ principu grindžiamų rikiavimo algoritmų. Pirmiausia objektai atsitiktinai padalijami į dvi lygias (arba panašias, jei yra nelyginis objektų skaičius) grupes. Kiekviena grupė rikiuojama atskirai, paskui abi grupės vėl sujungiamos: lyginant skirtingų grupių objektus po du iš eilės mažiausias objektas perkeliamas į jungtinės objektų grupės pabaigą. Paveiksle 40 ir 60 gramų objektai yra kiekvienos grupės pradžioje, į jungtinę grupę dedamas mažesnio svorio objektas – 40 g ir t. t.



Kaip išrikiuoti mažai objektų? Patartinas sąlajinio rikiavimo algoritmas! Visos grupės padalijamos po vieną objektą, todėl nereikia rūpintis, kada baigti rikiuoti.



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Apie ką visa tai?

Išrikiuotame sąrašė daug lengviau surasti reikiamą informaciją. Telefonų knygoje, knygų kartotekoje elementai rikiuojami pagal abėcėlę. Jei jie nebūtų išrikiuoti pagal abėcėlę, būtų daug sunkiau susigaudyti informacijos gausybėje ir rasti reikiamą pavardę telefonų knygoje ar autorių kartotekoje. Išrikiuotame skaičių sąrašė lengva pamatyti didžiausią, mažiausią ar vienodus elementus.

Dažnai rikiuojant pasitelkiami kompiuteriai, todėl informatikai ieško vis greitesnio rikiavimo algoritmo. Dauguma rikiavimo algoritmų, pavyzdžiui, *išrenkamojo*, *įterpiamojo ar burbulinio*, yra naudingi sprendžiant tam tikras problemas. Algoritmai, tokie kaip *greitojo* ar *sqlajinio rikiavimo*, yra daug efektyvesni ir pritaikomi didesniam objektų skaičiui. Pavyzdžiui, 100000 objektų *greitojo rikiavimo algoritmas* išrikiuoja 2000 kartų greičiau nei *išrenkamojo rikiavimo algoritmas*, o 1000000 objektų – 20000 kartų greičiau. Kompiuteris turi apdoroti milijonus duomenų (daugybė tinklalapių turi milijonus vartotojų, vienoje nuotraukoje yra milijonai pikselių). Šiuodu algoritmai skiriasi tuo, kad pirmasis apdoroja duomenis per 1 s, o antrajam apdoroti tuos pačius duomenis reikia 5 valandų. Ne tik prireikia ilgesnio laiko, bet ir energijos sunaudojama 20000 kartų daugiau. Laiko ir energijos sąnaudos neturi įtakos programinei įrangai, tačiau yra svarbios naudojant prietaisus su baterijomis (planšetinius ar nešiojamuosius kompiuterius, mobiliuosius telefonus ir kt.).

Greitojo rikiavimo algoritmas veikia „skaldyk ir valdyk“ principu. Objektai padalijami į mažesnes grupes, kiekvienai iš jų pritaikomas greitojo rikiavimo algoritmas. Algoritmas kartojamas, kol grupėse lieka po vieną objektą. Toliau paprasta rikiuoti, kai lieka tik po vieną objektą! Praktiškai šis metodas yra daug greitesnis nei kiti rikiavimo metodai. Jis realizuoja *rekursijos* idėją, pagal kurią algoritmas keletą kartų kreipiasi į patį save, kol išsprendžia problemą.



Sprendimai ir užuominos

1. Geriausias būdas surasti lengviausią objektą yra tikrinti visus objektus lyginant vieną su kitu. Kitaip tariant, objektai lyginami po du, iš kiekvienos poros ant svarstyklių lėkštelės paliekamas lengvesnis objektas tol, kol perrenkami ir palyginami visi objektai.
2. Objektų svoris lyginamas svirtinėmis svarstyklėmis. Tai galima atlikti palyginus svorį triskart, o kartais ir dukart, jei mokiniai supranta, kad lyginimo operacija yra tranzityvi (t. y. jei A yra lengvesnis už B, o B yra lengvesnis už C, tai A turi būti lengvesnis už C).

Gudručiams

Norint nustatyti mažiausio svorio objektą iš dviejų, reikia vieną kartą juodu palyginti. Ieškant lengviausio iš trijų objektų reikia tuos objektus palyginti du kartus, iš keturių objektų – reikia juos palyginti tris kartus ir t. t. Rikiuojant aštuonis objektus ir norint iš jų surasti pirmą lengviausią reikia 7 kartus palyginti visus objektus, šešis kartus – norint surasti antrą lengviausią objektą ir t. t. Išeina, kad reikia

$7 + 6 + 5 + 4 + 3 + 2 + 1 = 28$ kartus palyginti.

Rikiuojant n objektų reikėtų palyginti $1 + 2 + 3 + 4 + \dots + n - 1$ kartų.

Šiuos skaičius sudėti yra paprasta, kai jie sugrupuojami.

Pavyzdžiui, sudedant skaičius $1 + 2 + 3 + \dots + 20$, sugrupuojama

$(1 + 20) + (2 + 19) + (3 + 18) + (4 + 17) + (5 + 16) +$

$(6 + 15) + (7 + 14) + (8 + 13) + (9 + 12) + (10 + 11) = 21 \times 10 = 210$

Bendroju atveju gaunama $1 + 2 + 3 + 4 + \dots + n - 1 = n(n - 1)/2$.



8 veikla

Dar greičiau! Tinklinis rikiavimas

Santrauka

Kompiuterio sparta ne visada lemia problemos sprendimo spartą. Vienas iš būdų greičiau išspręsti problemą yra padalyti ją į dalis ir sprendimui naudoti kelis kompiuterius. Šiame skyriuje aiškinamas *tinklinio rikiavimo algoritmas*, kuriuo vienu metu galima lyginti kelis rikiavimo metodus.

Ryšiai su ugdymo programomis

- ✓ Matematika: skaičiai, skaičių palyginimas

Gebėjimai

- ✓ Lyginimo
- ✓ Rikiavimo
- ✓ Algoritmo sudarymo
- ✓ Problemos sprendimo bendradarbiaujant

Amžius

- ✓ Nuo 7 metų

Priemonės

Lauko žaidimas.

- ✓ Kreida
- ✓ Du šešių kortelių rinkiniai. Paruošiamos *tinklinio rikiavimo algoritmas* kortelės
- ✓ Laikmatis

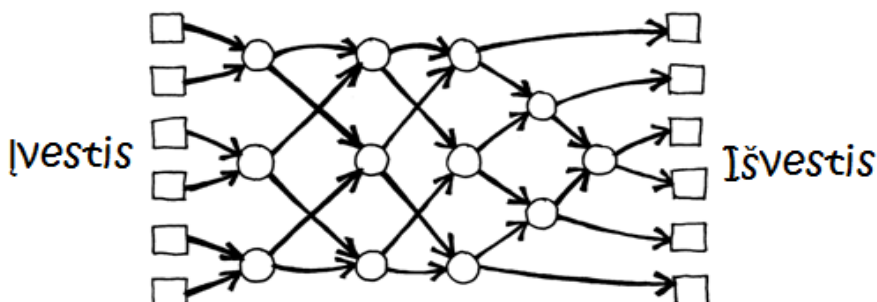


Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Tinklinis rikiavimas

Pirmiausia ant žemės (grindinio) nupiešiamas tinklas, kaip parodyta paveikslėlyje. Tinklas turi būti gana didelis, kad mokiniai galėtų po jį vaikščioti.



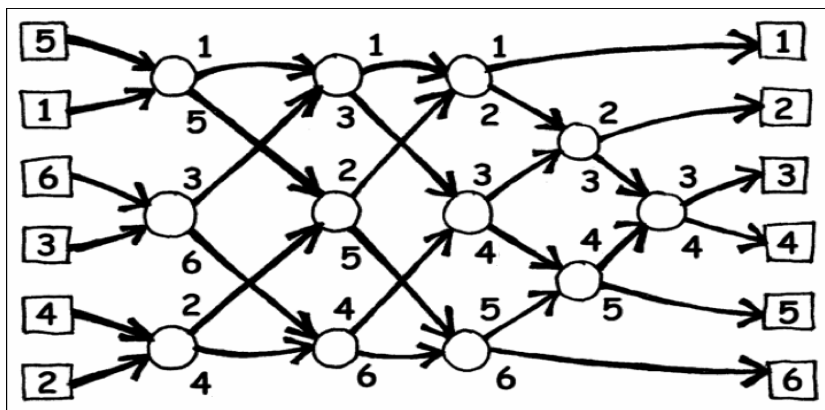
Instrukcijos

Šiame skyriuje supažindinama, kaip kompiuteris rikiuoja atsitiktinius skaičius *tinklinio rikiavimo algoritmu*.

1. Mokiniai susiskirsto į grupes po šešis. Vienu metu tik viena grupė naudojami tinklu.
2. Kiekvienas mokinys paima kortelę su ant jos užrašytu skaičiumi.
3. Kiekvienas mokinys atsistoja ant kvadrato kairėje tinklo pusėje (ten, kur pažymėta įvestis). Mokinių turimų kortelių skaičiai turėtų neiti iš eilės.
4. Mokinys juda nurodyta kryptimi. Atsistojęs ant apskritimo **palaukia kito mokinio**.
5. Du mokiniai susitikę apskritime palygina savo kortelių skaičius. Mokinys, kurio skaičius kortelėje yra mažesnis, eina toliau kairiosios rodyklės kryptimi, o kurio skaičius didesnis – dešinės rodyklės kryptimi.
6. Mokinys teisingai baigia žaidimą, kai pasiekia kvadratą tinklo dešinėje (ten, kur pažymėta išvestis).

Jei komanda suklysta, turi pradėti iš naujo. Prireikus mokiniams padedama suprasti, ką reikia daryti pasiekus apskritimą.

Rikiavimo pavyzdys parodytas žemiau paveiksle.



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Tinklinis rikiavimas

1

2

3

4

5

6

156

221

289

314

422

499



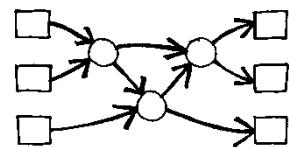
Variantai

1. Kiekvienos grupės tinklo perėjimo laikas gali būti išmatuojamas laikmačiu ir nustatoma greičiausiai šį rikiavimą atlikusi grupė.
2. Galima naudoti ir korteles su didesniais (pvz., triženkliais) skaičiais.
3. Galimos kortelės ir su labai dideliais skaičiais arba žodžiais, kuriuos galima išrikiuoti pagal abėcėlę.

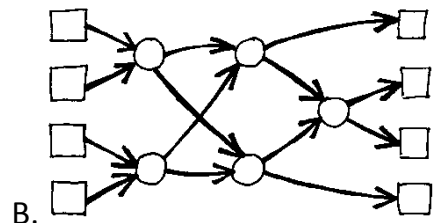
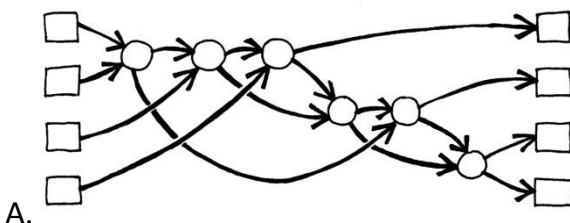
Gudručiams

4. Kas atsitiktų, jei visi mokiniai, turintys kortelių su mažesniais (didesniais) skaičiais, pasuktų ne į dešinę (kairę), o į kairę (dešinę)? (Skaičiai būtų išrikiuoti atvirkščiai.)
5. Ar visada skaičiai išrikiuojami teisingai pasiekus išvestį, jei rikiavimo kryptys pakeičiamos priešingomis? (Ne visada. Mokiniai turėtų pateikti pavyzdį, kada išrikiuojama neteisingai.)

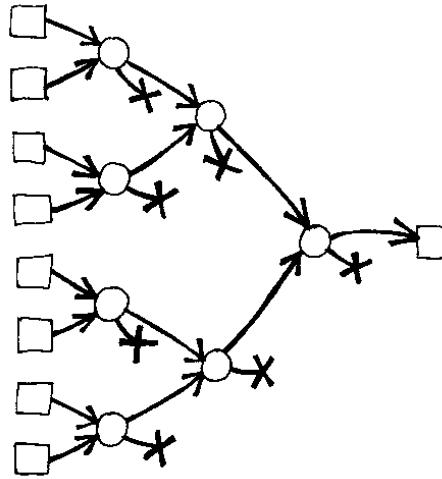
6. Mokiniam pasiūloma sukurti mažesnį ar didesnį tinklą. Pavyzdžiui, dešinėje, paveikslėlyje, vaizduojamas tinklas, kuriuo išrikiuojami tik trys skaičiai. Mokiniai turėtų patys sukurti tokį ar panašų tinklą.



7. Žemiau pateikiami du tinklai, kuriais išrikiuojama po keturis skaičius. Kuriuo iš jų – A ar B – tai padaroma greičiau? (B tinklu. A tinkle visi skaičiai lyginami iš eilės, o B tinkle keli skaičiai lyginami vienu metu. A tinkle naudojamas nuoseklusis palyginimas, o B – lygiagretusis.)



8. Mokiniam pasiūloma sukurti didesnį tinklą.
9. Naudojant tinklą galima surasti mažiausią arba didžiausią skaičių. Pavyzdžiui, žemiau paveikslėlyje išrenkamas mažiausias skaičius iš aštuonių. Atliekant šį algoritmą išvedamas tik mažiausias skaičius, o kiti lieka tinklo aklavietėse.



10. Kokie kasdieniniai darbai gali būti arba negali būti paspartinti taikant lygiagretumo principą? Pavyzdžiui, maisto ruošimas būtų daug lėtesnis, jei naudotume tik vieną kaitvietę, nes tada viskas turėtų būti verdama ar kepama paeiliui. Kurie darbai gali būti atlikti greičiau pasamdžius daugiau žmonių? Kuriuose darbuose neišeina panaudoti lygiagretumo principo?

Apie ką visa tai?

Kompiuteriai tampa neatsiejami nuo žmonių gyvenimo, todėl būtina, kad jie vis sparčiau apdorotų informaciją.

Yra keletas būdų paspartinti kompiuterio darbą: problemai spręsti kurti algoritmus, kurių atlikimo laikas būtų kuo trumpesnis, arba problemą dalyti dalimis ir jai spręsti naudoti keletą kompiuterių. Pavyzdžiui, jei norima išrikiuoti šešis skaičius tinkle, reikia 12 kartų palyginti skaičių poras. Tačiau vienu metu gali būti lyginamos net trys skaičių poros, o tai daugiau nei perpus sutrumpina rikiavimo laiką.

Žinoma, ne visas problemas galima išspręsti greičiau lygiagretumo principu. Pavyzdžiui, vienas žmogus kasa 10 m griovį. Jei dešimt žmonių kastų po 1 m, užduotis būtų atlikta daug greičiau. Tačiau to paties metodo negalima taikyti kasant 10 m gylio duobę, nes antrojo metro žemių iš duobės negalima kasti, kol neiškastas metras žemių virš jo. Informatikai aktyviai darbuojasi ieškodami geriausio problemos padalijimo metodo, kad būtų galima pritaikyti lygiagretumo metodą.



9 veikla

Purvinas miestas. Minimalūs jungiantieji medžiai

Santrauka

Realiaame gyvenime įvairių tinklų esama daug – tai telefonų tinklai, komunalinių paslaugų tinklai, kompiuterių tinklai, kelių tinklai ir kt. Kuriant paslaugų tinklus svarbu nustatyti, kur optimalu tiesti kelią, ryšio kabelius ar kt. Todėl dažnai sprendžiama efektyvaus objektų sujungimo į tinklą problema.

Ryšiai su ugdymo programomis

- ✓ Matematika: geometrija, geometrinės figūros, trumpiausias kelias žemėlapyje

Amžius

- ✓ Nuo 9 metų

Gebėjimai

- ✓ Problemų sprendimo

Priemonės

Kiekvienam mokiniui reikės:

- ✓ Darbo lapo „Purvinas miestas“



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Purvinas miestas

Įvadas

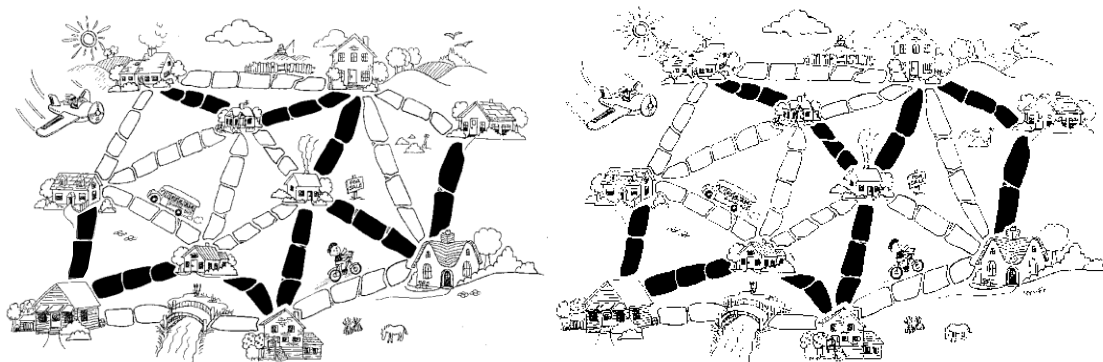
Ši veikla skiriama susipažinti, kaip kompiuteriai padeda surasti geriausią realios problemos – kaip geriausiai nutiesti kelią – sprendimą. Mokiniais išdalijami darbo lapai, kuriuose paaiškinta *Purvino miesto* problema.

Diskusija

Aptariama, kokiais būdais mokiniai galvoja ieškoti sprendimo.

Šią užduotį galima spręsti keliais būdais. Vienas būdas – pradėti nuo miesto plano, kuriame nėra sužymėta gatvių. Reikia žymėti po vieną gatvę, jungiančią du namus, prie kurių gatvė dar nenutieta. Kitas uždavinio sprendimo būdas, kai miesto plane yra sužymėtos visos gatvės, – šalinti po vieną nereikalingą gatvę ir tikrinti, ar į visus namus galima patekti likusiomis gatvėmis.

Šios užduoties sprendiniai gali būti skirtingi – visi namai gali būti sujungti vienodo ilgio, bet skirtingomis gatvėmis. Žemiau paveiksle parodyti du skirtingi šios užduoties sprendiniai.



Toliau su mokiniais galima aptarti, kur prireikia tinklų realiame gyvenime.

Informatikoje šie tinklų vaizdavimo būdai vadinami *grafais*. Grafas – tai taškai (viršūnės), sujungti linijomis (briaunomis). Realiame gyvenime tinklai gali būti vaizduojami grafais ieškant optimalaus sprendimo.

Grafų uždaviniams spręsti yra sukurta ir daug algoritmų, kuriais, pavyzdžiui, lengviau rasti trumpiausią atstumą tarp dviejų taškų arba trumpiausią kelią, jungiantį visus taškus.



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

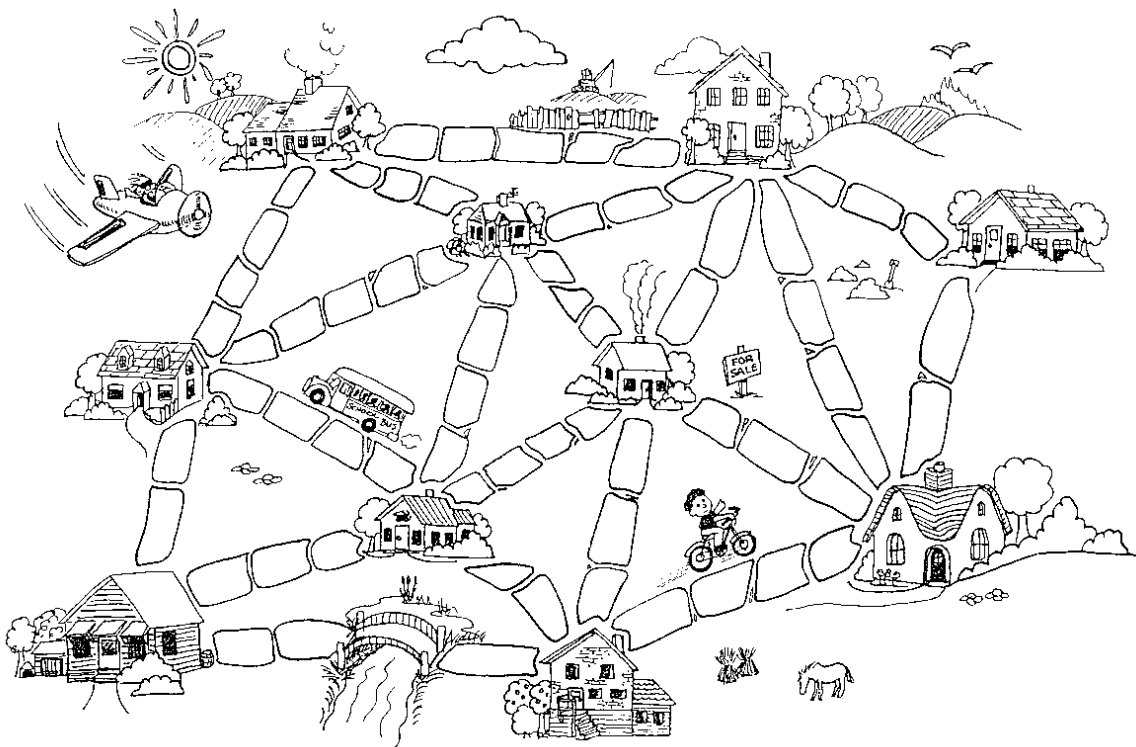
Darbo lapas *Purvinas miestas*

Vienoje šalyje yra miestas, kuriame nenuitiesta grįstų gatvių. Po lietaus žemė mieste pažliunga, automobiliai klimpsta purve, o žmonės išsipurvina. Miesto meras nusprendė išgrįsti kai kurias miesto gatves, tačiau jis nenori išleisti daugiau pinigų, nei būtina, nes mieste dar reikia pastatyti baseiną. Meras iškėlė du reikalavimus:

1. Grįstų gatvių turi būti tiek, kad jomis būtų galima nueiti į kiekvieną namą.
2. Gatvių grindimo išlaidos turi būti mažiausios.

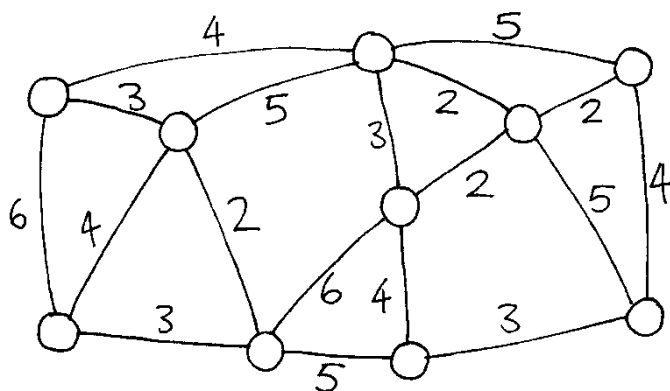
Pateiktas miesto planas. Akmenų skaičius tarp kiekvieno namo proporcingas gatvės grindimo kainai. Suraskite geriausią gatvių grindimo variantą, kuris sujungtų visus namus ir būtų pigiausias.

Pagalvokite, koku būdu galima pradėti spręsti šią problemą?



Variantai gudručiams

Paveiksle *grafu* pavaizduoti namai ir keliai. Šiame grafe namai vaizduojami apskritimais, o keliai – linijomis, jungiančiomis apskritimus. Kelių ilgi ai nurodyti skaičiais ties linijomis.



Informatikoje ir matematikoje dažnai uždaviniams spręsti naudojami *grafai*. Sutinkant pirmą kartą grafo sąvoką, galima pagalvoti, kad tai kažkas panašaus į grafiką, vartojamą statistikoje duomenims vaizduoti. Nieko panašaus, grafai informatikoje yra kaž kita.

Grafuose linijų ilgis neturi reikšmės – jokio mastelio nereikia laikytis.

Mokiniai galėtų sukurti savo miestą ir draugų paprašyti sujungti vaizduojamus namus gatvėmis. Kiek mažiausiai reikia gatvių miesto namams sujungti? Ar mažiausias miesto gatvių skaičius priklauso nuo namų skaičiaus?

Apie ką visa tai?

Tinklo sujungimo problemos dažnos realiame gyvenime. Kuriant naują (ar plečiant jau esančią) gyvenvietę reikia suprojektuoti elektros, dujų ir vandens linijas į kiekvieną namą. Jomis tiesiami elektros laidai, dujų ar vandens vamzdžiai. Kiekvienas namas turi būti prijungtas prie reikiamo tinklo.

Problema, kai reikia suprojektuoti trumpiausią tinklą, vadinama *minimalaus jungiančiojo medžio* užduotimi.

Minimalus jungiantysis medis pravartus ne tik komunalinių tinklų problemai spręsti, bet ir kompiuterių tinklams, telefono ryšio, naftotiekio ar oro linijoms projektuoti. Planuojant geriausią kelionės maršrutą reikia atsižvelgti ne tik į kelionės patogumą, bet ir į kainą. Niekas nenori praleisti valandų valandas lėktuve keliaudamas į kitą šalį aplinkiniu maršrutu vien dėl to, kad tokia kelionė yra pigesnė. Planuojant kelionę „purvino miesto“ algoritmas nėra labai tinkamas, nes juo sutrumpinamas tik bendras gatvės ilgis.

Minimalus jungiantysis medis naudojamas sprendžiant ir kitus *grafų* uždavinius, pavyzdžiui, sprendžiant „Keliaujančio pirklio“ uždavinį ieškoma trumpiausio kelio, kuriuo turi būti aplankytos visos *grafo* viršūnės.

Minimalaus jungiančiojo medžio uždaviniams spręsti sukurta daug efektyvių algoritmai. Galima pradėti nuo miesto plano, kuriame nesusūžymėtos gatvės, ir žymėti po vieną gatvę, jungiančią du, iki šiol dar neprijungtus namus. Šį algoritmą 1956 m. aprašė J. B. Kruskalas, todėl jis vadinamas J. B. Kruskalo algoritmu.

Informatikoje *grafų* uždaviniams, pavyzdžiui, „keliaujančio pirklio“, kuo geriau išspręsti sukurta ir kuriami gana efektyvūs metodai.



Sprendimai ir užuominos

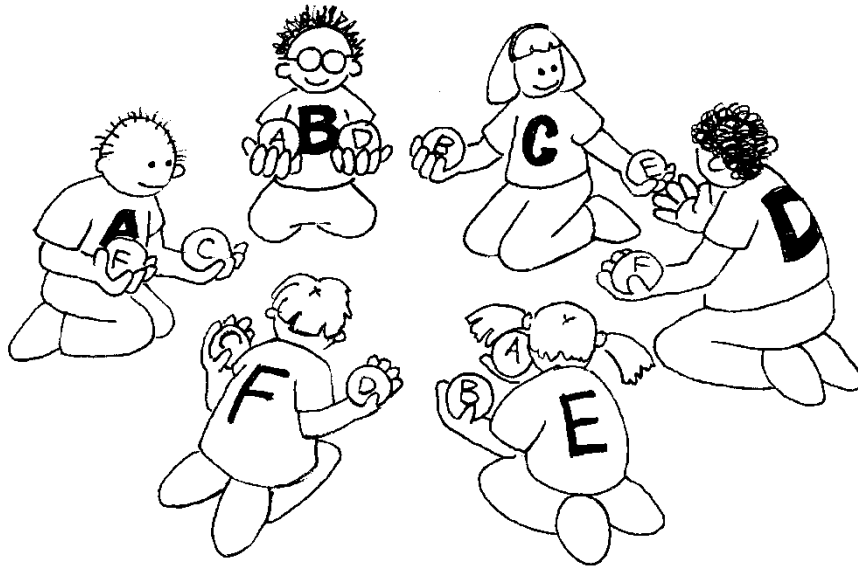
Variantai gudručiams

Kiek gatvių reikia norint sujungti n namų? Optimalus sprendimas – reikia $n-1$ gatvių.



10 veikla

Apelsinų žaidimas. Maršruto parinkimas ir tinklo aklavietė



Santrauka

Kai vienu metu daugeliui žmonių prireikia to paties (pavyzdžiui, daug vairuotojų pasirenka tą patį maršrutą ar vienu metu gaunama daug žinučių), gali susidaryti aklavietė. Veikiant sutartinai galima išvengti šio nemalonumo.

Ryšiai su ugdymo programomis

- ✓ Matematika: loginės problemos

Gebėjimai

- ✓ Problemos sprendimo bendradarbiaujant
- ✓ Loginio pagrindimo

Amžius

- ✓ Nuo 9 metų

Priemonės

Kiekvienam mokiniui reikės:

- ✓ Dviejų apelsinų (ar kitų vaisių) arba teniso kamuoliukų
- ✓ Priemonės raidei užrašyti ar priklijuoti ant kiekvieno mokinio ir jo apelsinų arba teniso kamuoliukų



Apelsinų žaidimas

Įvadas

Žaisdami šį žaidimą mokiniai sprendžia bendradarbiavimo problemą. Kiekvieno žaidėjo tikslas yra perduoti savo apelsinus kitiems grupės nariams.

1. Penki ar daugiau mokinių susėda ratu.
2. Kiekvienam mokiniui išdalijama po du apelsinus, išskyrus vieną mokinį. Jam duodamas vienas apelsinas, kad būtų viena laisva vieta rate.
3. Mokiniai pavadinami abėcėlės raidėmis iš eilės (galima priklijuoti lipdukus su raidėmis ant jų marškinėlių). Kiekvieno mokinio apelsinai pažymimi jo raidėmis.
4. Apelsinai siunčiami ratu, kol kiekvienam mokiniui lieka po vieną apelsiną su jo raide. Apelsinai siunčiami pagal dvi taisykles:
 - a) Rankoje gali būti tik vienas apelsinas vienu metu.
 - b) Apelsinas gali būti perduotas tik šalia esančiam mokiniui, jei jis rankoje neturi apelsino. (Mokinys gali perduoti bet kurį iš savo apelsinų.)

Mokiniai turėtų pastebėti: jei jie skuba ir paima kaimyno apelsiną, kol savo dar neatidavė, grupė negali pasiekti žaidimo tikslo. Galbūt reikėtų pabrėžti, kad mokiniai tada nelaimi žaidimo, o žaidimas baigiamas, kai kiekvienas laiko ne savo apelsinus rankose.

Diskusija

Kokiais būdais mokiniai sprendžia problemą?

Kokiose realaus gyvenimo situacijose pasitaiko aklaviečių? (Pavyzdžiui, eismo spūstys, žaidžiant krepšinį, žmonių spūstis tarpduryje bandant išeiti iš patalpos.)

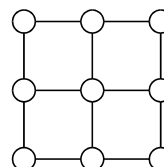
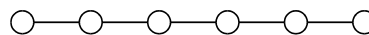
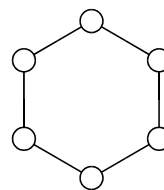
Gudručiams

Mokiniai gali išbandyti žaidimą su daugiau (ar mažiau) mokinių.

Tegul mokiniai pasvarsto apie naujas žaidimo taisykles.

Tegul mokiniai atlieka veiklą nekalbėdami.

Tegul mokiniai išbando pakeistus žaidimus: susėsti viena linija arba turėti daugiau nei du kaimynus, parodyta paveikslėliuose.



kaip



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Apie ką visa tai?

Maršruto parinkimas ir aklavietė yra dažna problema reguliuojant eismą, telefono ryšio linijas ir informacijos perdavimo kompiuteriu srautus. Inžinieriai praleidžia daug laiko ieškodami šios problemos sprendimo, bandydami suprojektuoti tinklus, kad šios problemos sprendimas būtų paprastesnis.

Maršruto parinkimas, spūstis, grūstis ir aklavietė – tai įvairių tinklų sudarymo nesėkmių išraiška. Didžiuosiuose miestuose dažnai pasitaiko eismo spūsčių. Kartais jos tampa aklavietėmis automobiliams, kai jie negali pajudėti nei pirmyn, nei atgal. Inžinieriai sprendžia sudėtingas problemas ieškodami paprastesnio ir pralaidesnio maršruto, kuriame spūsčių tikimybė būtų kuo mažesnė.

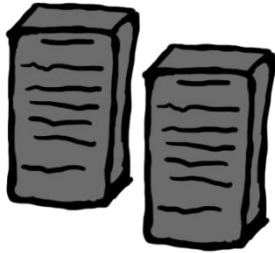
Kartais daugiau nei vienas vartotojas naudoja tuos pačius tinklo duomenis tuo pačiu metu. Jei vienas vartotojas nori atnaujinti duomenis tinkle, patartina juos „užblokuoti“. Jei vienu metu du vartotojai bando atnaujinti tuos pačius duomenis, tai duomenys gali būti įrašyti neteisingai. Tačiau, jei bandoma iš skirtingų vietų užblokuoti tuos pačius duomenis, gali susidaryti aklavietė.

Vienas iš didžiausių informatikos pasiekimų yra lygiagretusis skaičiavimas, kai keletas kompiuterių dirba vienu metu sudarydami vieną didelio galingumo kompiuterį. Tokius žaidimus, kaip „Apelsinų“ žaidimas, su šiuo kompiuteriu galima žaisti be perstojo ir daug greičiau.



11 veikla

Akmeninės plokštelės. Tinklo komunikacijų protokolas



Santrauka

Kompiuteriai perduoda informaciją vieni kitiems internetu, tačiau internetas nėra patikimas ir kartais siunčiama informacija prarandama. Pridėjus prie siunčiamos žinutės tam tikros papildomos informacijos, šios žinutės perdavimo patikimumas padidėja daug kartų. Ši papildoma informacija sudaro komunikacijų protokolą.

Ryšiai su ugdymo programomis

- ✓ Matematika: loginės problemos
- ✓ Lietuvių kalba: komunikavimas, kito asmens klausymasis

Gebėjimai

- ✓ Problemos sprendimo bendradarbiaujant
- ✓ Loginio pagrindimo

Amžius

- ✓ Nuo 9 metų

Priemonės

Kiekvienam mokiniui reikės:

- ✓ Daug tuščių plokštelių (apie 30)

Kiekvienam pasiuntiniui reikės:

- ✓ Žinučių kortelių rinkinio

Mokytojui reikės:

- ✓ Laikmačio



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

Akmeninės plokštelės

Išvadas

Ši veikla skiriama supažindinti, kaip veikia skirtingi komunikavimo metodai. Pateikiamos taisyklės ir aprašomi veiksmai mokinius supažindina su komunikavimo protokolais. Mokiniai dalyvauja vaidmenų žaidime, testuoja savo protokolo veikimą nepatikimoje aplinkoje, panašioje į paketų perdavimą internetu (TCP/IP).

Pasirengimas (30 min.)

1. Pasiruošiamos kortelės: jos atspausdinamos ir sukarpomos.
2. Suformuluojama žinutė. Nusprendžiama, kokia žinutė bus siunčiama. Žinutė turi būti beprasmė, kad nebūtų galima nuspėti jos struktūros perduodant. Pavyzdžiui, „1LHC255HD(RLLS“ arba telefono numeris.
3. Pasiruošiamos plokštelės: jos atspausdinamos ir sukarpomos. Kiekvienoje plokštelės yra vietos 6 ženklams (vienas žinutės ženklas rašomas į vieną langelį), todėl viena žinutė netelpa į vieną plokštelę. Žinutei perduoti reikės apie 30 plokštelių kiekvienam mokiniui.

Patarimai. Kortelės žymi tris veiksmus: atidėti, pristatyti ir nepristatyti. Šių veiksmų derinimas nusako perdavimo tinklo kokybę. Kuo daugiau kortelių „pristatyti“, tuo tinklas patikimesnis. Kuo daugiau kortelių „atidėti“, tuo tinklas mažiau patikimas. Šios kortelės vaizduoja kompiuterio tinklo ir komunikacijų kanalą.

Žaidimo instrukcija

1. Mokiniai pasidalija į poras. Vienos poros nariai turi atsisėsti kuo nuošaliau, kad nematytų vienas kito ir negalėtų kalbėtis. Jie gali būti skirtinguose kambariuose ar nusisukę vienas nuo kito skirtingose vieno kambario pusėse.
2. Vienam poros nariui duodama žinutė, kurią jis turi perduoti savo porininkui per pasiuntinį.
3. Veiksmų kortelės sumaišomos ir pasirenkamas pasiuntinys. Juo gali būti mokytojas ar mokinys, jei mokinių skaičius nelyginis ir vienas mokinys lieka be poros. Jei klasė didelė, galima pasirinkti du pasiuntinius.
4. Mokiniui, turinčiam žinutę, duodamos plokštelės. Pirmoje plokštelėje jis parašo dalį žinutės. Virš plokštelės užrašo mokinio, kuriam siunčia žinutę, vardą ir perduoda plokštelę pasiuntiniui.
5. Pasiuntinys, paėmęs plokštelę, traukia veiksmų kortelę, ją apverčia ir perskaito, ką turi padaryti su šia plokštele.
6. Kartojami 4 ir 5 žingsniai su kiekviena plokštele.



Po 5 ar daugiau chaoso ir nesėkmių minučių, mokiniai turėtų suprasti, kad protokolui tik vardo neužtenka. Sustabdomas žaidimas ir pradedama diskusija. Su kokia problema mokiniai susidūrė pirmiausia? Ar tai plokštelių eiliškumas? Tikriausiai būtų protinga vieną vietą iš tų 6 skirtų ženklams skirti siunčiamos žinutės plokštelės eilės numeriui? Tai reiškia, kad žinutės tekstui lieka mažiau vietos. Kaip tai lemia naudojamų plokštelių skaičių?

Po kurio laiko mokiniai turėtų pastebėti kitas problemas. Žaidimas sustabdomas ir padiskutuojama. Galima pastebėta problema – dingusios plokštelės, nežinoma, ar plokštelė buvo neišsiųsta, ar reikia siųsti iš naujo. Tikėtina, kad mokiniai pasiūlys sprendimą: gavęs plokštelę porininkas turi išsiųsti patvirtinimą ir tik gavus patvirtinimą išsiunčiama kita žinutės dalis. Tai reiškia, kad porininkas taip pat turi turėti tuščią plokštelę, o prieš tęsdami žaidimą mokiniai turi susitarti, kam bus naudojamos 6 ženklų vietos jų plokštelėse.

Šiam žaidimui užtenka dviejų mokinių, bet rekomenduojama žaisti su visa klase. Jei klasė didelė, gali būti ir keli pasiuntiniai. Aptariama su klase, kas atsitiktų, jei būtų daug pasiuntinių arba jų būtų tik vienas?



Pristatyti šią žinutę dabar	Pristatyti šią žinutę po kitos
Pristatyti šią žinutę dabar	Pristatyti šią žinutę po kitos
Pristatyti šią žinutę dabar	Pristatyti šią žinutę po kitos
Pristatyti šią žinutę dabar	Nepristatyti šios žinutės
Pristatyti šią žinutę dabar	Nepristatyti šios žinutės



Kam: <table border="1"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> Nuo ko:							Kam: <table border="1"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> Nuo ko:						
Kam: <table border="1"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> Nuo ko:							Kam: <table border="1"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> Nuo ko:						
Kam: <table border="1"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> Nuo ko:							Kam: <table border="1"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> Nuo ko:						
Kam: <table border="1"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> Nuo ko:							Kam: <table border="1"> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> <tr><td></td><td></td></tr> </table> Nuo ko:						



Akmeninės plokštelės

Antikos laikais vieną miestą valdė keletas valdytojų, kurie gyveno skirtingose to miesto vietose. Valdydami vieną miestą, jie kartu turėdavo aptarti labai svarbius klausimus.

Norėdami bendrauti nuotoliniu būdu, valdytojai turėdavo siųsti vienas kitam žinutes per visą miestą. Pasiuntiniai valdytojus skirdavo pagal jų namų numerius.

Vienintelis būdas nusiųsti žinutę – užrašyti ją ant akmeninės stačiakampės plokštelės ir nunešti ją tam, kam liepdavo siunčiantis miesto valdytojas. Akmeninės plokštelės dydis buvo nustatytas toks, kad joje tilpdavo tik 6 informacijos ženklai. Vienas informacijos ženklas galėjo būti viena raidė arba vienas skaičius. Visa žinutė galėdavo būti užrašoma ir ant kelių plokštelių, bet jos buvo labai sunkios ir vienu metu pasiuntinys galėdavo nunešti tik vieną plokštelę.

Be to, pasiuntiniais pasitikėti nebuvo galima, nes jie buvo užmaršūs ir tingūs. Jie užmiršdavo, kam skirta žinutė arba sustodavo pailsėti ilgesnį laiką ir nenunešdavo plokštelės laiku.

Miesto valdytojai norėjo surasti patikimesnį bendravimo būdą. Jie sumanė sukurti taisykles, kurios padėtų sužinoti, ar išsiųsta žinutė pasiekė adresatą, ar gauta žinutė buvo teisinga. Tačiau iš pradžių valdytojai sutarė tik dėl to, kad plokštelėje turi būti rašoma, kam žinutė skirta.

Ši veikla turėtų padėti suformuluoti taisykles, kaip miesto valdytojai turėtų rašyti žinutes.



Apie ką visa tai?

Siunčiami internetu duomenys dalijami į paketus. Tačiau tų duomenų siuntimo kanalai ne visada patikimi. Vienas ar kitas paketas kartais yra sugadinamas, prarandamas arba sumaišoma jo eilė.

Akmeninių plokštelių žaidime plokštelės vaizduoja paketus, o jų turinys – duomenis. Paketuose yra ir duomenys, ir *antraštės* informacija. *Antraštės* informacijos kiekis lemia, kiek informacijos galima perduoti vienu paketu. Kadangi paketai yra nustatyto dydžio, reikia numatyti žinutės ir papildomos informacijos kiekio balansą.

Mokiniai pastebi, kad keičiantis plokštelėms dalį žinutės vietos reikia skirti kitokiai informacijai: paketo numeriui, paketą sudarančių pakų skaičiui arba patvirtinimui, ar paketas gautas. Kuo daugiau papildomos informacijos įtraukiama į paketus, tuo daugiau reikia tokių paketų.

Interneto protokolai, pavyzdžiui, TCP ir UDP, suderina šiuos veiksnius ir užtikrina patikimą ir efektyvų duomenų perdavimą.

Ši veikla parengta ir aprašyta pagal projekto „Computing Science Inside“ skelbiamą medžiagą (csi.dcs.gla.ac.uk).



III dalis

**Kaip pasakyti kompiuteriui, ką
jis turi daryti?**

Algoritmų užrašymas



Kaip pasakyti kompiuteriui, ką jis turi daryti?

Kompiuteris įvykdo milijonus paliepimų per vieną sekundę. Kad kompiuteris vykdytų tuos paliepimus, reikia teisingai surašyti jam nurodymus. Tai nėra taip lengva, kaip gali atrodyti!

Gavęs kokį nors nurodymą žmogus naudojasi savo patirtimi ir suvokia, ką turėtų daryti. Jei žmogui sakoma „eik pro šias duris“, jis supranta, kad reikia eiti pro duris, o jei jos uždarytos – pirma reikia jas atidaryti. Kompiuteris nurodymus supranta kitaip. Iš tikrųjų, jei kompiuteris yra įmontuotas į judantį robotą, norint išvengti nesėkmių reikia imtis atsargumo ir saugumo priemonių, nes jis komandas supranta pažodžiui. Jei roboto kompiuteriui liepiama eiti pro duris, tai robotas ir eina kiaurai pro duris, nekreipdamas dėmesio, ar jos uždarytos, ar ne.

Šioje dalyje aiškinama, kaip bendraujama su viską pažodžiui suprantančia mašina naudojant baigtinę komandų seką.

Pirmos veiklos skyriuje supažindinama su „mašina“, atpažįstančia žodžius, skaičius ir simbolių eilutes, su kuriomis kompiuteris dirba. Ši „mašina“ vadinama baigtiniu automatu.

Atliekant antrąją veiklą sužinoma, kaip susikalbėti su kompiuteriu. Programuotojas turi išmokti nurodyti kompiuteriui, ką šis turi daryti, aiškiomis komandomis, kurias kompiuteris supranta pažodžiui. Komandų sąrašas vadinamas programa. Yra daug įvairių programavimo kalbų, kurias programuotojas gali vartoti programoms kompiuteriui rašyti. Mes pasirinkome paprastą kalbą, kuri gali būti vartojama ir be kompiuterio.



12 veikla

Lobio ieškojimas. Baigtiniai automatai

Santrauka

Kompiuterių programos apdoroja įvairių dokumentų arba kitų programų tekstų sekas, sudarytas, pavyzdžiui, iš raidžių arba žodžių. Informatikos mokslininkai šiuo tikslu naudoja baigtinį automatą, kuris atlieka komandas ir patikrina, ar kompiuteris atpažįsta žodžių ar simbolių eilutes. Pasitelkdami lobių žemėlapius, susipažinsime su baigtinio automato darbu.

Ryšiai su ugdymo programomis

- ✓ Matematika: loginės užduotys
- ✓ Socialiniai mokslai: Baltijos jūros salos
- ✓ Lietuvių kalba: sakinio sudarymas

Gebėjimai

- ✓ Žemėlapių skaitymo
- ✓ Schemos aiškinimo
- ✓ Logikos
- ✓ Komandų (instrukcijų) vykdymo

Amžius

- ✓ Nuo 9 metų

Medžiaga

Reikės:

- ✓ Salų kortelių rinkinio. (Instrukcijų neturi matyti mokiniai, braižantys maršrutus žemėlapyje!) Salų kortelės (p. 108). (Salų kortelė perlenkiama per lenkimo liniją taip, kad vienoje pusėje būtų sala su pavadinimu, o kitoje – komandos.)

Kiekvienam mokiniui reikės:

- ✓ Darbo lapo „Maršrutai iki lobio Hyjumos saloje“ (p. 107)
- ✓ Rašymo priemonės

Gudručiams siūloma atlikti papildomų veiklų. Jiems reikės:

- ✓ Darbo lapo „Lobio slėpimas“ (p. 113)
- ✓ Darbo lapo „Paslaptingas monetos žaidimas“ (p. 114)



Lobių sala

Įvadas

Tikslas – surasti lobių salą Baltijos jūroje. Draugiški Baltijos jūros piratai laivais plaukioja pastoviais maršrutais tarp salų. Norėdami pralobti piratai siūlo keliones turistams. Iš kiekvienos salos išplaukia du jū laivai A ir B. Turistai gali pasirinkti, kuriuo laivu plaukti. Į kiekvieną salą galima atvykti arba laivu A, arba laivu B, bet ne abiem. Saloje yra žmogus, kuris pasako, į kurią salą laivas plauks toliau. Tačiau piratai yra mažaraščiai ir neturi visų salų žemėlapių, todėl turistams duotame žemėlapyje reikia pažymėti maršrutą ir laivus, kuriais galima keliauti toliau. Reikia surasti geriausią maršrutą į salą, kurioje paslėptas lobis.

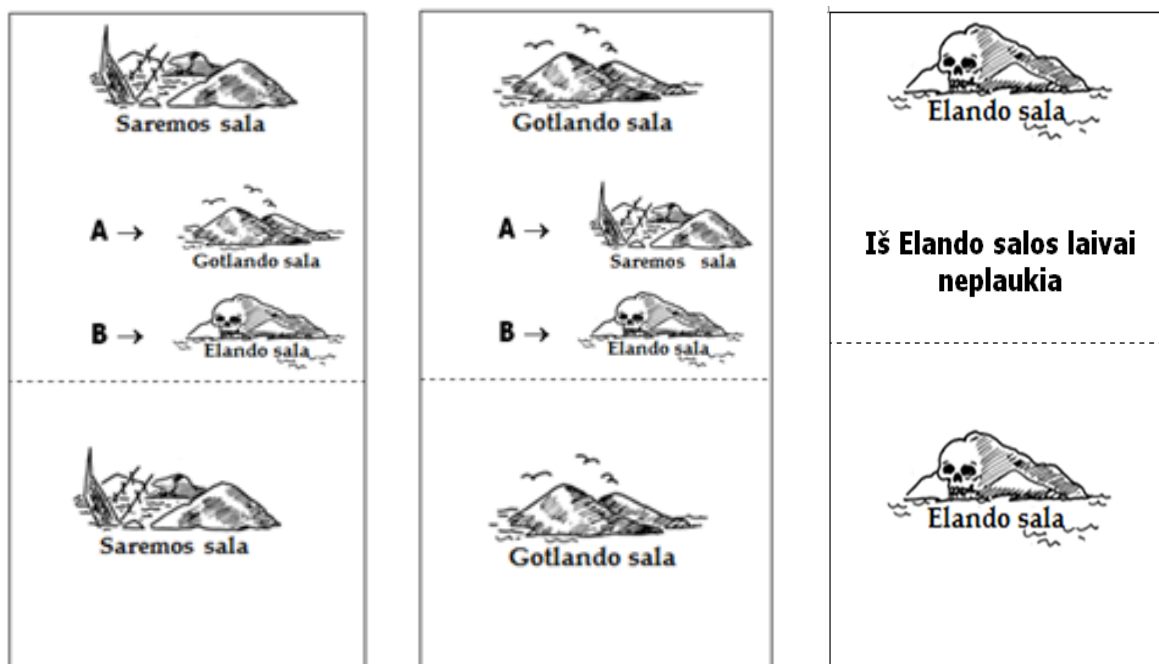
Demonstracija

(**Patarimai.** Kiekvienai veiklai yra skirtingi salų žemėlapiai.)

Lentoje nupiešiamas trijų salų žemėlapis, kaip parodyta žemiau paveiksle.

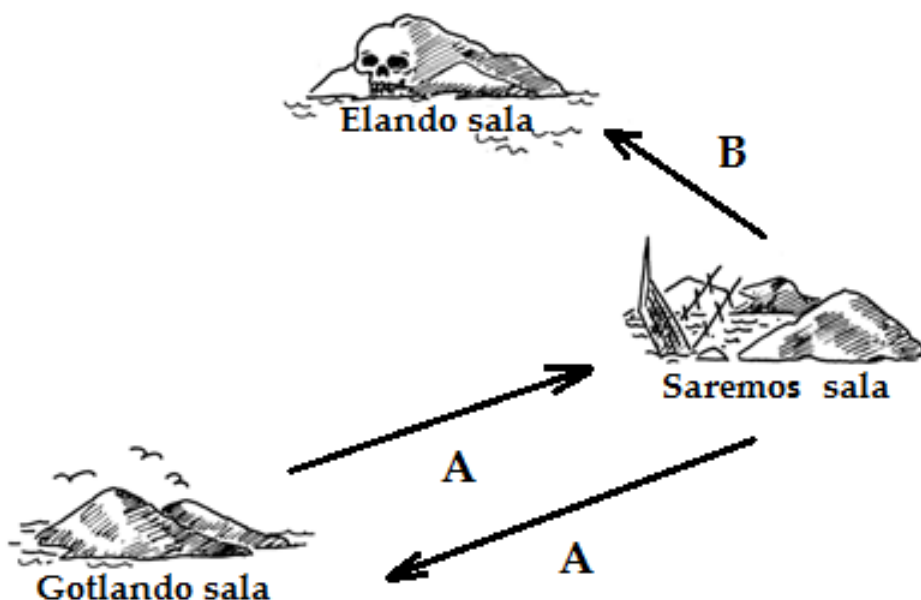


Atspausdinamos trys demonstravimo kortelės (p. 104–106). Pakviečiami trys mokiniai, kurių kiekvienam duodama kortelė su sala ir instrukcija. Įspėjama, kad veikloje, kurią atliks mokiniai vėliau, bus naudojamos kortelės su kitais maršrutais.



Kelionė pradedama Saremos saloje. Pasirenkamas laivas A. Mokiniai gali nukreipti laivą į Gotlando salą. Pažymimas maršrutas lentoje. Gotlando saloje vėl pasirenkamas laivas A. Grįžtama į Saremos salą. Lentoje pažymima kelionė. Dabar pasirenkamas laivas B. Šis laivas plaukia į Elando salą, kur patenkama į aklavietę.

Galutinis maršrutas atrodo taip:



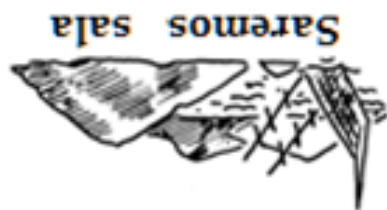
Demonstravimo kortelės



A →



B →



A →



B →



Demonstravimo kortelės



Iš Elando salos laivai
neplaukia

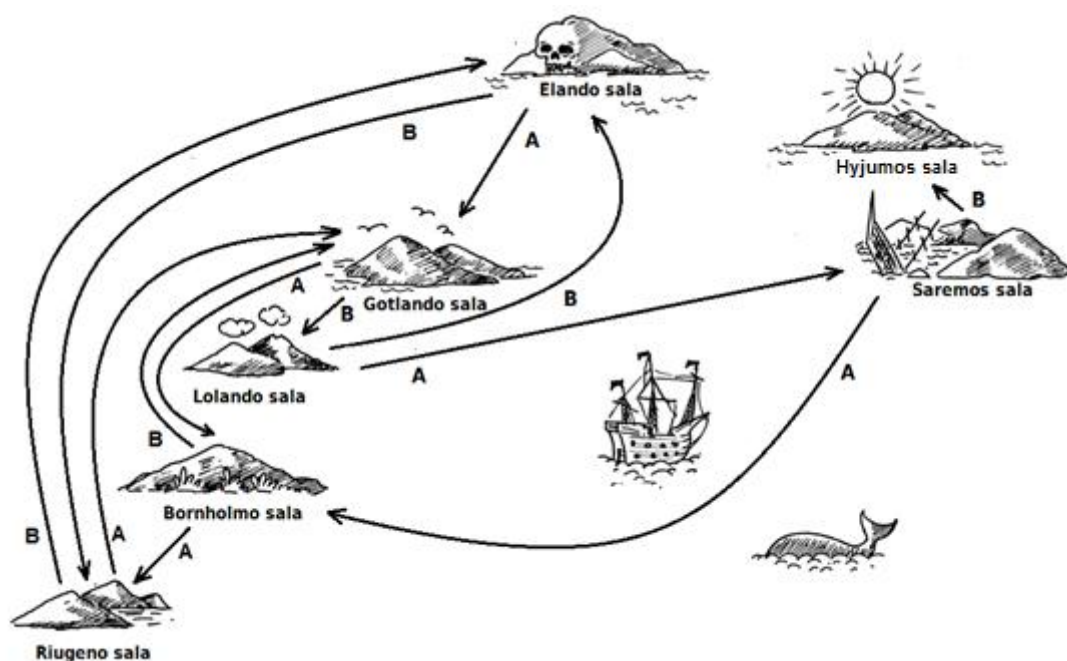


Veikla

Parenkami 7 mokiniai, turėsiantys vaizduoti salas. Jie laiko savo salos kortelę (p. 108–111), kurios kitoje pusėje užrašytos slaptos instrukcijos. Salas vaizduojantys mokiniai atsistoja kas sau aikštelėje ar kambaryje. Likusieji mokiniai vaizduos turistus. Kiekvienam iš jų duodamas salų žemėlapis (p. 107), kuriame reikia nubraižyti maršrutą iš **Gotlando salos** į **Hyjumos salą**. Pastarojoje saloje paslėptas lobis. Kol salas vaizduojantys mokiniai renkasi korteles ir sustoja į vietas, „turistus“ geriausia išsiųsti už durų.

Jei kuris iš „turistų“ greitai nubraižo maršrutą, pasiūlykite rasti daugiau maršrutų iki Iobio salos.

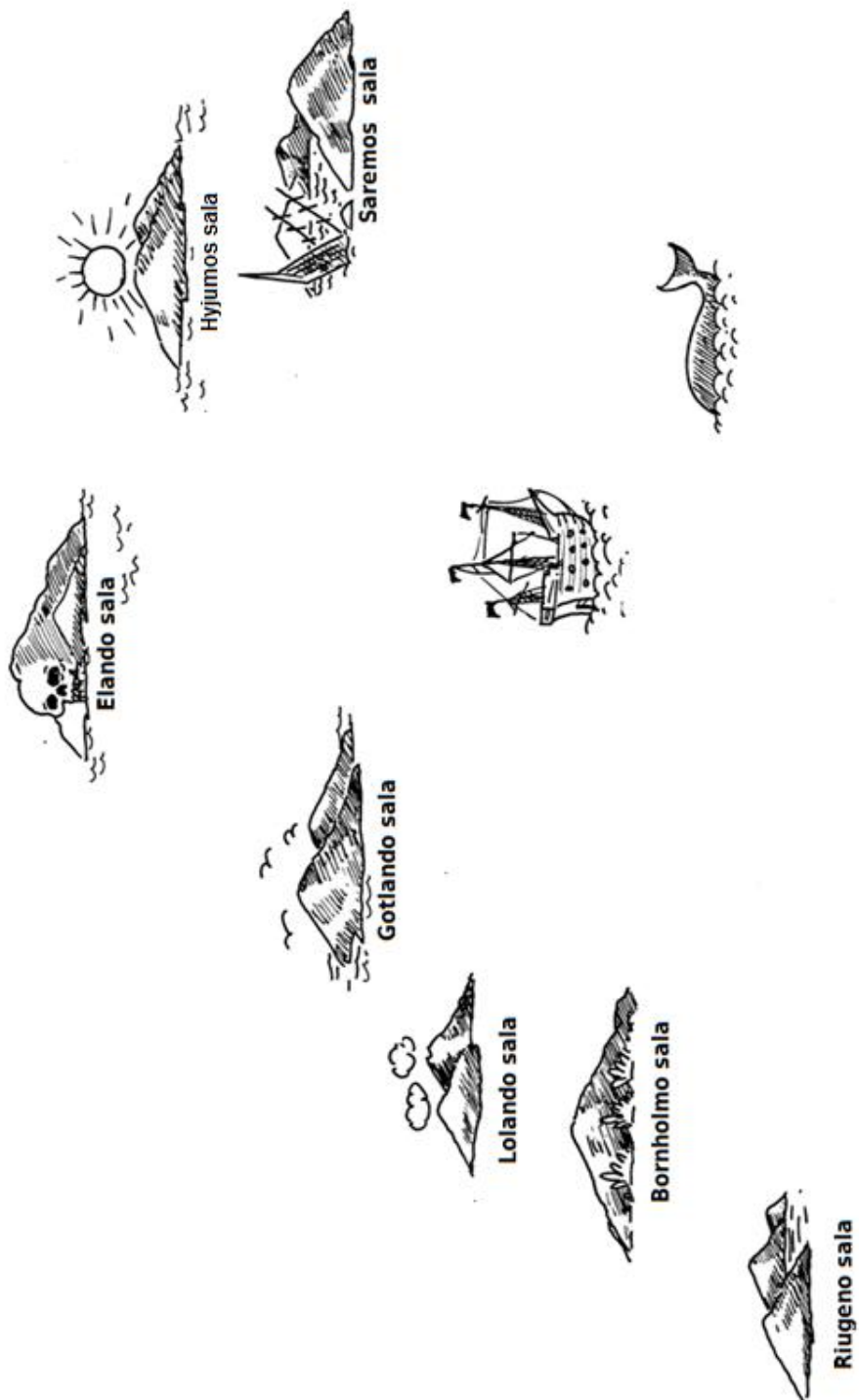
Visi maršrutai parodyti paveiksle žemiau:



Diskusija

Koks yra trumpiausias maršrutas? Koks galėtų būti ilgiausias maršrutas? Kai kuriuose maršrutuose gali būti ciklų. Tegul mokiniai pateikia maršrutų su ciklais pavyzdžių. (Pavyzdžiui, ir maršrutu BBBABAB, ir maršrutu BBBABBABAB pasiekama Hyjumos sala.)

Darbo lapas *Maršrutai iki lobio Hyjumos saloje*



Salų kortelės su instrukcijomis (1/4)



Bornholmo sala

A → 

Riugeno sala

B → 

Gotlando sala

Bornholmo sala





Riugeno sala

A → 

Gotlando sala

B → 

Elando sala

Riugeno sala



Salų kortelės su instrukcijomis (2/4)



Gotlando sala

A → 
Bornholmo sala

B → 
Lolando sala



Gotlando sala



Elando sala

A → 
Gotlando sala

B → 
Riugeno sala



Elando sala

Salų kortelės su instrukcijomis (3/4)



Lolando sala

A →



Saremos sala

B →



Elando sala

Lolando sala



Saremos sala

A →



Bornholmo sala

B →



Hyjunos sala

Saremos sala



Salų kortelės su instrukcijomis (4/4)

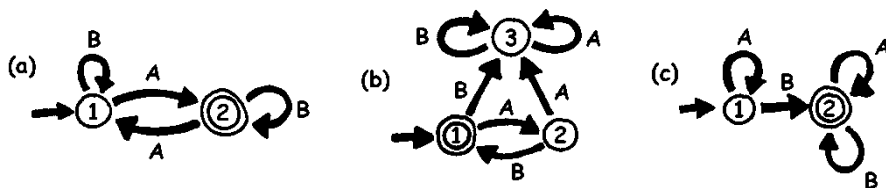


Sveikiname!
Pasiekėte lobio
salą.



Baigtinis automatas

Žemėlapij galima braižyti ir taip:



Šiose schemose salos vaizduojamos apskritimais ir žymimos skaičiais, galutinis tikslas (lobio sala) vaizduojamas dvigubu apskritimu. Kuriais maršrutais galima nuplaukti į lobio salą?

Patarimai

A schemoje kelionė bus baigta tik tada, jei maršrute bus nelyginis laivų A skaičius. Pavyzdžiui: AB, BABAA ar AAABABA.

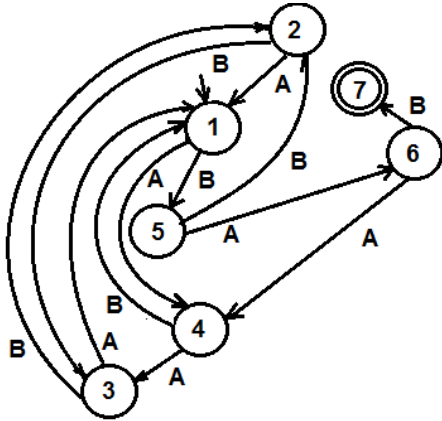
B schemoje pasiekti lobio salą galima tik nuosekliai kaitaliojant laivus A ir B. Pavyzdžiui: AB, ABAB, ABABAB, ...

C schemoje pasiekti lobio salą galima tik tokiais maršrutais, kuriuose yra bent vienas laivas B. Netinkami maršrutai: A, AA, AAA, AAAA, ...

Darbo lapas *Lobio slėpimas*

Ar galėtumėte labai gerai paslėpti lobį? Kur geriausia paslėpti lobį, kad būtų sunku jį surasti? Laikas patiems nubraižyti lobio žemėlapi!

1. Ankstesnės veiklos lobio paieškos žemėlapis pavaizduotas schema ir pateiktas žemiau paveiksle. Šitaip baigtiniai automatai vaizduojami informatikoje.

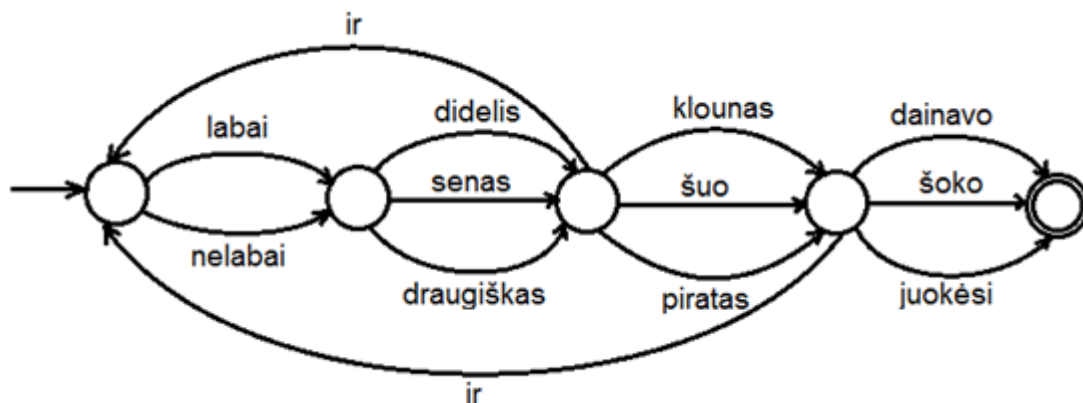


Pavaizduokite savo piratų plaukiojimo maršrutus schema, kaip tai daroma informatikoje. Nupieškite savo salų žemėlapius ir sudarykite salų korteles su savo instrukcijomis. Koks trumpiausias maršrutas, kuriuo pasiekiamas jūsų paslėptas lobis?

2. Išbandykite savo salų žemėlapi ir korteles su draugais. Duokite jiems A ir B sekas ir pažiūrėkite, ar jie suras paslėptą lobį.

Galite sukurti kitokių žaidimų, kurie iliustruoja baigtinio automato idėją.

3. Žemiau paveiksle pavaizduotas sakinių sudarymo būdas taikant baigtinį automatą. Sakiniai sudaromi pasirenkant leidžiamus kelius scheme ir užrašant kelyje pasitaikančius žodžius.



Pabandykite nubraižyti naujų sakinių schemų. Tikriausiai galėtumėte sukurti linksmų istorijų!

Darbo lapas *Paslaptingas monetos žaidimas*

Draugai atsisiuntė žaidimą iš interneto. Žaidime robotas meta monetą ir reikia spėti, kuria puse – herbu ar skaičiumi – ji atsivers. Iš pradžių žaidimas pasirodė labai lengvas. Tikimybė laimėti yra 0,5. Bent jau taip manė draugai. Po kurio laiko draugai pradėjo abejoti. Jie pastebėjo, kad moneta metama pagal tam tikrą taisyklę. Ar žaidime yra kokia apgaulė? Tikriausiai ne! Jie nusprendė tai išsiaiškinti. Vienas iš jų surašė metimų rezultatus. Štai ką jis surašė (H – herbas, S – skaičius):

HSHHSHHHSSHHHHSSHSSSHHHHHSHHHSSSHHHSSSHHHHHHSSHSSSSSHSSHSSSHHH
SSHHHSHHHHHHHHHSSHSSSHHHHHSSSSSSSS

Ar galėtumėte nuspėti, pagal kokią taisyklę metamos monetos?

Galima nubraižyti schemą, pagal kurią gaunama atitinkama metamos monetos atsitiktimų seka.

Užomina: joje yra tik 4 „salos“!



Apie ką visa tai?

Informatikoje baigtinis automatas padeda sugeneruoti ženklų ar įvykių sekas.

Vienas iš baigtinio automato pavyzdžių yra informacijos skambučiai. Kai skambinama informacijos numeriu, iš pradžių atsiliepiama įrašu: „Jei norite ..., spauskite 1, jei norite ..., spauskite 2, jei norite ..., spauskite 3“ ir t. t. Telefono mygtuko paspaudimas yra baigtinio automato kitame telefono linijos gale įvestis. Šis dialogas gali būti labai paprastas arba labai sudėtingas. Kartais galima patekti į baigtinio automato ciklą. Jei taip atsitinka, sistema parodo klaidą, o tai gali nuvilti skambintoją.

Kitas pavyzdys yra bankomatas. Bankomato programa pateikia seką įvykių, iš kurių vartotojas pasirenka reikiamą. Programoje yra visos įmanomos įvykių galimybės kaip baigtinio automato būsenos. Kiekvieno klavišo paspaudimas perkelia į kitą būseną. Kai kurios būsenos pateikia instrukcijas kompiuteriui, pavyzdžiui, „išduoti 100 Eur“, „spausdinti sąskaitą“ arba „išimti kortelę“.

Kai kurie kompiuteriai naudoja schemas (panašias į p. 113), kurios padeda sudaryti anglų kalbos sakinius. Jie gali ir generuoti sakinius, ir vykdyti tai, ką vartotojas įveda. 1960 m. buvo parašyta programa „Eliza“ (pavadinta Elizos Dolittle garbei), kuri gali bendrauti su žmogumi. Programa primena bendravimą su psichoterapeutu, nes vartoja tokias frazes, kaip „papasakokite apie savo šeimą“ ir „prašom tęsti“. Žinoma, ji nieko nesupranta, tačiau yra pakankamai įtikinama. Kai kurie ją naudoję žmonės galvojo, kad kalba su psichoterapeutu.

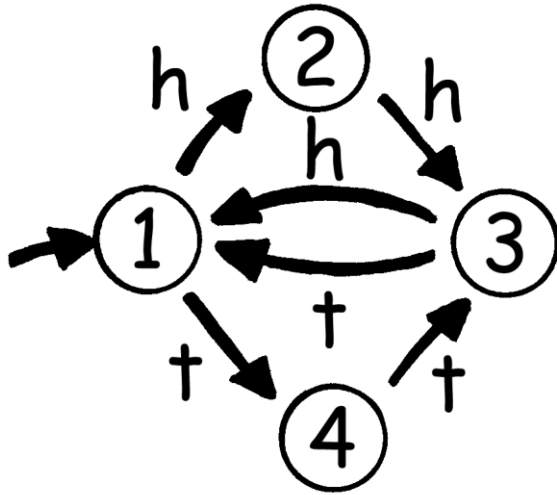
Nors kompiuteriai nesupranta natūralios žmogaus kalbos, tačiau jie lengvai gali apdoroti dirbtines kalbas. Viena iš svarbiausių dirbtinių kalbų grupių yra programavimo kalbos. Naudodamas baigtinį automatą kompiuteris skaito programas ir verčia jas į elementarias instrukcijas, kurias pats gali tiesiogiai įvykdyti.



Sprendimai ir užuominos

Paslaptingas monetos žaidimas

Paslaptingas monetos žaidimas vyksta pagal tokią schemą:



13 veikla

Nurodymų tvarka. Programavimo kalbos

Santrauka

Kompiuteris gali atlikti veiksmus tik pagal algoritmą – programą, užrašytą taikant sutartas taisykles. Programa – tai algoritmas, užrašytas programavimo kalbos žymenimis taip, kad jį būtų galima atlikti kompiuteriu. Kompiuteris supranta kiekvieną komandą paraidžiui. Ši veikla iliustruoja kaip tik šį programavimo aspektą.

Ryšiai su ugdymo programomis

- ✓ Lietuvių kalba: sakomo teksto interpretacija

Gebėjimai

- ✓ Instrukcijų kūrimo ir vykdymo

Amžius

- ✓ Nuo 7 metų

Priemonės

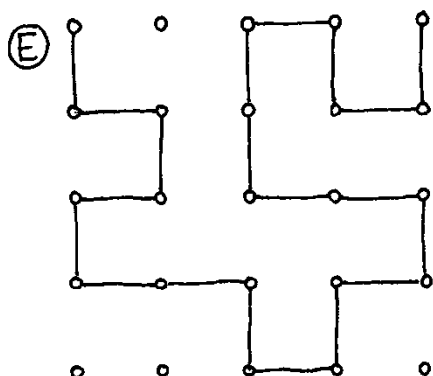
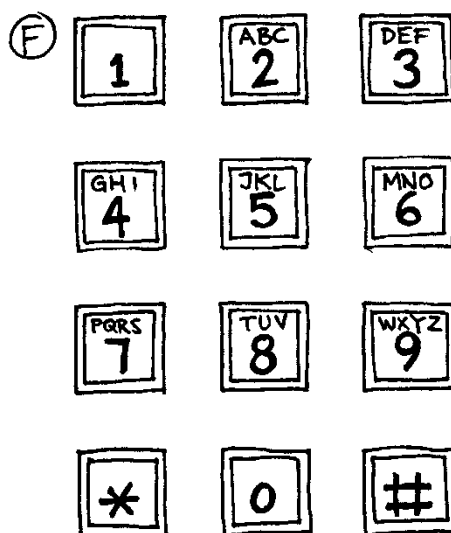
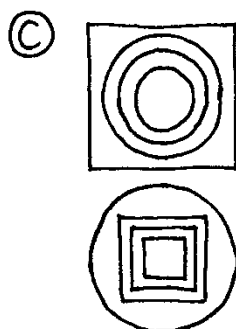
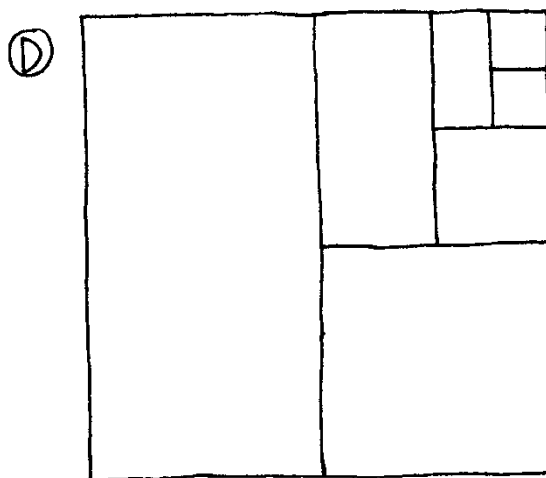
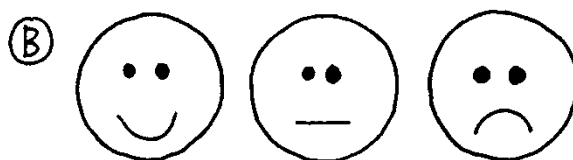
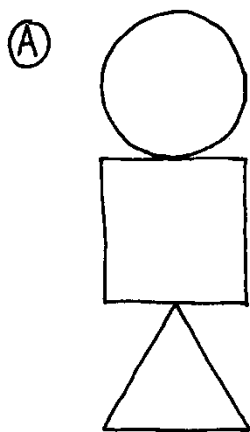
Reikės:

- ✓ Kortelių su paveikslėliais

Kiekvienam mokiniui reikės:

- ✓ Rašymo priemonės, popieriaus lapo ir liniuotės





Nurodymų tvarka

Įvadas

Diskutuojama, kas būtų, jei žmonės tiksliai vykdytų nurodymus. Pavyzdžiui, kaip pasielgtų žmogus, jei paprašytume jo eiti pro uždarytas duris?

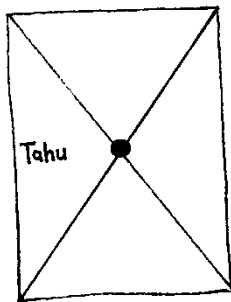
Kompiuteris tiksliai vykdo nurodytas komandas, net jei jos ir neturi jokios prasmės.

Demonstracija

Mokiniai pabando vykdyti tikslius nurodymus:

11. Pažymėkite tašką lapo viduryje.
12. Pradėdami nuo kairiojo viršutinio lapo kampo su liniuote brėžkite tiesią liniją per tašką centre iki apatinio dešiniojo lapo kampo.
13. Pradėdami nuo apatinio kairiojo lapo kampo su liniuote brėžkite liniją per tašką centre iki viršutinio kairiojo lapo kampo.
14. Parašykite savo vardą viduriniame trikampyje lapo kairėje.

Mokinių rezultatas turėtų atrodyti taip:



Veiklos

1. Parenkamas mokinys, kuriam duodamas vienas paveikslėlis pateiktų aukščiau. Mokinys turi sakyti žingsnius, kaip tą paveikslėlį nupiešti. Kiti mokiniai piešia pagal jo nurodymus. Jie gali klausti, jei ko nesupranta. Tikslas – padėti mokiniams suprasti, kaip vykdomi tikslūs nurodymai.
2. Pratimas kartojamas, tačiau mokiniams nebeleidžiama klausti. Patariama naudoti paprastesnį paveikslėlį, nes mokiniai greitai pradeda klysti.
3. Pratimas kartojamas, bet šį kartą nurodymus duodantis mokinys pasislepia, kad piešiantys mokiniai galėtų jį tik girdėti. Mokiniai negali klausinėti.
4. Atkreipiamas dėmesys, kad paskutinis pratimas yra labai panašus į tai, kaip programuotojas rašo programą kompiuteriui. Jis surašo nurodymus ir rezultatą pamato tik tada, kai kompiuteris juos visus įvykdo.

Tegul mokiniai nupiešia piešinį ir parašo to piešinio nurodymus, kurias gali išbandyti poromis ar su visa klase.

Variantai

Mokiniai galėtų:

1. Parašyti popierinio lėktuvėlio lankstymo nurodymus.
2. Parašyti nurodymus, kaip nueiti į tam tikrą vietą mokyklos teritorijoje naudodami frazes: „eik x metrų tiesiai“, „pasisuk dešinėn (90 laipsnių)“ ir „pasisuk kairėn (90 laipsnių)“.

Mokiniai turi tikrinti parašytas instrukcijas su draugais, jas taisyti, kol gaunamas norimas rezultatas.

„Akla višta“. Vienam mokiniui užrišamos akys. Kiti mokiniai nurodinėja, kur šis turi eiti, kad pasiektų tikslą.

Apie ką visa tai?

Kompiuteris gali atlikti tik tiksliai užrašytą algoritmą, vadinamą programa. Programai užrašyti vartojamos programavimo kalbos, sukurtos pagal tam tikras taisykles ir turinčios ribotą skaičių kompiuteriui suprantamų komandų. Vienos kalbos yra tinkamesnės vienam uždaviniui spręsti, kitos – kitam.

Nepriklausomai nuo vartojamos programavimo kalbos, programuotojas turi tiksliai nurodyti, ką kompiuteris turi atlikti. Priešingai nei žmogus, kompiuteris įvykdo visas komandas paraižiu, net jei jos yra visiškai absurdiškos.

Programos turi būti parašytos teisingai, nes net maža klaida gali sukelti daug problemų. Galima tik įsivaizduoti klaidos padarinius erdvėlaivių paleidimo, branduolinės elektrinės ar traukinių valdymo kompiuterių programose. Anglų kalboje šios klaidos vadinamos „bugs“ (liet. *vabalai*). Įdomu tai, kad šis angliškas pavadinimas atsirado dėl kandies, pašalintos (angl. *debugged*) iš 1940 m. elektroninės skaičiavimo mašinos elektros perjungiklio.



Kuo sudėtingesnė programa, tuo didesnė jos klaidų tikimybė. Programos sudėtingumas tapo pagrindiniu iššūkiu, kai JAV buvo kuriama Strateginės gynybos („Žvaigždžių karų“) programos kompiuterių valdymo sistema. Ši sistema turėjo sudaryti neįveikiamą gynybą nuo branduolinės atakos. Kai kurie informatikai tvirtino, kad sistema niekada neveiks, nes programinė įranga yra labai sudėtinga ir nepatikima. Jų pagrindinis argumentas buvo tai, kad programinę įrangą reikia testuoti ir nustatyti kiek įmanoma daugiau klaidų, o šios sistemos ištestuoti neįmanoma, nes norint įsitikinti, kad ji veikia, reikia patikrinti visas jos galimybes, iš kurių viena – paleisti JAV branduolines raketas.

IV dalis

Ko kompiuteris negali?

Sunkiai sprendžiami

uždaviniai



Sunkiai sprendžiami uždaviniai

Ar egzistuoja tokie uždaviniai, kuriuos sunkiai sprendžia net ir kompiuteriai? Žinoma! Galima iškart pateikti pavyzdį: paprastas pokalbis kompiuteriui yra neįveikiamas, nes jis negali suprasti, mąstyti ir pateikti reikšmingų atsakymų. To kompiuteris negali padaryti tik dėl to, kad žmogus pats nežino, kaip šie procesai vyksta, ir negali jų aprašyti kompiuteriui suprantamomis komandomis. Tačiau yra kita grupė kompiuteriui sunkių uždavinių. Šiuos uždavinius parašyti kompiuteriui suprantama programavimo kalba yra nesunku, bet juos kompiuteris spręstų milijonus šimtmečių. Net šimtą kartų spartesnis kompiuteris juos spręstų milijonus metų, o milijoną kartų spartesnis – šimtus metų. Sunkiai sprendžiamas uždavinys kompiuteriui yra toks, kurį net ir sparčiausias kompiuteris sprendžia daug ilgiau, nei trunka žmogaus gyvenimas.

Šioje dalyje pareikiami uždaviniai, neturintys efektyvių sprendimų. Kompiuteris juos spręstų milijonus šimtmečių. Taip pat supažindinama su didžiausia informatikos mokslo paslaptimi: niekas nežino, ar yra efektyvesnis uždavinio sprendimo būdas! Tai gali reikšti, kad nėra įrodymo, kad efektyvus sprendimo būdas egzistuoja, arba kad tokio sprendimo nėra. Informatikoje yra tūkstančiai neišspręstų visiškai skirtingų, tačiau sprendžiamų vienu būdu uždavinių. Jei būtų rastas efektyvus vieno uždavinio sprendimas, tai jį būtų galima pritaikyti ir kitiems uždaviniams spręsti. Šios dalies skyriuose supažindinama su keliais sunkiai sprendžiamais uždaviniais.

Mokytojams

Šioje dalyje aprašomos trys veiklos. Atliekant pirmą veiklą spalvinami žemėlapiai ir skaičiuojama, kiek reikia spalvų kaimyninėms šalims skirtingai nuspalvinti. Užsiimant antra veikla paprastame gatvių žemėlapyje tam tikrose vietose statomi ledų furgonai. Atliekant trečią veiklą virve ir kabliais jungiami taškai į tinklą.

Atliekant šias veiklas stengiamasi parodyti idėjų sudėtingumą: paprasti, aiškūs ir nesunkiai suprantami uždaviniai yra sunkiai sprendžiami. Dažniausiai šie uždaviniai kyla iš paprastų kasdieninio gyvenimo situacijų, kai reikia sudaryti žemėlapius ar tvarkaraščius, projektuoti kelius. Tokie uždaviniai vadinami NP pilnumo uždaviniais. NP pilnumo sąvoka aprašoma skyreliuose „Apie ką visa tai?“ kiekvienos veiklos aprašo pabaigoje. Veiklos turi būti atliekamos iš eilės. Visas veiklas atlikę mokiniai turėtų įgyti supratimą apie svarbiausius atviruosius informatikos klausimus.

Ši dalis pavadinta „Sunkiai sprendžiami uždaviniai“, nes joje aptariami uždaviniai yra labai sunkiai sprendžiami arba visiškai neišsprendžiami. Tokie uždaviniai iš tiesų labai sudėtingi, jiems spręsti reikia labai daug laiko, o jų sprendimai mažai kam suprantami. Vienaip ar kitaip, šie uždaviniai yra labai svarbūs praktiniame gyvenime, jų išsprendimas būtų reikšmingas įvairioms mokslo tyrimų sritims. Pavyzdžiui, daugelis kriptografijos uždavinių remiasi sunkiai sprendžiamais uždaviniais. Jie svarbūs ir dėl to, kad, tarkime, tokį uždavinį išsprendęs nusikaltėlis galėtų iškoduoti slaptus duomenis ir juos parduoti arba, pavyzdžiui, tiesiog klastoti bankų pavedimus. Apie tai bus rašoma V dalyje „Kriptografija“.



14 veikla

Vargšas kartografas. Grafo spalvinimas

Santrauka

Daugelyje optimizavimo uždavinių tam tikri įvykiai negali vykti vienu metu ar tam tikri objektai negali būti gretimi. Pavyzdžiui, kiekvienas pabandęs sudaryti klasės ar susitikimo tvarkaraštį susidūrė su visų dalyvių sąlygų tenkinimo problema. Tokio pobūdžio uždaviniai gali būti sprendžiami taikant grafo spalvinimą. Spalvinant grafą siekiama, kad gretimi jo elementai būtų skirtingų spalvų. Atliekant šią veiklą grafo spalvinimo būdu sudaromas žemėlapis.

Ryšiai su programomis

- ✓ Matematika: skaičiai, dvejetainė skaičiavimo sistema
- ✓ Matematika: algebra, begalinė skaičių seka, skaičių sekos taisyklės, dvejetainis laipsnis

Gebėjimai

- ✓ Problemų sprendimo
- ✓ Loginio pagrindimo
- ✓ Algoritmų vykdymo
- ✓ Bendradarbiavimo ir įžvalgumo

Amžius

- ✓ Nuo 7 metų

Priemonės

- ✓ Lenta arba kitas rašymo paviršius ir rašymo priemonė
- ✓ Kiekvienam mokiniui reikės:
- ✓ Vieno ar kelių darbo lapo kopijų
- ✓ Spalvotų smeigtukų ar spalvotų lipniųjų lapelių
- ✓ Spalvotų pieštukų, flomasterių ar kreidelių



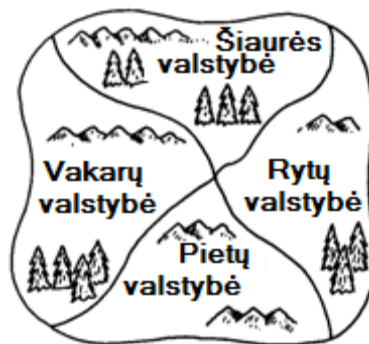
Grafo spalvinimas

Įvadas



Šiame skyriuje pasakojama apie kartografa, kuriam mokiniai turi padėti nuspalvinti valstybes žemėlapyje taip, kad gretimos valstybės būtų skirtingų spalvų.

Pavyzdžiui, paveiksle dešinėje parodytas žemėlapis su keturiomis valstybėmis: Šiaurės, Vakarų, Rytų ir Pietų. Jei Šiaurės valstybė nuspalvinama raudonai, tai Vakarų negali būti spalvinama raudonai, nes nesimatys sienos su Šiaurės valstybe. Vakarų valstybę galima spalvinti, pavyzdžiui, žaliai. Tada ir Rytų valstybė gali būti spalvinama žaliai, nes neturi bendrų sienų su Vakarų valstybe (jei dvi valstybės liečiasi tik viename taške, laikoma, kad jos neturi bendrų sienų). Pietų valstybė gali būti spalvinama raudonai, nes neturi bendrų sienų su Šiaurės valstybe.



Šio pasakojimo kartografas gyvena labai vargingai ir negali įsigyti daug spalvotų pieštukų, todėl užduotyje prašoma naudoti kuo mažiau spalvų.

Diskusija

Mokiniais užduotis paaiškinama lentoje.

Išdalijami pirmieji darbo lapai (p. 127). Žemėlapiui darbo lape nuspalvinti reikia tik dviejų spalvų. Sumažinti spalvų skaičių iki dviejų gali būti sunku daugeliui mokinių. Tačiau ši užduotis yra paprasta, palyginti su žemėlapiais, kuriuose yra daugiau valstybių ir reikia daugiau spalvų.

Spalvindami žemėlapią dviem spalvomis mokiniai gali suformuluoti taisyklę: jei viena valstybė spalvinama viena spalva, tai jos kaimynė turi būti spalvinama kita spalva. Ši

taisyklė kartojama, kol nuspalvinamas visas žemėlapis. Geriausia, kad šią taisyklę mokiniai sugalvotų patys.

Baigusiems spalvinti pirmąjį lapą mokiniams duodamas antrasis Darbo lapas

Galbūt mokiniai, užuot spalvinę, sugalvos naudoti spalvotus smeigtukus arba lipnituosius lapelius. Šis būdas leidžia apsigalvoti ir prireikus valstybę pažymėti kita spalva.

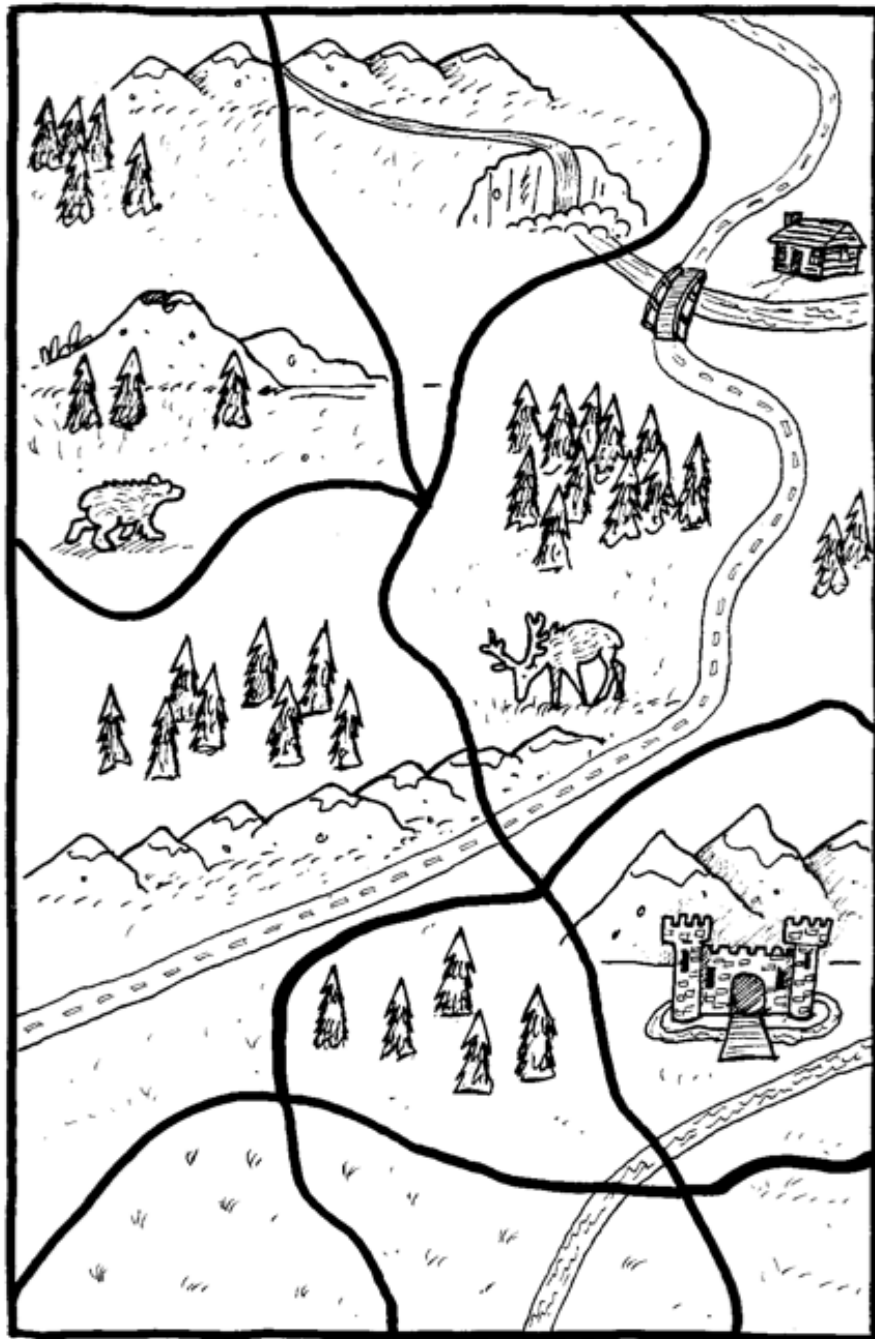
Vyresnių mokinių galima paprašyti paaiškinti, koku būdu spalvinant žemėlapij pakanka mažiausiai spalvų. Pavyzdžiui, šiam žemėlapiui reikia bent trijų spalvų, nes daugiausiai trys valstybės turi bendrų sienų.

Kai mokiniai nuspalvina visus žemėlapius, galima paprašyti, kad sukurtų ir savo žemėlapių, kuriems nuspalvinti reikėtų mažiausiai penkių spalvų, arba kad įrodytų, kad bet kuriam žemėlapiui nuspalvinti reikia mažiausiai keturių spalvų. Tokia užduotis mokinius turėtų užimti ilgesnį laiką. Mokiniai gali greitai nubraižyti žemėlapij, kuriam nuspalvinti reikia penkių spalvų, bet daugeliu atvejų galima parodyti, kad jį galima nuspalvinti ir keturiomis spalvomis.



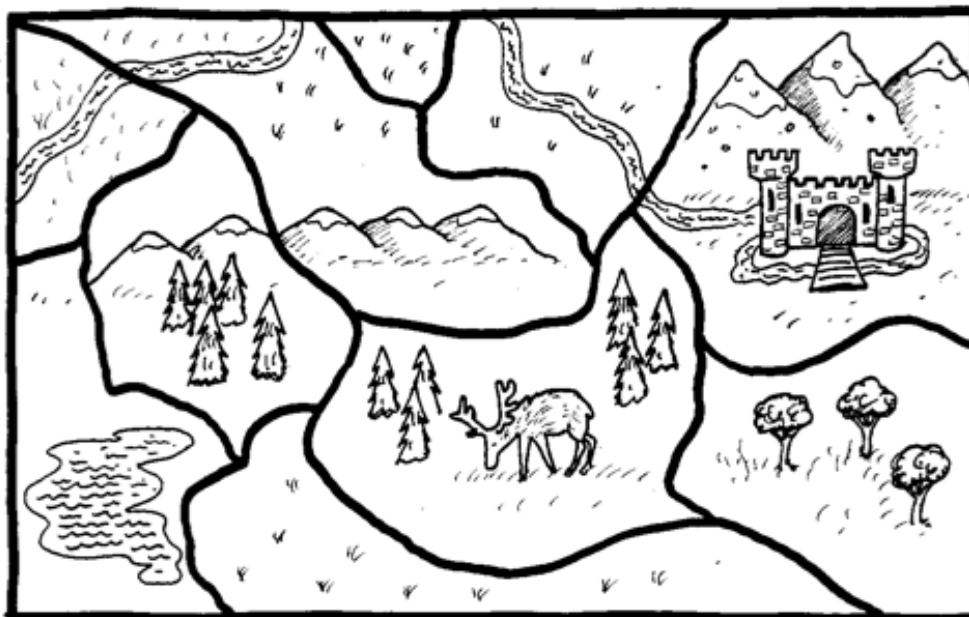
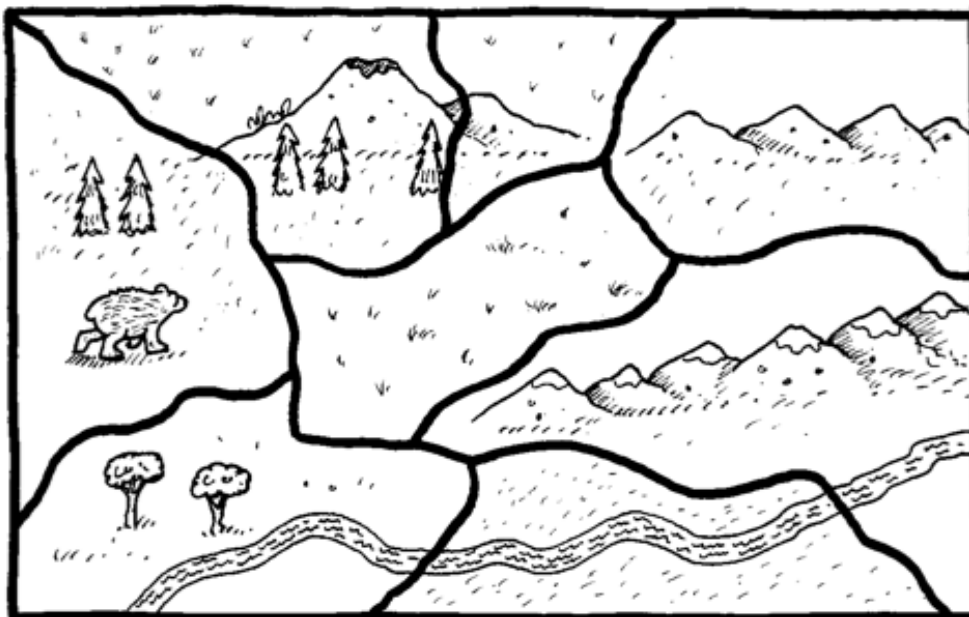
Darbo lapas *Grafo spalvinimas (1)*

Nuspalvinkite valstybes šiame žemėlapyje naudodami kuo mažiau spalvų. Valstybės, turinčios bendrą sieną, negali būti spalvinamos tokia pat spalva.



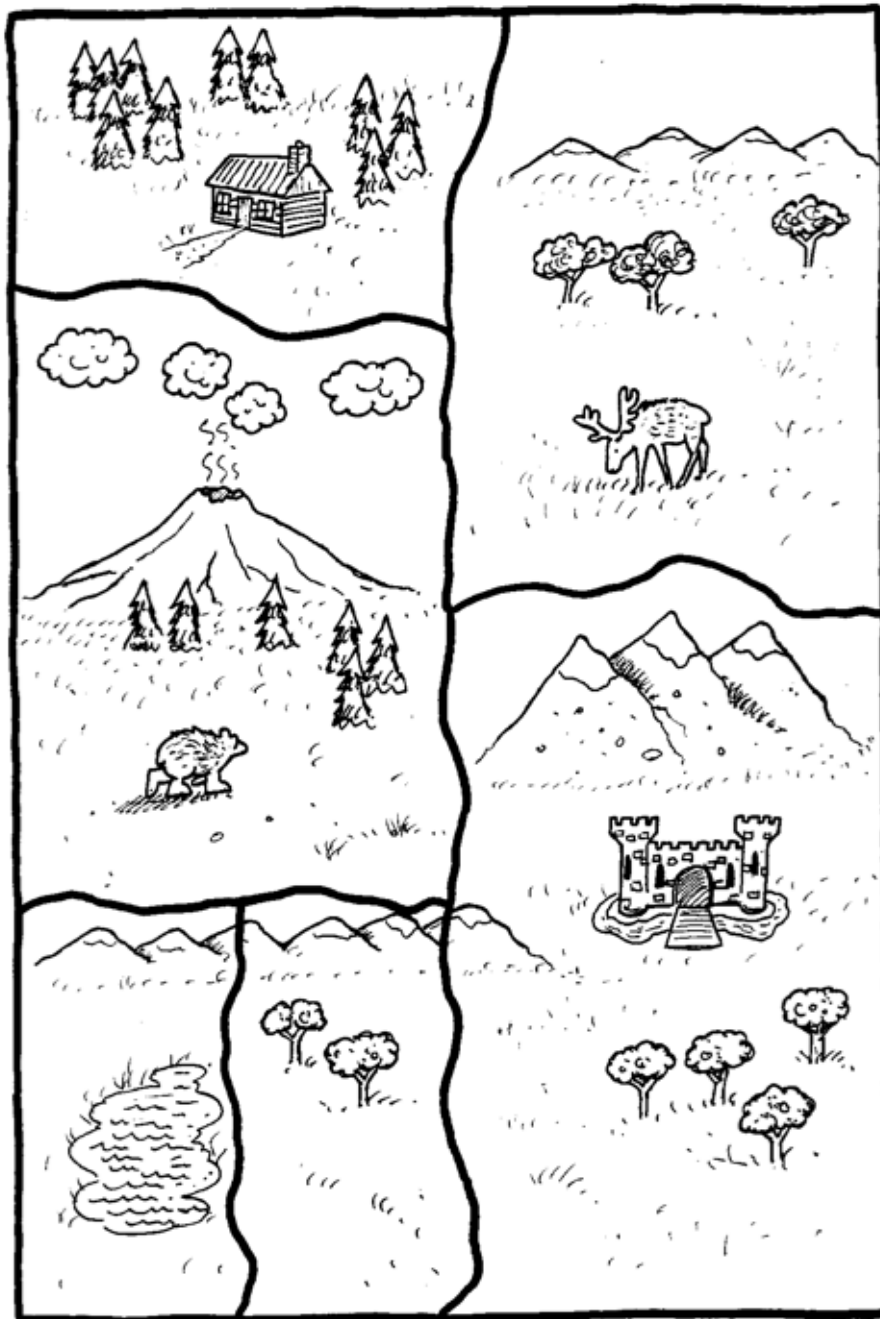
Darbo lapas *Grafo spalvinimas (2)*

Nuspalvinkite valstybes šiuose žemėlapiuose naudodami kuo mažiau spalvų. Valstybės, turinčios bendrą sieną, negali būti spalvinamos tokia pat spalva.



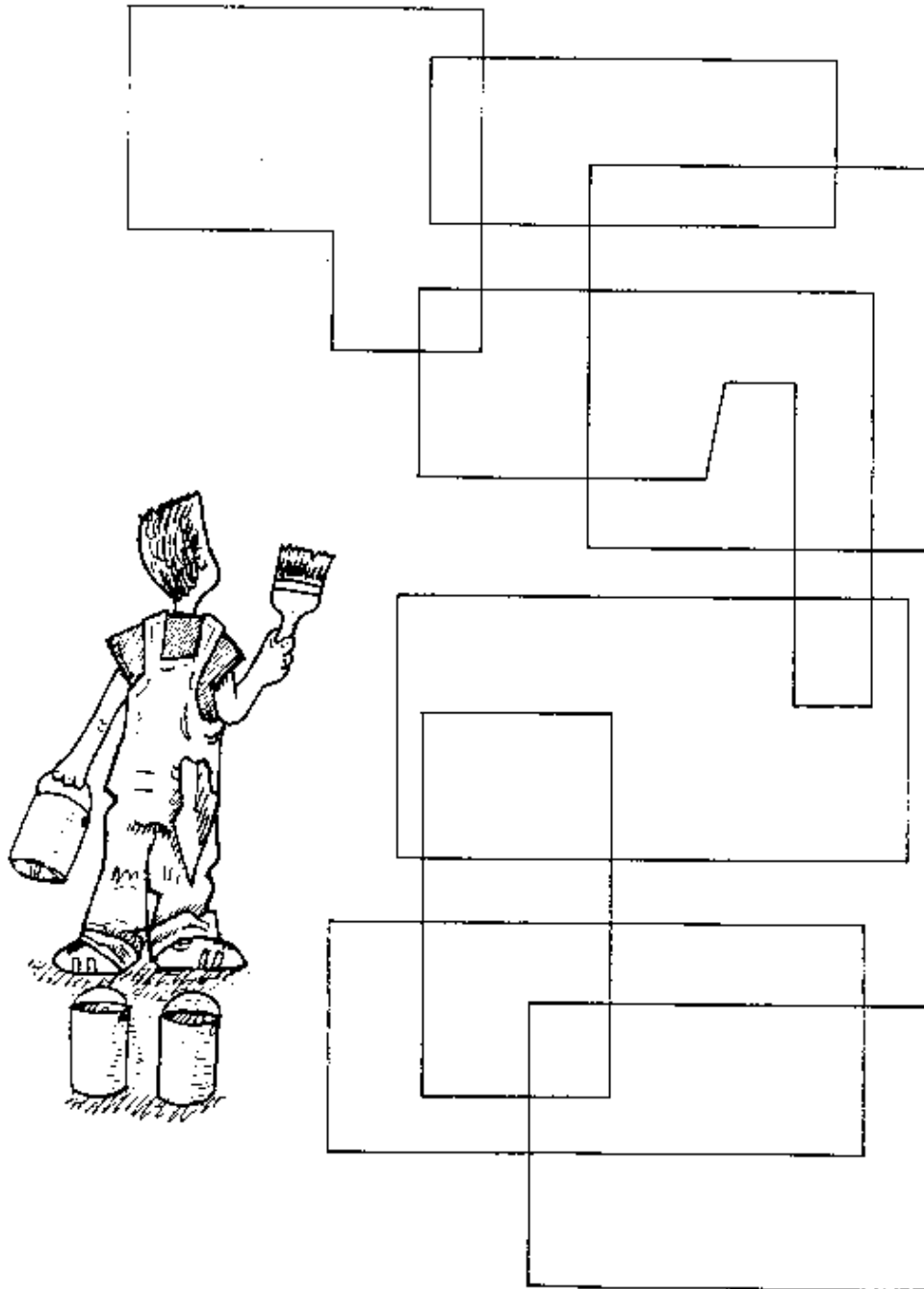
Darbo lapas *Grafo spalvinimas* (3)

Nuspalvinkite valstybes šiame žemėlapyje naudodami kuo mažiau spalvų. Valstybės, turinčios bendrą sieną, negali būti spalvinamos tokia pat spalva.



Darbo lapas *Grafo spalvinimas (4)*

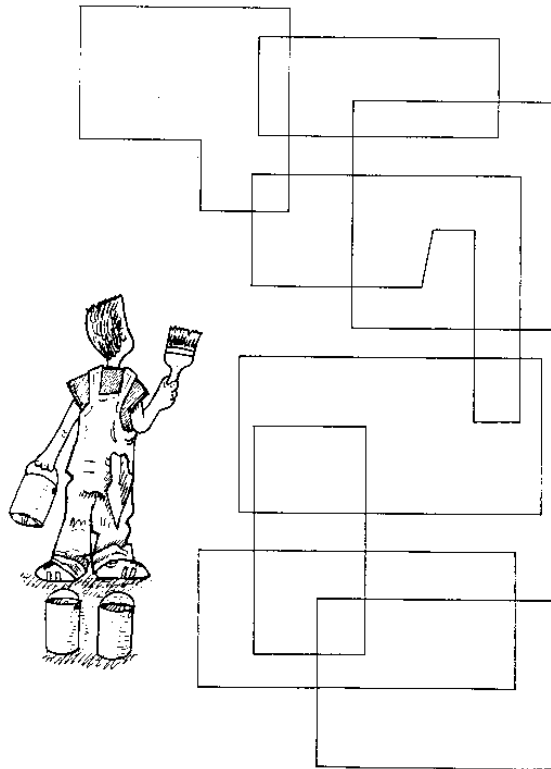
Nuspalvinkite sritis šioje scheme naudodami kuo mažiau spalvų. Sritys, turinčios bendrą ribą, negali būti spalvinamos tokia pat spalva.



Gudručiams

Paprasta sukurti žemėlapij, kuriam nuspalvinti reikia mažiausiai dviejų spalvų. Toks pavyzdys parodytas ketvirtame darbo lape, kur schema yra sudėliota iš susiklojančių uždarytų geometrinių figūrų. Tokiu būdu sudėliotą schemą ar žemėlapij visada bus galima nuspalvinti mažiausiai dviem spalvomis. Mokiniai gali sukurti ir patikrinti keletą tokių žemėlapių.

Žemėlapiui, nupieštam ant lapo ar sferos (gaublio), visada užtenka keturių spalvų. Mokslininkams įdomu, kiek spalvų reikia žemėlapiui, kuris nupieštas ant kitokio paviršiaus, pavyzdžiui, toro. Šiuo atveju visada pakanka penkių spalvų. Mokiniai galėtų tai išbandyti.



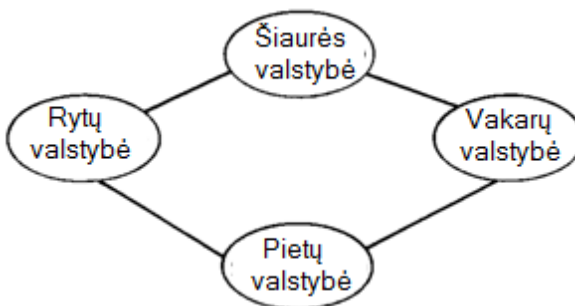
Grafų teorijoje yra įvairių grafo spalvinimo uždavinių interpretacijų. Pavyzdžiui, jei vienas žmogus spalvina žemėlapij, nubraižytą ant lapo, tai tikrai žino, kad protingai spalvinant užtenka keturių spalvų. O jei tas žmogus spalvintų valstybes paeiliui ne vienas, o su priešiška nusiteikusi ar grafų spalvinimo teorijos nežinančiu partneriu? Vienas spalvina viską gerai apgalvodamas, o kitas neapdairiai spalvina valstybes paeiliui. Kiek mažiausiai skirtingų spalvų reikėtų žemėlapiui nuspalvinti? Didžiausias skaičius nėra žinomas. 1992 m. buvo įrodyta, kad 33 skirtingų spalvų visada pakaktų. 2008 m. buvo įrodyta, kad visada pakaktų 17 skirtingų spalvų. Ekspertai spėja, kad turėtų užtekti 10 skirtingų spalvų, bet tai nėra įrodyta. Mokiniai galėtų išbandyti šį uždavinį poromis spalvindami žemėlapius.

Kita grafo spalvinimo uždavinio interpretacija yra *imperijos spalvinimas*. Šiai interpretacijai reikia dviejų skirtingų, bet vienodą skaičių valstybių turinčių žemėlapių. Kiekviena valstybė viename žemėlapyje (Žemės) turi atitikti valstybę kitame žemėlapyje (Žemės valstybės koloniją Marse). Spalvinant šiuos žemėlapius keliamos dvi sąlygos: valstybės, turinčios bendrą sieną, negali būti tokios pat spalvos ir kiekviena valstybė viename žemėlapyje turi būti nuspalvinta tokia pat spalva kaip ją atitinkanti valstybė kitame žemėlapyje. Kiek mažiausiai skirtingų spalvų reikia šioms žemėlapiams nuspalvinti? Atsakymas dar nežinomas.

Apie ką visa tai?

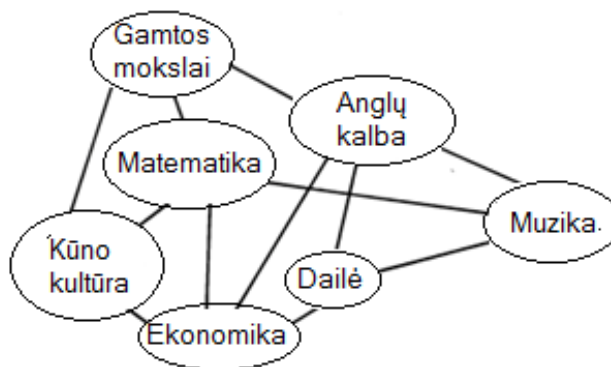
„Žemėlapio spalvinimo“ uždavinyje, spręstame atliekant šią veiklą, reikėjo surasti mažiausią skaičių skirtingų spalvų (dvi, trys, keturios spalvos), kuriomis būtų galima nuspalvinti vieną ar kitą žemėlapij. 1852 m. buvo suformuluotas teiginys, kad bet kokį žemėlapij galima nuspalvinti mažiausiai keturiomis skirtingomis spalvomis, bet iki 1976 m. jis nebuvo įrodytas. Ši keturių spalvų teorema buvo įrodyta tik po 120 metų, nes informatikos mokslas turėjo ir turi daugybę neatsakytų klausimų.

Žemėlapio spalvinimas priklauso grafo spalvinimo uždavinių grupei. Informatikoje grafas yra abstrakti objektų ryšių sistema, ji parodyta paveiksle.



Informatikoje grafai vaizduojami apskritimais arba taškais (vadinamais viršūnėmis), sujungtais briaunomis arba lankais, kurie rodo tam tikrus ryšius tarp vaizduojamų objektų. Paveiksle pateikiamas demonstracijos žemėlapij vaizduojantis grafas. Viršūnės vaizduoja valstybes, o linijos – valstybių sienas. Grafe spalvinimo uždavinys formuluojamas taip: viršūnės turi būti sujungtos skirtingų spalvų linijomis. Spalvinant grafą neribojamas skirtingų spalvų skaičius, nes grafo viršūnės gali būti jungiamos daugeliu būdų. Dvimatis vaizdas sumažina jungimo būdų skaičių. Grafo spalvinimo uždavinys yra surasti mažiausią skaičių skirtingų spalvų tam tikram grafui nuspalvinti.

Paveiksle dešinėje viršūnės atitinka mokykloje mokomus dalykus. Linija, jungianti dvi viršūnes, reiškia, kad mažiausiai vienas mokinys mokosi tų dviejų dalykų, todėl tvarkaraštyje abiejų dalykų pamokos negali vykti vienu metu. Optimalaus pamokų tvarkaraščio sudarymas yra grafo spalvinimo uždavinys, kai kiekviena spalva reiškia kitą pamokos laiką.



Informatikos grafų spalvinimo algoritmas dažnai naudojamas realaus gyvenimo problemoms spręsti, bet veikiausiai netaikomas žemėlapiams spalvinti. Vargšo kartografo veikla tik fikcija. Tokie uždaviniai sprendžiami praktiškai, nes realiame gyvenime galime naudotis gal ir optimaliu, bet vis tiek labai geru rezultatu.

Kompiuterio sugaištas laikas grafų spalvinimo uždaviniui spręsti ilgėja eksponentiškai didėjant grafo dydžiui. Žemėlapio spalvinimo uždavinys gali būti sprendžiamas perrenkant visus įmanomus spalvinimo variantus. Žinoma, kad pakanka mažiausiai keturių skirtingų spalvų. Vadinasi, reikia patikrinti visas žemėlapio spalvinimo keturiomis spalvomis galimybes. Jei žemėlapyje yra n valstybių, tai galimybių yra 4^n . Šis skaičius auga labai greitai: kiekviena šalis padidina galimybių skaičių 4 kartais ir tokiu būdu keturgubina sprendimo laiką. Net jei būtų toks kompiuteris, kuris galėtų

rasti 50 valstybių spalvinimo sprendimą per valandą, tai 51 šaliai nuspalvinti reikėtų jau 4 valandų. Tokie uždaviniai niekur nedingsta, nes išrandami vis spartesni kompiuteriai.

Grafo spalvinimas yra geras pavyzdys uždavinio, kurio sprendimo laikas ilgėja eksponentiškai. Paprastam atvejui, kai mažai valstybių, surasti optimalų sprendimą yra gana paprasta ir be kompiuterio, bet daugėjant valstybių, uždavinys tampa nebeįsprendžiamas net ir kompiuteriu.

Realiam gyvenime dažnai tenka spręsti tokio pobūdžio uždavinius. Nors jie ir sunkiai sprendžiami, realiam gyvenime reikia gauti rezultatą. Informatikos metodai pateikia gana gerus, bet ne puikius sprendimus. Šie euristinių metodų rezultatai yra labai artimi optimaliems, greitai apdorojami kompiuteriu ir tinkami praktiniams tikslams. Pavyzdžiui, mokyklose gali prireikti vieno kabineto daugiau, nei būtina, dėl mažos pamokų tvarkaraščio paklaidos, taip pat ir žemėlapyje gali būti viena spalva daugiau, nei būtina.

Niekas neįrodė, kad nėra efektyvaus būdo šiems uždaviniams spręsti, bet taip pat niekas neįrodė, kad jis yra. Informatikos mokslininkai lieka skeptiški, kad kada nors bus rastas efektyvus metodas. Kitų veiklų skyriuose daugiau bus pasakojama apie sunkiai sprendžiamus uždavinius.

Daugiau informacijos

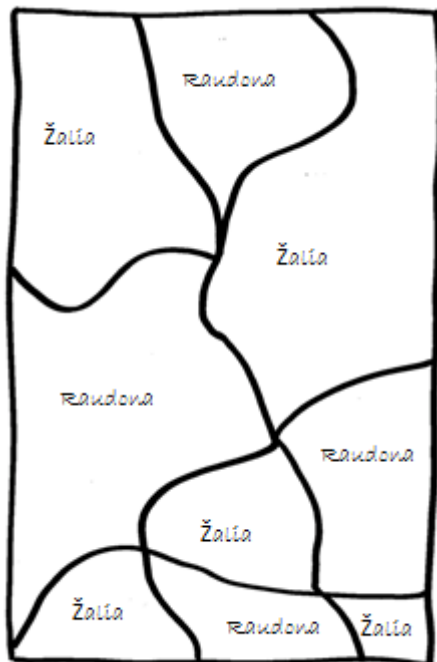
Davidas Harel'o knygoje „Algorithmics“ nagrinėjama keturių spalvų teorema, pateikiama jos istorija. Daugiau žemėlapių spalvinimo aspektų aptariama (The Spirit of Computing, Addison–Wesley, Reading, MA, 1st edition, 1987; 2nd edition, 1992. 3rd edition (with Y. Feldman), 2004)

Nancy Casey ir Mike'o Fellowso knygoje „This is MEGA–Mathematics!“ (Los Alamos National Labs, 1992).

2004 m. Mareko Kubale knygoje „Graph Colorings“ aprašoma grafo spalvinimo uždavinių istorija. Šia tema galima rasti daug tinklalapių. (American Mathematical Society, Providence, Rhode Island, 2004)



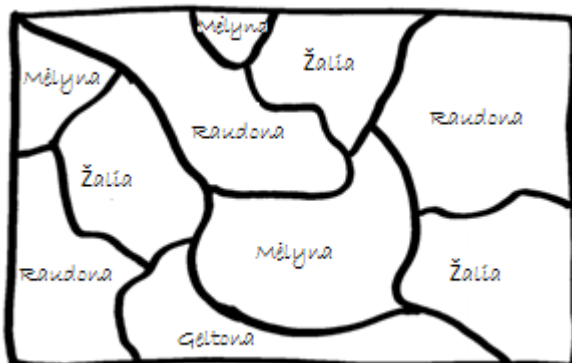
Sprendimai ir užuominos



A.



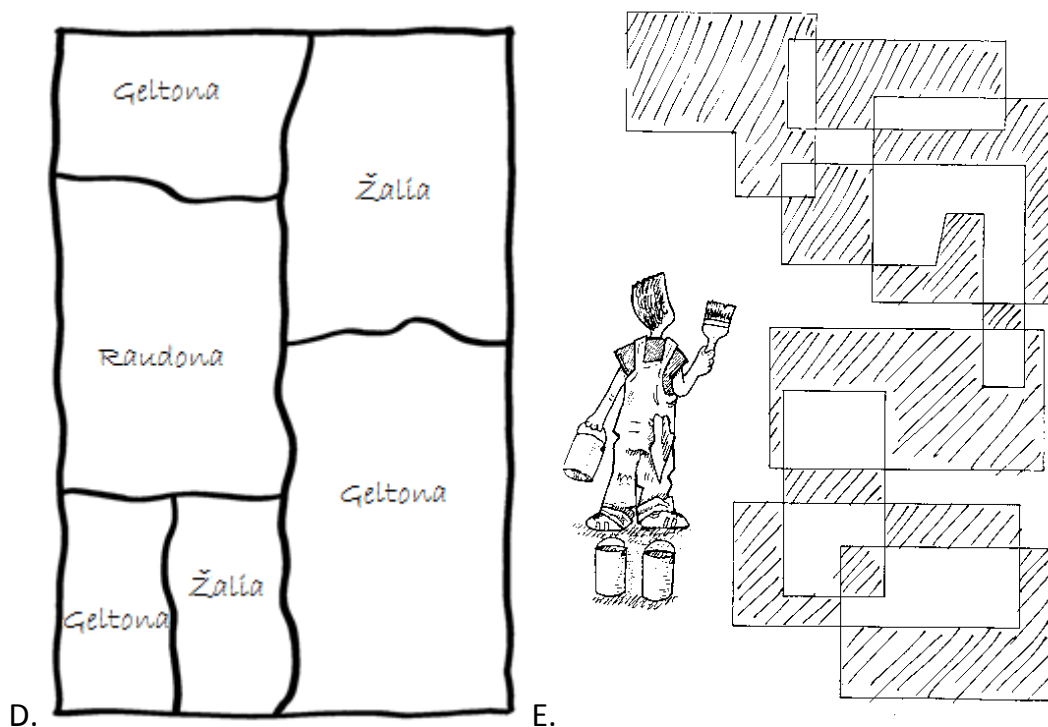
B.



C.

Pirmojo darbo lapo vienintelis sprendimas parodytas A paveiksle (spalvos gali būti pasirenkamos).

Antrojo darbo lapo viršutiniam žemėlapiui reikia trijų spalvų. Galimas sprendimas parodytas B paveiksle. Apatiniam žemėlapiui reikia keturių spalvų. Galimas sprendimas parodytas C paveiksle.



Trečiojo darbo lapo žemėlapiui reikia trijų spalvų. Galimas sprendimas parodytas D paveiksle.

Ketvirtojo darbo lapo sprendimas parodytas E paveiksle (užbrūkšniuotos ir baltos figūros).

15 veikla

Turistų miestas. Dominuojančios sekos

Santrauka

Daugelis realaus gyvenimo situacijų gali būti vaizduojamos abstrakčiai – grafais ar tinklais. Tinklai ir grafai suteikia daug galimybių parašyti naujų naudingų algoritmų. Atliekant šią veiklą siekiama pažymėti susikirtimus arba grafo viršūnes taip, kad kiekviena viršūnė būtų nutolusi ne daugiau kaip per vieną žingsnį nuo kurios nors iš kitų pažymėtų viršūnių. Kiek reikia mažiausiai pažymėti tokių viršūnių?

Ryšiai su ugdymo programomis

- ✓ Matematika: vieta ir padėtis
- ✓ Matematika: loginiai uždaviniai

Gebėjimai

- ✓ Žemėlapių skaitymo
- ✓ Ryšių nustatymo
- ✓ Dėlionės sprendimo
- ✓ Pasikartojančio tikslo siekimo

Amžius

- ✓ Nuo 7 metų

Priemonės

Kiekvienai mokinių grupei reikės:

- ✓ Darbo lapo „Ledų furgonas“
- ✓ Dviejų spalvų žymiklių ar spalvotų smeigtukų

Reikės:

- ✓ Projektoriaus ir baltos lentos, kad būtų galima demonstruoti medžiagą, arba tik baltos lentos, kurioje būtų galima paaikškinti medžiagą visai grupei.



Dominuojančios sekos

Įvadas

„Ledų furgono“ darbo lape grafu vaizduojamas Turistų miesto žemėlapis. Briaunos arba lankai jame vaizduoja gatves, o viršūnės yra gatvių sankryžos. Miestas yra labai karštoje šalyje. Vasarą ledų furgonai pastatomi prie gatvių sankryžų, čia parduodami ledai turistams. Furgonai turi būti sustatyti taip, kad bet kas, gyvenantis prie gatvės, pasiektų furgoną vos tik paėjęs iki gatvės kampo, o gyvenantis name prie gatvių sankryžos pasiektų furgoną paėjęs daugiausiai vieną kvartalą. Kiek mažiausiai furgonų reikia ir kuriose vietose jie turi būti pastatyti?

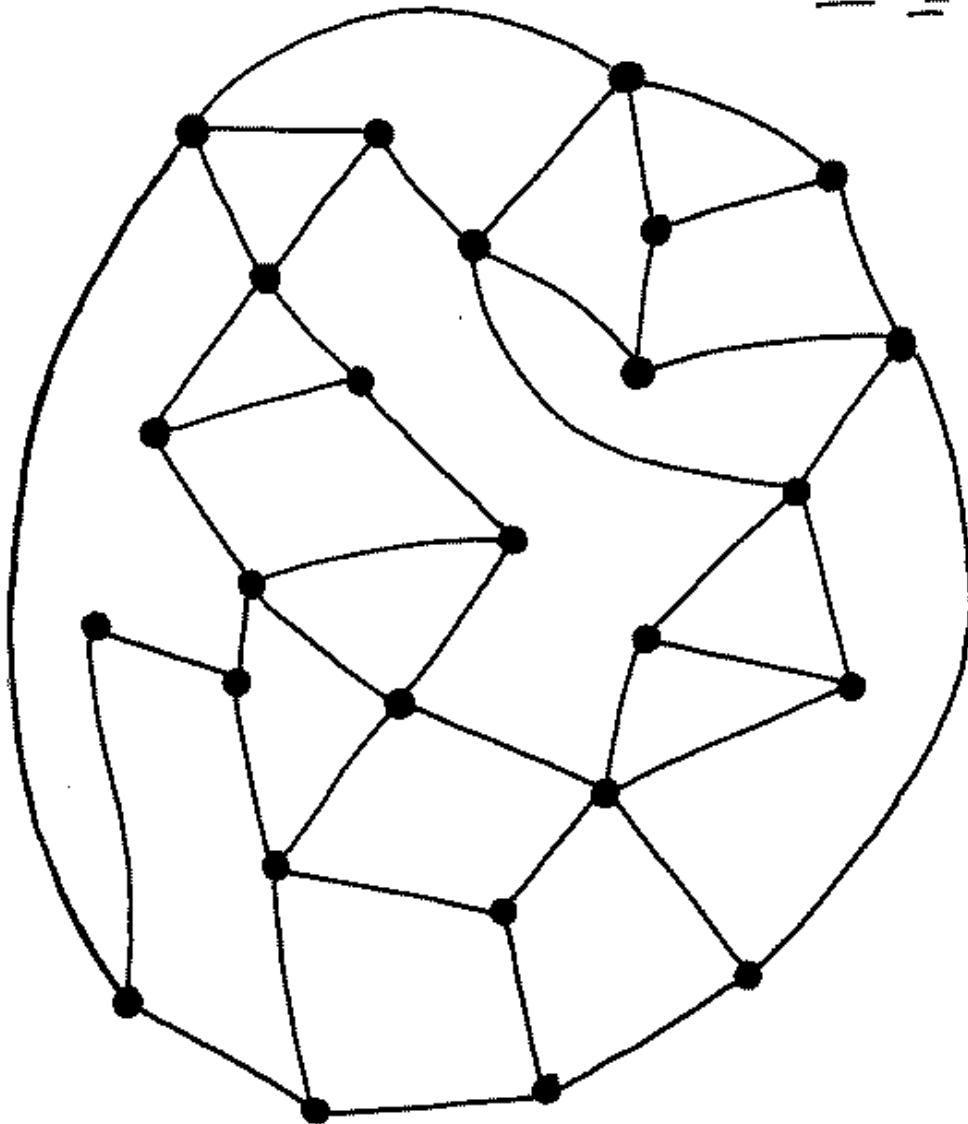
Diskusija ir veiklos

3. Mokiniai pasiskirsto mažomis grupėmis. Kiekvienai grupei duodamas Turistų miesto žemėlapis ir keletas žymiklių. Paaiškinama užduotis.
4. Mokiniams paaiškinama, kaip pažymėti sankryžą, kur statomas ledų furgonas. Kitos spalvos žymikliu pažymimos sankryžos per vieną gatvę nuo furgono. Ledų furgonas skirtas žmonėms, gyvenantiems prie tų sankryžų ar gatvėse, kurios susikerta tose pažymėtose sankryžose.
5. Mokiniai eksperimentuodami žymi skirtingas ledų furgonų vietas. Kai mokiniai sudėlioja furgonus taip, kad jų vieta patogi visiems gyventojams, primenama, kad furgonai yra brangūs, todėl jų turi būti kuo mažiau. Akivaizdu, užduoties sąlygos tenkinamos, kai kiekvienoje sankryžoje pastatoma po furgoną, bet įdomesnis klausimas yra, kiek furgonų galima pašalinti, kad likusių vis tiek užtektų visiems gyventojams.
6. Šešių furgonų pakanka visiems Turistų miesto gyventojams. Sprendimas parodytas paveiksle dešinėje, tačiau šį sprendimą labai sunku surasti! Jei mokiniams sunku surasti mažiausią furgonų skaičių, palaukus kiek laiko galima jiems tą skaičių pasakyti ir paprašyti furgonus pažymėti grafe. Tačiau ir tai gali būti gana sunku. Mokiniams nelengva surasti net ir sprendimą su aštuoniais ar devyniais furgonais.
7. Galima paaiškinti mokiniams, kaip braižomas toks žemėlapis (žr. šios veiklos sprendimus ir užuominas). Mokiniai gali pabandyti sukurti savo Turistų miestą parodytu metodu. Vieniems gali būti įdomu sukurti lengvai išsprendžiamą dëlionę, o kitiems ir tai gali būti labai sunku. Šis pavyzdys vadinamas *vienos krypties funkcija*: lengva pasiūlyti sunkiai sprendžiamą dëlionę, kai žinomas jos kūrimo nuo sprendinio metodas. Vienos krypties funkcijos yra kriptografijos pagrindas (17 ir 18 veiklos).



Darbo lapas *Ledų furgonai*

Pažymėkite, kuriose sankryžose turėtų stovėti ledų furgonai, kad būtų patogu visiems Tūristų miesto gyventojams.



Gudručiams

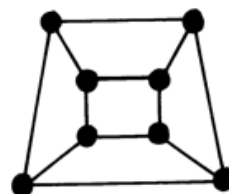
Realiam gyvenime „Ledų furgonų“ uždavinys pasitaiko gana dažnai, kai reikia nustatyti pašto punktų, gaisrinių ir kitų reikalingų objektų vietas, kad būtų galima greitai aptarnauti visus gyventojus. Tačiau realiam gyvenime negalima pasinaudoti gudrybe, aprašyta šios veiklos skyriuje, ir lengvai išspręsti problemą. Ką daryti?

Galima pasitelkti visiškai nesudėtingą būdą: perrinkti visus įmanomus ledų furgono pastatymo variantus ir išrinkti geriausią. Kadangi Turistų mieste yra 26 gatvių sankryžos, tai yra 26 vietos ledų furgonui pastatyti. Nesudėtinga patikrinti visas 26 galimybes. Akivaizdu, kad nė viena iš jų netenkina uždavinio sąlygų. Dviem ledų furgonams pastatyti turime $26 \times 25 = 650$ galimybių, nes, aišku, du furgonai nebus statomi vienoje vietoje. Patikrinti kiekvieną galimybę atskirai nėra sunku, bet patikrinti visas – varginantis ir nuobodus darbas. Be to, užtektų patikrinti pusę galimybių (325), nes nesvarbu, kuris furgonas pastatomas sankryžoje: jei patikrinamas 1 furgonas A sankryžoje, o 2 furgonas B sankryžoje, tai nebereikia tikrinti 2 furgono A sankryžoje ir 1 furgono B sankryžoje. Pastatyti tris furgonus būtų 2600 galimybių, keturis – 14950 galimybių ir t. t. Aišku, 26 furgonų tikrai pakaktų, nes yra tik 26 sankryžos ir nėra prasmės statyti po du furgonus vienoje sankryžoje.

Kitas ledų furgonų vietų nustatymo būdas – suskaičiuoti visas įmanomas galimybes, kai sankryžos yra 26, o furgonų skaičius bet koks. Kadangi kiekvienai sankryžai yra tik dvi galimybės (yra pastatytas furgonas arba jo nėra), tai galimybių skaičius yra $2^{26} = 67108864$. Šis sprendimo metodas vadinamas *visiško perrinkimo algoritmu* ir užima daug laiko. Nors atrodo, kad kompiuteriai yra gana spartūs ir gali atlikti šią užduotį gana greitai, iš tikrųjų net ir spartus kompiuteris ilgai ieško sprendinio. Visiško perrinkimo algoritmo atlikimo sparta priklauso nuo to, kaip greitai patikrinama, ar galimybė yra uždavinio sprendinys, t. y. tenkina visas sąlygas. Įdomu, kiek laiko galėtų sugaišti kompiuteris tikrindamas visas 2^{26} galimybių. Jei patikrinti vieną galimybę, ar ji yra sprendinys, užtruktų 1 s, tai patikrinti visas galimybes būtų sugaišta apie 67 mln. sekundžių, arba apie 777 dienų, arba dvejus metus. Net jei viena galimybė būtų tikrinama tik 0,001 s, tai per tuos dvejus metus kompiuteris galėtų patikrinti tik 2^{36} galimybių. Jei kompiuteris būtų milijoną kartų spartesnis ir milijoną galimybių patikrintų per 1 s, tai per dvejus metus patikrintų miestą tik su 46 sankryžomis. O tai tikrai nedidelis miestelis. Kiek sankryžų yra jūsų mieste?

Jei visiško perrinkimo algoritmas toks lėtas, gal yra koks kitas šio uždavinio sprendimo būdas? Galima pabandyti taikyti „godumo“ principą. Pirmasis ledų furgonas statomas sankryžoje, kurioje susikerta daugiausia gatvių, antrasis – kitoje sankryžoje, kurioje vėl susikerta daugiausia gatvių ir t. t. Tačiau šio metodo rezultatas nebūtinai bus mažiausias furgonų skaičius, nes sankryža su didžiausiu gatvių skaičiumi gali būti ir ne pati geriausia vieta furgonui pastatyti pagal uždavinio sąlygas. Tai galima patikrinti kartu su mokiniais.

Aiškintis galima pradėti nuo paprastesnės užduoties. Mokiniais pasakomas furgonų skaičius ir pasiūloma išsiaiškinti, ar jis yra mažiausias. Kai kuriais atvejais tai visai nesunku. Pavyzdžiui, grafe dešinėje vaizduojamas daug paprastesnis miestas. Jei gatvės įsivaizduojamos kaip kubo



briaunos, akivaizdu, kad du ledų furgonai gali būti pastatyti priešinguose kubo įstrižainės galuose. Be to, būtų galima įsitikinti, kad vieno ledų furgono tikrai nepakanka šiame mieste. Taip pat galima pabandyti įsitikinti, kad ir Turistų mieste negali būti mažiau kaip 6 furgonai.



Apie ką visa tai?

Įdomu tai, kad kol kas nėra kito „Ledų furgonų“ uždavinio sprendimo metodo nei visiškas perrinkimas. Visiško perrinkimo algoritmo atlikimo laikas ilgėja eksponentiškai didėjant sankryžų skaičiui. Toks algoritmas vadinamas *eksponentinio laiko algoritmu*. Informatikoje žinomas ir *polinominio laiko* sprendimo algoritmas, kurio laikas ilgėja sankryžų skaičiaus laipsniais (kvadratu, trečiuoju ir t. t.). Toks algoritmas yra greitesnis nei eksponentinis algoritmas, kai reikia sudaryti didelius žemėlapius. Pavyzdžiui, kai viršūnių skaičius n yra didesnis už 117, tai n^{17} yra mažesnis už 2^n . Ar yra polinominio laiko algoritmas „Ledų furgonų“ uždaviniui spręsti? Dar nėra, nors informatikai stengiasi jį sukurti. Lengvesnis uždavinys, kai reikia patikrinti, ar tam tikras furgonų skaičius yra mažiausias, taip pat sprendžiama visiško perrinkimo algoritmu. *Polinominio laiko* algoritmas šiam uždaviniui spręsti dar nėra sukurtas.

Ar šis uždavinys neprimena žemėlapių spalvinimo (13 veikla)? Turėtų! „Ledų furgonų“ uždavinys priklauso vadinamajai *minimalios dominuojančios sekos* (angl. *minimum dominating set*) uždavinių grupei, kurią sudaro tūkstančiai uždavinių, pavyzdžiui, žemėlapių spalvinimas, optimalaus maršruto paieška, tvarkaraščio sudarymas ir kt. Dar nėra žinoma, ar galima sukurti *polinominio laiko* algoritmus visiems šiems uždaviniams. Jei būtų sukurtas *polinominio laiko* algoritmas bent vienam iš šių uždavinių spręsti, jis galėtų būti pritaikytas ir visiems likusiems šio tipo uždaviniams.

Šie uždaviniai vadinami *NP pilnumo* uždaviniais. NP yra termino *nedeterministinis polinomas* santrumpa. Tai reiškia, kad uždavinys galėtų būti išspręstas per priimtina laiką, jei kompiuteris iškart galėtų apdoroti gana daug sprendinių (tai ir yra nedeterministinė dalis). Deja, neįmanoma sukurti tokio kompiuterio, nes jis turėtų būti labai didelis. Tačiau tokio kompiuterio konceptas yra iš esmės svarbus, nes NP pilnumo uždaviniai negali būti išspręsti per priimtina laiką neturint nedeterministinio kompiuterio.

Šie uždaviniai vadinami *pilnumo*, nes rastą vieno uždavinio sprendimą būtų galima pritaikyti ir kitiems tokiems uždaviniams, nors patys uždaviniai atrodo labai skirtingi: žemėlapių spalvinimas skiriasi nuo ledų furgonų statymo ir t. t. Šie uždaviniai visi kartu gali būti išspręsti arba ne.

Daugiau informacijos

Davidas Harelas knygoje „*Algorithmics*“ pateikia keletą NP pilnumo uždavinių ir samprotauja, ar polinominio laiko algoritmas jiems spręsti egzistuoja. (The Spirit of Computing, Addison–Wesley, Reading, MA, 2nd edition, 1992. 3rd edition, 2004 (with Y. Feldman)).

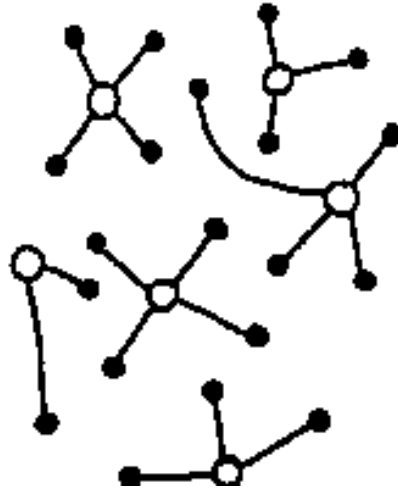
Alexanderis K. Dewdney knygoje „*Turing Omnibus*“ taip pat aptaria NP pilnumą. (Press, New York, 1989).

Michaelas R. Garey ir Davidas S. Johnsonas knygoje „*Computers and Intractability: A guide to the theory of NP-completeness*“ nagrinėja kelis šimtus NP pilnumo uždavinių ir NP pilnumo įrodymo metodų. (W. H. Freeman and co., New York, 1979).



Sprendimai ir užuominos

Turistų miesto žemėlapis pradedamas braižyti nuo šešių viršūnių, kur turėtų stovėti po vieną ledų furgoną. Iš kiekvienos viršūnės brėžiamos gatvės, jų gale pažymimos viršūnės, kaip parodyta paveiksle žemiau.



Laisvos viršūnės sujungiamos briaunomis arba lankais, kaip parodyta paveiksle žemiau.



16 veikla

Ledo kelias. Šteinerio medžiai

Santrauka

Kartais tik šiek tiek pakeista paprasta užduotis tampa labai sunkiai sprendžiama. Šiame skyriuje nagrinėjama trumpiausio kelio paieškos užduotis, panašiai kaip skyriuje „Purvinas miestas“. Tačiau šio skyriaus užduotyje leidžiama pridėti naujų taškų, jei jie sutrumpina bendrą kelią. Užduotis tampa daug sunkesnė ir nesusijusi su „Purvino miesto“ užduotimi, bet algoritmavimo požiūriu atitinka „Žemėlapių spalvinimo“ ir „Ledų furgonų“ užduotį.

Ryšiai su ugdymo programomis

- ✓ Matematika: vieta ir orientacija
- ✓ Matematika: logika

Gebėjimai

- ✓ Vizualizavimo
- ✓ Geometrinio pagrindimo
- ✓ Algoritmavimo

Amžius

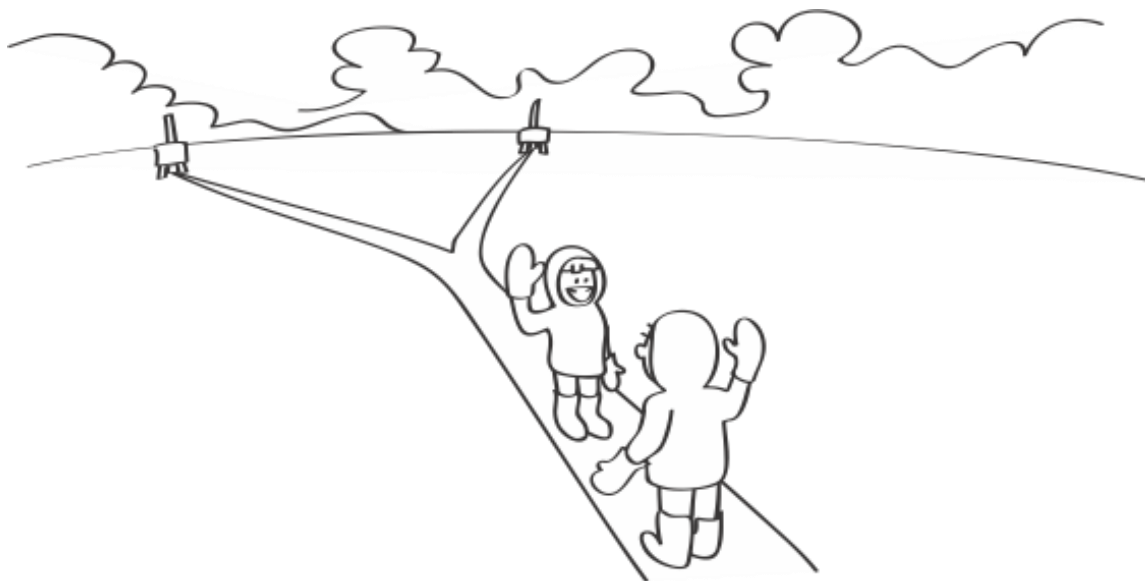
- ✓ Nuo 7 metų

Priemonės

Kiekvienai mokinių grupei reikės:

- ✓ Penkių ar šešių kuolelių (tinka palapinės kuoleliai)
- ✓ Kelių metrų virvės (mažiausiai 10 m)
- ✓ Ruletės
- ✓ Rašymo priemonės ir popieriaus lapo pastaboms

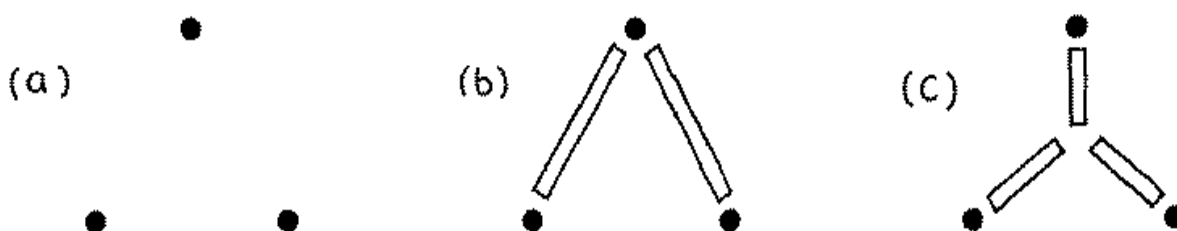
Ledo kelias



Ivadas

Šiaurinėje Kanados dalyje keletas gręžinių yra įrengti aplink didelį ežerą. Žiemą ežeras užšąla, jį storai užkloja sniegas. Gręžinių darbuotojai sniego valymo mašina turi pravalyti kelius per ežerą, kad galėtų susisiekti su kitomis brigadomis. Šiaurinėje Kanados dalyje žiemos labai šaltos, todėl bendras kelio ilgis turi būti kuo trumpesnis, kad būtų galima kuo greičiau kelią nuvalyti. Mokinio užduotis yra nubraižyti valomus kelius per ežerą. Nėra jokių ribojimų, nes visas ežeras užšalęs ir padengtas sniegu.

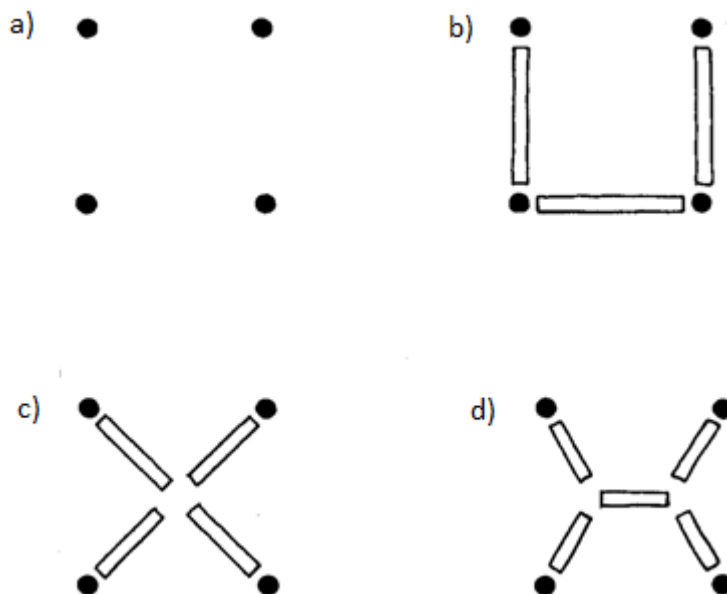
Akivaizdu, kad keliai turi būti visiškai tiesūs, nes lankai juos tik pailgina. Tačiau ne visada sujungti taškus tiesiomis linijomis yra geriausias sprendimas. Kartais, įterpus naują susikirtimo tašką, bendras kelių ilgis gali būti sutrumpintas.



Pavyzdžiui, a paveiksle pavaizduoti trys gręžiniai. Šiuos taškus galima sujungti tiesiomis linijomis, kaip parodyta b paveiksle. Kitas jungimo būdas pavaizduotas c paveiksle. Matuojant nesunku išsiaiškinti, kad c paveiksle bendras kelių ilgis yra trumpiausias. Pridėtas papildomas taškas, sutrumpinantis bendrą kelio ilgį, vadinamas Šteinerio tašku. Jakobas Šteineris (1796–1863) suformulavo šį uždavinį ir pirmasis pastebėjo, kad bendras kelio ilgis gali sutrumpėti pridėjus papildomą susikirtimo tašką.

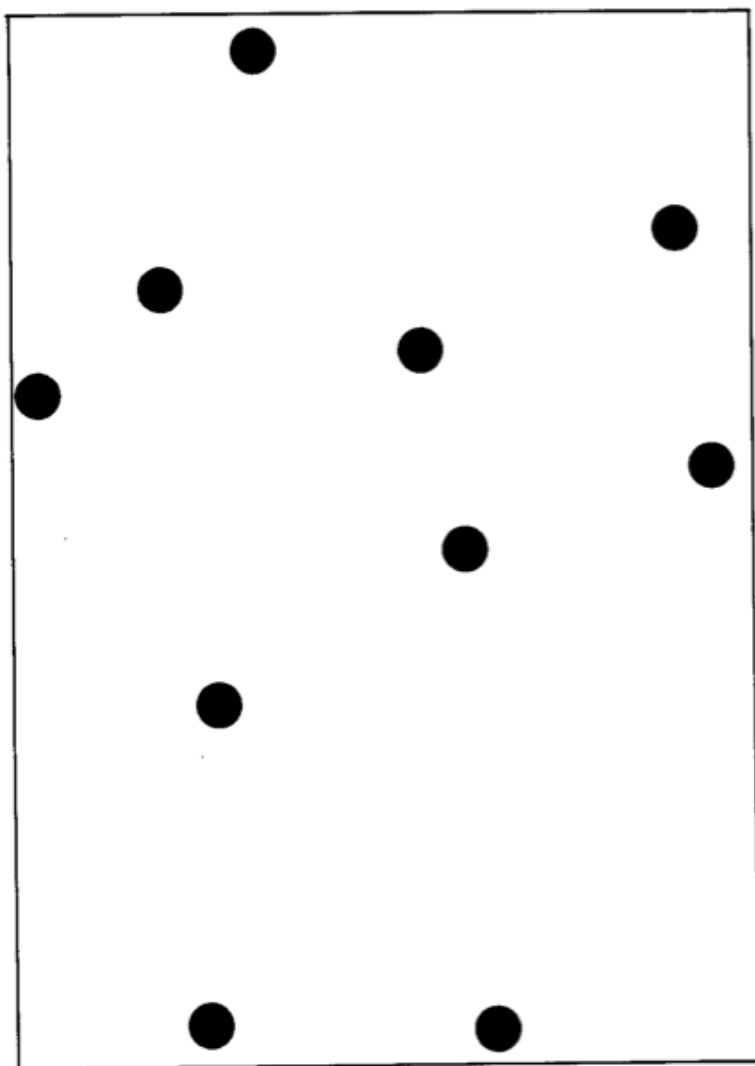
Diskusija ir veiklos

1. Užduotis aiškinama žemiau paveiksle pateiktu pavyzdžiu.

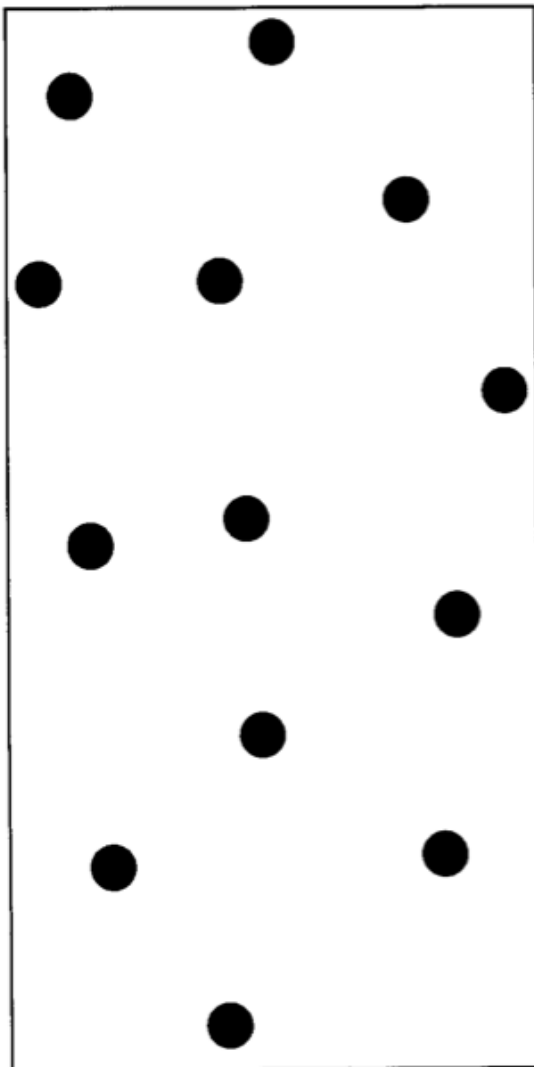


2. Užduotis išbandoma lauko aikštelėje. Mokiniai aikštelėje susmeigia keturis kuoliukus kvadratu, kurio kraštinės po 1 m. Iš pradžių jie nežymi Šteinerio taško, tik sujungia kuoliukus virve. Akivaizdu, kad reikia 4 m virvės. Ar galima būtų sujungti visus kuoliukus trumpesne virve?
3. Toliau mokiniams siūloma surasti Šteinerio taškus. Nustačius vieną Šteinerio tašką (įsmeigus dar vieną kuoliuką kvadrato viduryje), kaip parodyta c paveiksle, reikia mažiau virvės. Bendras virvės ilgis lygus $2\sqrt{2} = 2,83$ m. Suradus du Šteinerio taškus (įsmeigus du kuoliukus), kaip parodyta d paveiksle, sunaudojama $1 + \sqrt{3} = 2,73$ m virvės.
4. Ar virvės reikėtų mažiau, jei rastumėte 3 Šteinerio taškus? (Ne. Du taškai yra geriausias variantas.)
5. Diskutuojama su mokiniais, kodėl ši užduotis yra sudėtinga. (Nežinoma, kaip nustatyti Šteinerio taškus, todėl tenka išbandyti daug variantų.)

Darbo lapas Šteinerio medis



Darbo lapas *Šteinerio medis*



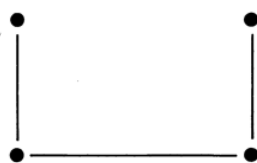
Gudručiams

1. Greitai pagrindines užduotis atlikę mokiniai gali bandyti rasti trumpiausią kelią stačiakampyje 1×2 m (a paveikslas). Sujungus b variantu reikia 4 metrų virvės. Suradus vieną Šteinerio tašką, reikia $2\sqrt{5} = 4,47$ m virvės (c paveikslas), kai turime du Šteinerio taškus, reikia $2 + \sqrt{3} = 3,73$ m virvės (d paveikslas). Mokiniai galėtų pagrįsti, kodėl pridėjus vieną papildomą tašką stačiakampyje kelias sutrumpėja labiau, nei pridėjus papildomą tašką kvadrate. (Taip yra todėl, kad kai kvadratas ištempiamas į stačiakampį, tai b ir d atvejais pailgėjimų suma padidėja po vieną kartą, o c atveju padidėja abi įstrižainės).

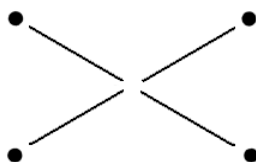
(a)



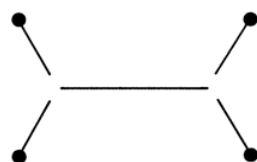
(b)



(c)

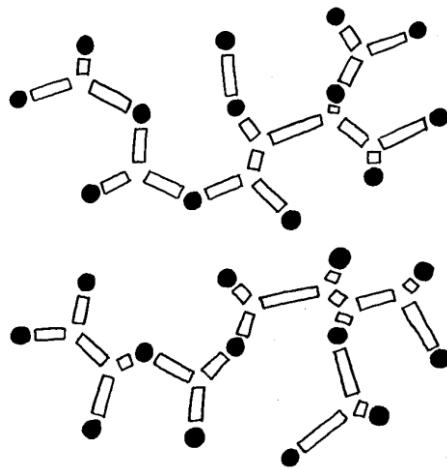


(d)



2. Vyresnieji mokiniai gali nagrinėti užduotį, kurioje yra daugiau taškų. Jie galėtų spręsti kitas darbo lapuose pateiktas užduotis naudodami pieštukus ir trintukus arba naujas darbo lapų kopijas. Užduoties sprendimą galima pavaizduoti ir aikštelėje, taškus sužymėti kuoliukais. Mokiniai galėtų diskutuoti tarpusavyje, kaip rasti geresnį sprendimą. (Pavyzdžių trumpiausio atstumo pavaizduoti paveiksluose dešinėje.) Pateikiami du panašūs Šteinerio medžiai rodo, kad užduotis yra sunkiai išsprendžiama. Čia galima pasirinkti gana daug Šteinerio taškų!

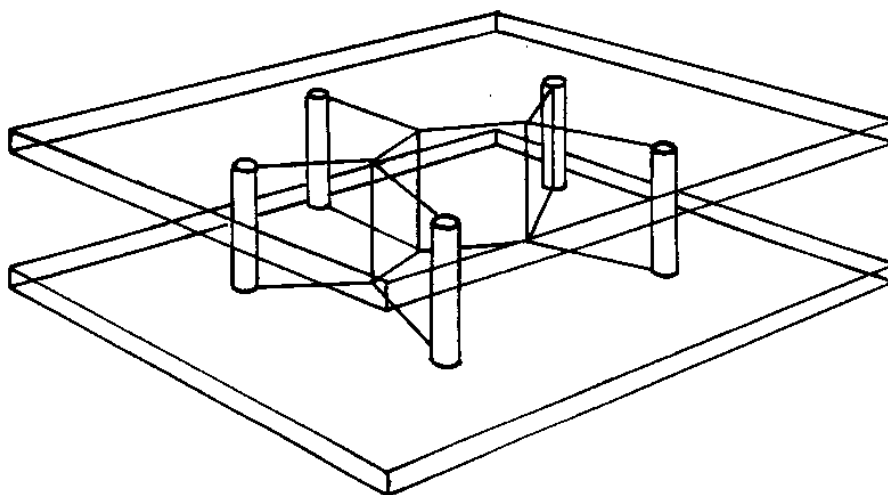
Du galimi Šteinerio medžiai antrajam darbo lapo pavyzdžiui.



3. Žemiau pateikiamas virvinių kopėčių užduoties pavyzdys, kurią galėtų spręsti mokiniai, baigę pagrindines užduotis:



4. Dviejų laiptelių virvinių kopėčių užduotis gali būti paverčiama kvadrato uždaviniu. Trijų laiptelių virvinių kopėčių užduoties sprendimas yra visiškai kitoks. Jį galima pabandyti nubraižyti. Keturių laiptelių virvinių kopėčių užduotį galima paversti dviejų sujungtų kvadratų uždaviniu. Penkių laiptelių virvinių kopėčių užduoties sprendimas yra išplėstas trijų laiptelių kopėčių uždavinys. Minimalusis Šteinerio medis virvinių kopėčių uždavinyje priklauso nuo taškų skaičiaus – ar tas skaičius yra lyginis, ar nelyginis. Jei taškų skaičius lyginis, užduotis sprendžiama taip, tarsi tai būtų sujungti keli kvadratai, o jei nelyginis – sprendžiama taip, tarsi būtų kartojama trijų laiptelių virvinių kopėčių uždavinys. Tačiau šio sprendimo įrodymas nėra lengvas.
5. Kita įdomi šių uždavinių interpretacija yra Šteinerio medžių muilo burbulai. Du permatomo plastiko lapai persmeigiami smeigėmis (ar virbalais), kaip parodyta žemiau paveiksle:



Įmerkus visą konstrukciją į muilo tirpalą ir po to ištraukus, muilo burbulų plėvelė, susidariusi tarp plastiko lapų, ima judėti, ieškodama minimalios energijos būsenos, bet gana greitai viskas nurimsta – gaunamas Šteinerio medžio tinklas!

Žinoma, tai nėra įrodymas, kad gaunamas minimalusis Šteinerio medis. Muilo burbulai rado mažiausią kiekvienos dalies kelią, bet nebūtinai mažiausią bendrą kelią. Nesunkiai galima pavaizduoti visiškai kitokį Šteinerio taškų išdėstymą šioje užduotyje.

Apie ką visa tai?

Šiame skyriuje nagrinėjami *minimalieji Šteinerio medžiai*. Jie vadinami *medžiais*, nes neturi ciklų, o tik šakas – kaip tikri medžiai; *Šteinerio medžiais*, nes į originalų medį gali būti įterpiama naujų Šteinerio taškų; *minimaliaisiais* medžiais, nes bendras tokio medžio briaunų ilgis yra mažiausias.

Įdomu tai, kad minimaliajam jungiančiajam medžiui yra sukurtas efektyvus algoritmas (14 veikla), bet minimalusis Šteinerio medis efektyvaus algoritmo neturi. Šteinerio medžių uždaviniai yra daug sunkesni, nes reikia nuspręsti, kur įterpti Šteinerio tašką, be to, tokių įterpimo galimybių yra labai daug. Tačiau nėra sunkiausia tiksliai įterpti Šteinerio tašką, kur kas sunkiau yra nuspręsti, kur apytikriai jį įterpti. Nusprendus, kur apytikriai įterpti tašką, surasti jo optimalią vietą jau ne taip sunku. Muilo burbulais tai padaroma labai efektyviai. Tą gali ir kompiuteris.

Minimalusis Šteinerio medis labai praverė taupant lėšas telefonų versle. Dar iki 1967 m. JAV vartotojų korporacija turėjo didelį privačių telefonų tinklą, kurio linijos buvo nuomosios iš telefonų kompanijos. Mokestis buvo skaičiuojamas ne pagal faktinį reikalingą laidų ilgį, bet pagal naudojamą trumpiausią tinklą. Vartotojas neprivalėjo mokėti daugiau, jei telefonų kompanija nutiesė ilgesnę liniją, nei pakaktų iki vartotojo telefono. Algoritmas, kuriuo buvo skaičiuojama, kiek reikia mokėti, nustatydavo minimalųjį jungiantįjį medį. Tačiau apie 1967 m. vartotojai pastebėjo, kad įterpus papildomą telefono linijos stotelę pakankamas atstumas iki vartotojo telefono sumažėja. Telefono kompanija buvo priversta sumažinti įkainius. Paprastai minimalusis Šteinerio medis yra tik 5–10 % trumpesnis nei minimalusis jungiantysis medis, bet mokant dideles sumas šis pokytis tampa reikšmingas. Šteinerio medžio uždavinys dar vadinamas „trumpiausiojo tinklo uždaviniu“, nes randama trumpiausią tinklą jungiančių taškų aibė.

Šteinerio medis, kaip ir anksčiau aptarti uždaviniai, yra NP pilnumo uždavinys. Didėjant taškų skaičiui, Šteinerio taškų įterpimo galimybių skaičius didėja eksponentiškai. Tai dar vienas iš anksčiau minėtų tūkstančio NP pilnumo uždavinių, kuris neturi efektyvaus sprendimo, ir nėra žinoma, ar polinominio laiko algoritmas egzistuoja šiam uždaviniui spręsti. Žinoma tik tai, kad, jei būtų surastas polinominio laiko algoritmas šiam uždaviniui spręsti, tai jį būtų galima pritaikyti ir kitiems NP pilnumo uždaviniams spręsti.

Ankstesniame skyriuje buvo paaiškinta, kas yra NP pilnumas. Uždaviniai, kuriems gali būti taikomas polinominio laiko algoritmas, žymimi „P“. Pagrindinis klausimas: ar $P = NP$? Atsakymas nėra žinomas ir tai didžiausia informatikos paslaptis!

Uždaviniai, sprendžiami taikant polinominio laiko algoritmus, vadinami „lengvai sprendžiamais“, nors kai kurie algoritmai yra gana lėti.

Ką galima pasakyti viršininkui, kuris liepė sugalvoti efektyvų algoritmą uždaviniui optimaliai išspręsti, o to padaryti negalima? Pavyzdžiui, kai oro linijos susiduria su faktu, kad skrydžių kaina gali būti sumažinta į skrydžių linijų tinklą įterpiančiais Šteinerio taškais. Būtų puiku įrodyti, kad geresnio sprendimo nėra, bet įrodyti neigiamus rezultatus labai sunku, ypač informatikoje. Sumanus programuotojas gali gudrauti.

Tačiau negalima tvirtai pasakyti viršininkui, kad jo paliepimas neįvykdomas ir uždavinys sunkiai sprendžiamas. Vienaip ar kitaip, jei galima parodyti, kad tai yra NP pilnumo uždavinys, reiškia, kad jis yra ekvivalentus vienam iš tūkstančių uždavinių, kuriuos bando išspręsti daugelis informatikų įvairiose šalyse. Tai neišspręstų jums pateikto uždavinio, tačiau bent turėtumėte rimtą pasiteisinimą!



„Negaliu rasti efektyvaus algoritmo. Tikriausiai aš per kvailas.“

„Negaliu rasti efektyvaus algoritmo, nes, manau, toks algoritmas neįmanomas.“

„Negaliu rasti efektyvaus algoritmo, bet to padaryti negali ir šie protingi žmonės.“

Trys galimybės, ką galima pasakyti viršininkui, kai nerandama efektyvaus algoritmo.

Aišku, realiame gyvenime šie uždaviniai kyla ir turi būti sprendžiami, jiems taikomi *euristiniai* algoritmai, kurie neužtikrina geriausio rezultato, bet randa artimą rezultatą. Euristiniai algoritmai gali būti labai greiti. Jie labai praverčia realiame gyvenime, nes ir neradus geresnio sprendimo nuostolis būna minimalus. Tačiau žinojimas, kad gali būti sudarytas geresnis pamokų tvarkaraštis, geresnis koks nors tinklas arba kelių sistema, tiesiog truputį erzina.

Daugiau informacijos

Animacinis filmas pagal M. Garey ir D. S. Johnsono vadovėlį „*Computers and Intractability*“.

Žurnalo „*Scientific American*“ skyrelyje „Computer recreations“ 1984 m. liepos mėn. išspausdintas išsamus aprašymas, kaip konstruoti Šteinerio medžius naudojant muilo burbulus.



V dalis

Informacijos šifravimas

Kriptografija

Informacijos šifravimas

Žodis kriptografija“ sudarytas iš dviejų graikiškų žodžių: „kripto“ graikiškai reiškia „slėpti“, „graphin“–„raštas“. Taigi kriptografija –mokslas apie slaptaraščius. Šis mokslas visada rūpėjo diplomatams, kariškiams, tačiau daugumai žmonių iki informacinių technologijų revoliucijos jis buvo visai nereikalingas. Kai atsirado kompiuterių tinklai, kai pasaulį apraizgė saitynas, kai internetu kasdien siunčiama ir gaunama svarbios informacijos, kriptografija tapo reikalinga visiems. Ji nebėra vien mokslas apie šifrų kūrimą ir jų saugumą. Per kelis pastaruosius dešimtmečius kriptografija virto matematikos ir informatikos metodų taikymo informacijos apsaugai sritimi. Apsaugoti informaciją mūsų laikais – tai ne vien tik paslėpti jos prasmę nuo tų, kuriems nederėtų jos matyti. Tai taip pat ir apsauga nuo neteisėto informacijos keitimo, klastojimo, jos naudojimo nenumatytiems tikslams.

Kartais manoma, kad kriptografija – tai slaptažodžių saugojimas kompiuteriuose ir pranešimų šifravimas, kad priešiški asmenys negalėtų jų perskaityti, tačiau iš tikrųjų taip nėra. Šiuolaikinės kompiuterių sistemos nesaugo slaptažodžių, nes, jei saugotų, bet kuris prieigą prie slaptažodžių turintis asmuo galėtų įsibrauti į visas sistemas: galėtų daryti bankų pavedimus, siųsti pranešimus apsimesdamas kitu asmeniu, skaityti slaptus failus, vadovauti kariuomenei ar net nuversti vyriausybę. Dabar slaptažodžiai yra tvarkomi „vienos krypties funkcijomis“, kurios buvo pristatytos 15 veikloje. Kriptografija taip pat nėra tik raidžių kodavimas pranešimų tekstuose: pranešimai šifruojami metodais, siejamais su itin sudėtingomis problemomis – sunkiai sprendžiamais uždaviniais, pavyzdžiui, keletas tokių sunkiai sprendžiamų uždavinių aprašyta IV dalyje.

Pasitelkiant kriptografiją galima daryti, atrodytų, neįmanomus dalykus. Šioje dalyje sužinosite, kaip galima nesunkiai apskaičiuoti grupės žmonių vidutinį amžių, kai nė vienas grupės narys nenurodo savo amžiaus; kaip du vienas kitu nepasitikintys žmonės gali mesti monetą ir susitarti, kuria puse ji nukrinta, net jei jie yra skirtinguose miestuose ir negali abu vienu metu matyti tos metamos monetos; kaip saugiai užkoduoti slaptus pranešimus, kuriuos galėtų iškoduoti tik vienas žmogus, nors ir visi žinotų, kaip jie užkoduoti.

Mokytojui

Aprašant šios dalies veiklas remiamasi šiuolaikinės kriptografijos metodais, kurie iš esmės labai skiriasi nuo įprasto žmonių supratimo apie slaptumą ir kompiuterius.

Čia norima supažindinti su dviem pagrindinėmis idėjomis. Pirmoji – tai protokolas–formalus transakcijos patvirtinimas. Protokolai reikalingi ne tik diplomatiniams santykiams ar etiketo taisyklėms nusakyti– kompiuteriams jie taip pat reikalingi. Tariamai sunki užduotis gali būti išsprendžiama taikant stulbinančiai paprastą protokolą. 17 veikla parodoma, kaip grupė bendradarbiaujančių žmonių nesunkiai apskaičiuoja savo grupės amžiaus (arba pajamų) vidurkį nežinodami nė vieno grupės nario amžiaus (ar pajamų). Antroji idėja –skaičiavimų sudėtingumo priklausomybė nuo individų interaktyvios veiklos kompiuteriu. Iš 17 veiklos matyti,

kaip du nebūtinai vienas kitu pasitikintys žmonės, bendraudami tik telefonu, gali susitarti, kuria puse krinta išmesta moneta. (Ši veikla supažindina su Bulio logika ir jos taikymu.) 18 veikla supažindina, kaip žmogus skaičiavimo metodais gali saugiai užkoduoti pranešimus, net jei užkodavimo metodas yra visiems žinomas.

Atlikti kai kurias iš šių veiklų yra sunku, ypač paskutinę. Mokytojui reikia motyvuoti visą klasę, įtraukti mokinius į veiklas ir juos nustebinti, kad tokie dalykai, kurie daugumai atrodo neįmanomi, gali būti padaryti. Esminis dalykas – sukurti mokiniams netikėtumo atmosferą, aptariant reikiamą atlikti veiklą išlaikyti mokinių susidomėjimą, kad jie matytų ne vien smulkmenas, pavienius sunkumus, bet sugebėtų aprėpti nuostabių galimybių teikiančią visumą. Šios dalies veiklos yra techniškai sudėtingos, kartais tai tikri iššūkiai. Jei jos per sunkios, galima šią dalį praleisti ir pereiti prie VI dalies, kuri yra visiškai kitokia, nei kitos knygos dalys, ir neturi techninių aspektų.

Tiksliai mąstantiems

Kompiuteriai įsiveržia į kasdienį gyvenimą, o kriptografijos taikymas tampa kasdiene tendencija. Dauguma žmonių nežino, ką gali šiuolaikinės kriptografijos protokolai. Didelės organizacijos (vyriausybines ir komercines) naudoja įvairias informacines sistemas, kurios apima ir privačią informaciją, o tam tikri technokratai sprendžia, kaip ta informacija turi būti tvarkoma, kas turi būti renkama, kas viešai prieinama ir t. t. Jei žmonės daugiau žinotų apie šiuolaikinių technologijų suteikiamas galimybes, galėtų aktyviau įsitraukti ir dalyvauti priimant šiuos sprendimus.

Ši medžiaga apie informaciją slepiančius protokolus, kriptografijos protokolus ir viešojo rakto kriptografiją yra gana sudėtinga. Tačiau pačios idėjos gana paprastos, nors techninius jų aspektus be gilios konceptų analizės ir sunku suvokti. Praktinėse situacijose, pavyzdžiui, užsiimant elektronine prekyba, techninės detalės paslepamos kompiuterio programose, todėl galima lengvai taikyti naujas šifravimo technologijas. Tačiau vis tiek svarbu suprasti idėjas, kuriomis šios šifravimo technologijos grindžiamos, kad įgytume gilesnės patirties kriptografijoje.

Valstybės domisi kriptografijos sistemomis ne tik dėl to, kad nori išlaikyti savo informacijos ir keitimosi ja slaptumą, bet ir dėl to, kad savo informaciją dažnai užšifruoja ir nelegalia veikla užsiimantys asmenys (narkotikų prekeiviai, teroristai). Jei, pavyzdžiui, jie užšifruoja savo telefoninius pokalbius, tų pokalbių pasiklausyti įmanoma tik žinant iššifravimo metodus. Šių metodų išaiškinimo ar panaši teisėsaugos institucijų veikla kelia daug įvairių žmonių diskusijų: vieniems atrodo, kad reikia plėsti kriptografijos sistemų galimybes, o kiti jaučiasi suvaržyti, kai vyriausybės turi galimybių stebėti jų privačią komunikaciją. JAV vyriausybė apribojo kai kurių kriptografijos metodų naudojimą ir prilygino juos karo ginkluotei, kuri tampa pavojinga, kai pakliūva į blogas rankas. Daug diskusijų sukėlė JAV nacionalinės saugumo agentūros sukurta lustų sistema „Clipper Chip“, kurią telekomunikacijos kompanijos naudojo balsui perduoti. Šios sistemos pagrindas buvo raktų deponavimas (angl. *keyescrow*). Kiekvienam naujam telefonui ar kitam prietaisui šia lustų sistema galėjo būti suteikiamas kriptografijos raktas, kurį būtų galėjusi turėti ir



vyriausybė. Jei kuriai nors valstybės agentūrai reikėtų pasiklausyti pokalbių tokiu telefonu, ji būtų galėjusi naudotis šiuo raktu. JAV teisingumo organai norėjo, kad šis lustas būtų naudojamas visuose komunikacijų prietaisuose, tačiau tai sulaukė piliečių pasipriešinimo dėl grėsmės privatumui. Taigi visos kriptografijos sistemos yra techniškai įmanomos, bet nebūtinai politiškai priimtinos.

Kriptografija taikoma daug plačiau, ne tik pranešimams šifruoti. Ji pasitelkiama nustatant asmens tapatybę, pranešimo siuntėją. Negalint nustatyti tapatybės neįmanoma elektroninė komercija. Taip pat neįmanomas ir elektroninis balsavimas: žmonės gali balsuoti elektroniškai nebijodami, kad kas nors (net ir paleidęs balsavimo sistemą) sužinos, už ką jie balsavo, tačiau vis dar neleidžiama balsuoti daugiau nei vieną kartą. Dar vienas pavyzdys – kortų žaidimas telefonu. Šis pavyzdys atrodo gana nereikšmingai, tačiau tik tol, kol nepagalvojama, kad verslo sandoriai yra labai panašūs į pokerio žaidimą.

Dažnai šie dalykai atrodo neįmanomi. Kaip galima sumaišyti kortų malką telefonu, jei žaidžiama su nepažįstamu žmogumi, kuriuo galbūt negalima pasitikėti? Kaip galima nustatyti, ar gautas laiškas yra originalus, o galbūt kas nors jį paėmė, modifikavo, užšifravo ir išsiuntė? Jei visų šių dalykų nebūtų galima nustatyti ar išsiaiškinti, tai nebūtų galima tvarkyti verslo elektroninėje erdvėje. Reikia sutrukdyti nusikaltėliams klastoti prieigas prie banko sąskaitų perimant telefono liniją tarp pardavimo vietos terminalo ir banko. Reikia sutrukdyti verslo konkurentams kenkti kuriant neteisingus pranešimus ar sutartis. Šiuolaikinės kriptografijos metodai gali išspręsti aprašytąsias problemas. Šios dalies veiklos parodo, kaip tai galima padaryti.

Galima surasti keletą knygų apie šifravimą ir dešifravimą (šifrų nulaužimą):

Francis Harry Hinsley ir Alan Stripp (eds) „Code breakers: the inside story of Bletchley Park“ (Oxford University Press, 1993) – ši knyga pasakoja, kaip pirmaisiais kompiuteriais Antrojo pasaulinio karo metu buvo iššifruojama informacija.

Įdomi matematiko Viliaus Stakėno knyga „Šifrų istorijos“ (Vilnius, TEV, 2005).



17 veikla

Dalijimasis paslaptimis. Informacijos slėpimo protokolai

Santrauka

Kriptografijos metodai leidžia dalytis informacija su kitais žmonėmis ir kartu išlaikyti itin aukštą privatumo lygį. Šiame skyriuje aptariama, kaip galima dalytis neatskleista informacija: mokinių grupė skaičiuoja savo grupės amžiaus vidurkį, nors nežino tikslaus grupės narių amžiaus.

Ryšiai su ugdymo programomis

- ✓ Matematika: sudėtis ir vidurkis

Gebėjimai

- ✓ Vidurkio skaičiavimo
- ✓ Bendradarbiavimo

Amžius

- ✓ Nuo 7 metų

Priemonės

Kiekvienai mokinių grupei reikės:

- ✓ Užrašų knygelės ar sąsiuvinio
- ✓ Rašiklio



Dalijimasis paslaptimis



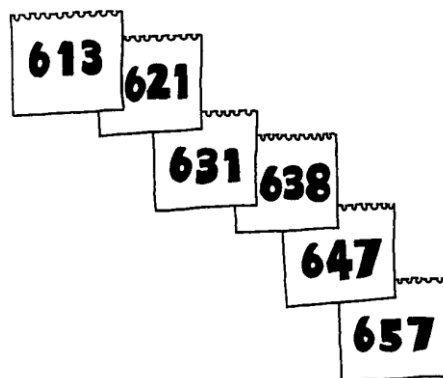
Įvadas

Atlikdami šią veiklą mokiniai skaičiuoja savo grupės amžiaus vidurkį, nors nežino tikslaus kiekvieno grupės mokinio amžiaus. Lygiagrečiai kitos grupės gali skaičiuoti grupės gaunamų dienpinigių vidurkį. Šių statistinių duomenų skaičiavimas ypač tinkamas suaugusiesiems – juk jiems labiau rūpi informacija apie amžių ir pajamas.

Grupėse turi būti bent po 3 mokinius.

Diskusija

1. Paaiškinama, kad grupių mokiniai vieni kitiems negali pasakyti savo tikslaus amžiaus (ar dienpinigių sumos). Jie turi suskaičiuoti amžiaus (dienpinigių) vidurkį nežinodami vienas kito amžiaus (dienpinigių sumos). Mokinių paklausama, ar jie tiki, kad tai įmanoma padaryti, ir kaip tai būtų galima padaryti.
2. Pasirenkama 6–10 mokinių. Pirmajam mokiniui duodama užrašų knygelė ir rašiklis ir paprašoma užrašyti bet kokį triženklį skaičių. Pavyzdyje dešinėje atsitiktinai parinktas skaičius 613.
3. Pirmasis mokinsys išplėšia lapą su užrašytu atsitiktiniu skaičiumi. Jis prideda savo amžių prie pirmojo skaičiaus ir užrašo gautą rezultatą antrajame lape. Pavyzdyje pirmojo mokinio amžius yra 8, todėl antrasis triženklis skaičius yra 621. Išplėštas lapas nerodomas kitiems mokiniams.
4. Užrašų knygelė perduodama antrajam mokiniui. Jis išplėšia lapą, prideda savo amžių prie užrašyto lape skaičiaus ir užrašo gautą rezultatą trečiajame lape. Pavyzdyje antrojo mokinio amžius yra 10 metų, todėl užrašoma 631.
5. Tai tęsiama, kol visi mokiniai užrašo skaičius.



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

6. Užrašų knygelė grąžinama pirmajam mokiniui. Mokinys atima savo sugalvotą skaičių iš užrašyto paskutinio skaičiaus. Pavyzdyje užrašų knygelė perėjo per penkių mokinių rankas, paskutinis užrašytas skaičius – 657. Iš 657 atimamas sugalvotas skaičius 613. Gaunama 44. Tai visų mokinių amžiaus suma. Šią sumą padalijus iš 5 gaunamas grupės mokinių amžiaus vidurkis – 8,8.
7. Pabrėžiama, kad nė vieno mokinio amžius nebus žinomas, kol nebus parodyti išplėstieji lapai su skaičiais.

Gudručiams

Ši sistema gali būti taikoma slapta balsuojant, kai kiekvienas balsuojantysis prideda vienetą, jei balsuoja „taip“, arba 0, jei balsuoja „ne“. Žinoma, jei kas nors prideda daugiau kaip vieną (arba mažiau kaip 0), balsavimas yra neteisingas. Taip pat rizikuojama sukelti įtarimų, jei visi balsuoja „taip“, nes balsų „taip“ skaičius būna didesnis užbalsavusių žmonių skaičių.



Apie ką visa tai?

Kompiuteriuose saugoma labai daug mūsų asmeninės informacijos: banko sąskaitų likučiai, socialinių tinklų informacija, kiek mokame mokesčių, kiek galioja turimas vairuotojo pažymėjimas, kokie mūsų egzaminų rezultatai, medicininiai įrašai ir pan. Šios informacijos privatumas labai svarbus, tačiau kai kuria informacija mums reikia dalytis su kitais. Pavyzdžiui, parduotuvėje mokant už prekes banko kortele, pardavėjui reikia įsitikinti, kad pirkėjas turi lėšų savo banko sąskaitoje.

Dažnai pateikiama daugiau informacijos, negu iš tikrųjų reikia. Pavyzdžiui, kai atliekamas elektroninis mokėjimas, pardavėjas gauna pirkėjo banko pavadinimą, sąskaitos numerį, vardą ir pavardę. Be to, bankui perduodama, kur asmuo perka. Iš banko gaunamų privačių kliento duomenų būtų galima sukurti to kliento profilį: kur perkami degalai, kur bakalėjos prekės, kiek ir kur išleidžiama per dieną. Mokant grynaisiais pinigais šios informacijos nebūtų galima sužinoti. Dauguma žmonių nekreipia dėmesio į savo asmens duomenų dalijimą, tačiau visada yra tikimybė, kad privačiais duomenimis gali būti piktnaudžiaujama: jie gali būti naudojami tikslinei rinkodarai (pavyzdžiui, siunčiama kelionių reklama žmonėms, daug išleidžiantiems kelionių bilietams), jais disponuojant galima diskriminacija (pavyzdžiui, geresnių paslaugų siūlymas pasiturintiems bankų klientams) ar šantažas (pavyzdžiui, grasinimai pavišinti abejotiną sandorį). Visada galima pakeisti pirkimo būdą, jei manoma, kad perkančiojo privačiais duomenimis kas nors neteisėtai pasinaudos.

Nors nurodyti privatumo pažeidimai priimtini daugumai žmonių, jau esama kriptografijos protokolų, kurie leidžia elektronines finansines operacijas atlikti tokiu pačiu privatumo lygiu, kaip mokant grynaisiais. Sunku patikėti, kad pinigai gali būti pervedami iš jūsų banko sąskaitos į parduotuvės sąskaitą niekam nežinant, iš kurios sąskaitos ir į kurią jie pervedami. Ši veikla leidžia įsitikinti, kad tokia operacija įmanoma: abiejose situacijose ribojamas informacijos dalijimasis, o tai įmanoma pasiekti „protingu“ protokolu.

Daugiau informacijos

Davidas Chaumo straipsnyje „Security without identification: transaction systems to make Big Brother obsolete“ pateikiama paprastų informacijos slėpimo protokolų pavyzdžių, kaip gali būti atliekamos visiškai privačios elektroninės finansinės operacijos. Straipsnis išspausdintas 1985 m. žurnale „Communication of the ACM“.



18 veikla

Monetos metimas. Kriptografijos protokolai

Santrauka

Šia veikla sprendžiama paprasta, bet iš pirmo žvilgsnio atrodanti neįmanoma užduotis. Du nepažįstami ir nebūtinai vienas kitu pasitikintys žmonės, bendraujantys tik telefonu, meta monetą ir priima nešališką sprendimą, kas atsivers: herbas ar skaičius.

Ryšiai su ugdymo programomis

- ✓ Matematika: loginis pagrindimas
- ✓ Matematika: Bulio algebra

Gebėjimai

- ✓ Bulio algebros taikymo
- ✓ Funkcijų taikymo
- ✓ Dėlionių sprendimo

Amžius

- ✓ Nuo 9 metų

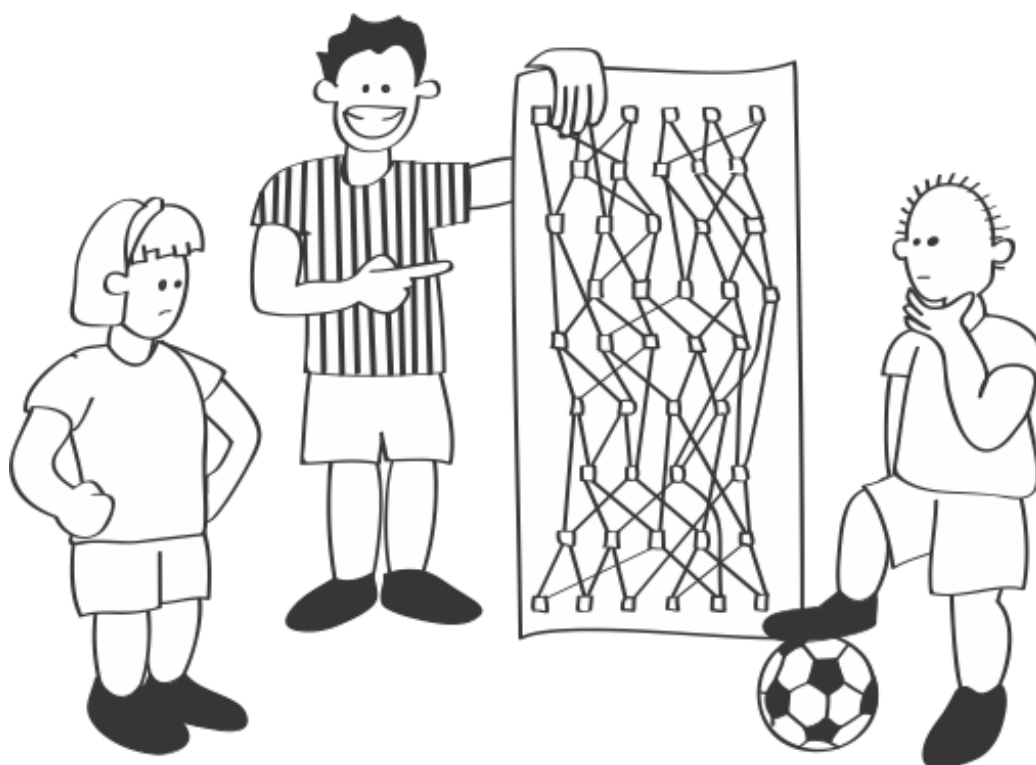
Priemonės

Kiekvienai grupei reikės:

- ✓ Darbo lapo „Monetos metimas“
- ✓ Po pakelį dviejų skirtingų spalvų sagų ar figūrėlių



Peru monetos metimas



Ivadas

Šią veiklą sukūrė vienas Peru mokytojas, dirbdamas su mokiniais. Veiklos istoriją galima pritaikyti savo šaliai.

Limos ir Kusko futbolo komandos turi išsiaiškinti, kuri taps šalies čempionė, tačiau niekaip negali susitarti, kur turėtų vykti rungtynės – Limoje ar Kuske. Paprasčiausias būdas tai nuspręsti – mesti monetą, bet abu miestus skiria didelis atstumas. Alisa, atstovaujanti Limai, ir Benitas, atstovaujantis Kuskui, negali skirti tiek pinigų ir laiko, kad būdami kartu vienoje vietoje galėtų mesti monetą. Ar jie gali tai padaryti telefonu? Alisa galėtų mesti monetą, o Benitas galėtų pasirinkti herbą arba skaičių, bet tai gali būti nesąžininga. Jei Benitas pasakytų „herbas“, tai Alisa galėtų pasakyti, kad iškrito skaičius, ir Benitas niekada nelaimėtų. Alisa paprastai neapgaukinėja, bet tai svarbios rungtynės, todėl pagunda labai didelė. Net jei Alisa ir pasakytų tiesą, ar Benitas ja patikėtų?

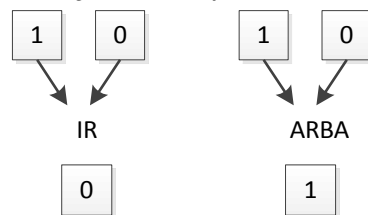
Pastaba. Mokiniai geriau atliktų šią veiklą, jei būtų susipažinę su dvejetainiais skaičiais (1 ir 4 veikla) ir būtų matę 15 veiklos vienos krypties funkcijos pavyzdį.

Jie nusprendė sudaryti schemą, kuri turėtų du loginius elementus – IR ir ARBA. Tai padaryti jie gali telefonu, nors būtų gana nuobodu (galima naudotis ir el. paštu). Kuriantieji schemą nori būti tikri, kad ji bus tokia sudėtinga, kad nė vienas negalės sukčiauti. Galutinė schema yra abiem žinoma.

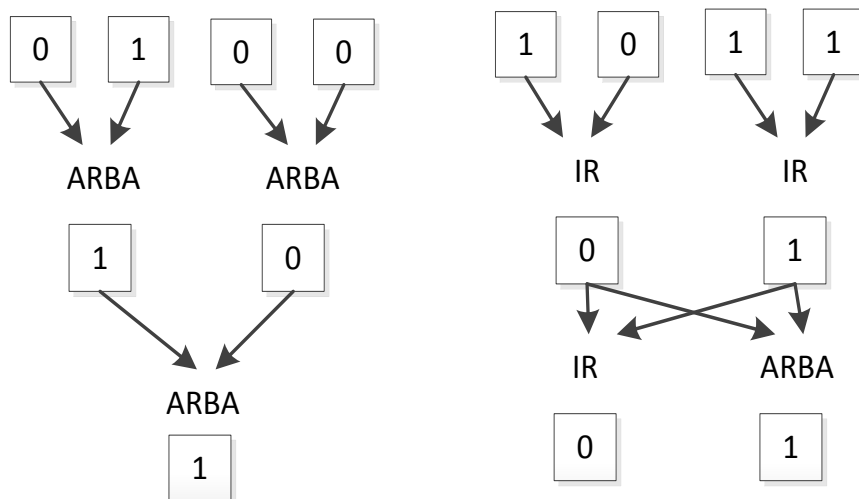


Diskusijos

Loginių elementų IR ir ARBA taisyklės yra labai paprastos. Gali būti du įvedimai 0 arba 1, kurie atitinkamai reiškia „netiesa“ arba „tiesa“. Jei abu įvedimai yra 1 („tiesa“), loginio elemento IR išvedimas yra 1 („tiesa“), visais kitais atvejais – 0 („netiesa“). Pavyzdžiui, paveiksle dešinėje elementas IR turi du įvedimus 1 ir 0, joišvedimas lygus 0. Loginio elemento ARBA išvedimas yra 0 („netiesa“), jei abu įvedimai yra 0, arba 1 („tiesa“) visais kitais atvejais. Pavyzdyje kairėje elemento ARBA išvedimas yra lygus 1, kai įvedimai yra 1 ir 0.

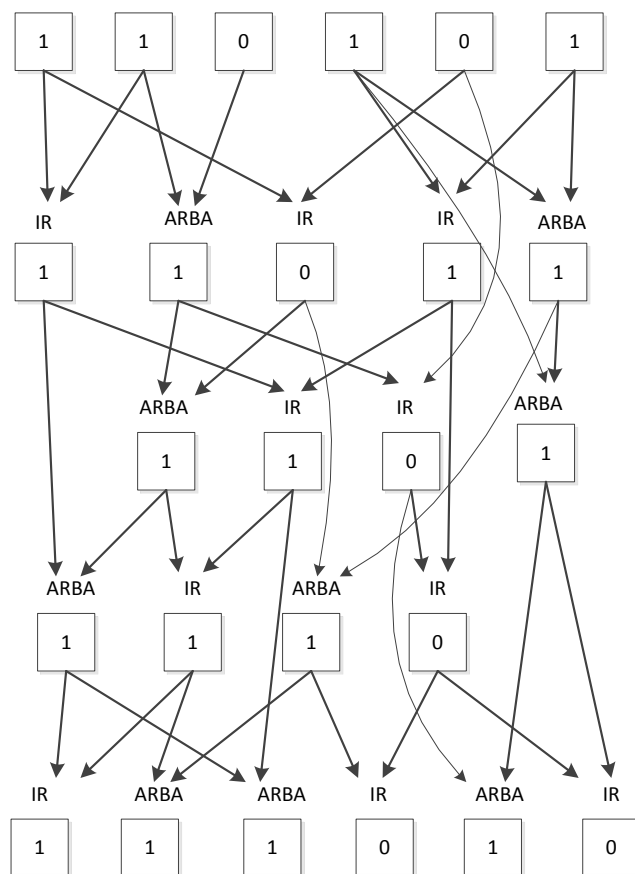


Vieno elemento išvedimas gali būti kito elemento (arba kelių elementų) įvedimas. Žemiau esančiame pavyzdyje kairiosios schemos dviejų elementų ARBA išvedimas yra trečiojo elemento ARBA įvedimas. Tokios schemos rezultatas yra 1, jei bent vienas iš keturių pradinių įvedimų yra 1. Dešiniojoje schemoje aukštesniųjų elementų IR išvedimai yra žemesniųjų schemos elementų IR ir ARBA įvedimai. Ši schema turi du išvedimus.



Peru problemos monetai mesti reikia netgi sudėtingesnės schemos. Žemiau pavaizduota schema turi 6 įvedimus ir 6 išvedimus. Pateikiamas galimas schemos įvedimų ir išvedimų pavyzdys:





Pagal šią schemą gali būti metama moneta telefonu toliau aprašytu būdu. Alisa pasirenka šešis pradinius įvedimus (nulis arba vienetus), kurių nesako Benitui. Sekdama schemos rodyklėmis atlieka numatytus veiksmus ir gautus šešis išvedimus nusiunčia Benitui. Gavęs Alisos siųstus išvedimus Benitas turi atspėti, ar buvo įvestas lyginis, ar nelyginis skaičius vienetų, t.y. jis turi atspėti Alisos įvedimų lyginumą. Jei schema yra pakankamai sudėtinga, Benitas nesiaiškina atsakymo, o tiesiog spėja (rinkdamasis atsakymą jis gali net mesti monetą!). Benitas laimi, taigi papildomos rungtynės vyksta Kusko mieste, jei jo spėjimas yra teisingas. Alisa laimi, vadinasi, rungtynės vyksta Limoje, jei Benito spėjimas yra neteisingas. Kai Benitas pasako Alisai savo spėjimą, Alisa parodo savo įvedimus, kad Benitas galėtų įsitikinti, jog viskas buvo teisinga.

1. Mokiniai suskirstomi į mažas grupes. Kiekvienai grupei duodamas Darbo lapas su nubraižyta schema, keletas žaidimo figūrėlių ir paaiškinama užduotis. Mokiniams bus įdomiau, jei įsivaizduos, kad reikia suorganizuoti savo ir kitos mokyklos komandų rungtynes. Susitariama dėl figūrėlių spalvų, pavyzdžiui, raudona reiškia 0, o mėlyna – 1 ar panašiai. Mokiniai pasižymi spalvas darbo lapo viršuje esančioje legendoje.
2. Mokiniams parodoma, kaip išdėlioti figūrėles schemos įvedimų kvadratėliuose. Tada paaiškinami elementų IR ir ARBA išvedimai, surašyti darbo lapo apačioje. Mokiniams galima pasiūlyti nuspaldinti schemų kvadratėlius sutartinėmis spalvomis.



3. Parodoma, kaip sekant schemos rodyklėmis ir atliekant nurodytus veiksmus dëliojamos figūrėlės ant kvadratelių ir šitaip apskaičiuojami išvedimų rezultatai. Viskas turi būti atlikta kruopščiai. Žemiau lentelėje (mokiniam ji nerodoma) pateikiami visi įmanomi įvedimai ir jų galimi išvedimai, jei kam nors kiltų abejonių.

Įvedimas	000000	000001	000010	000011	000100	000101	000110	000111
Išvedimas	000000	010010	000000	010010	010010	010010	010010	010010
Įvedimas	001000	001001	001010	001011	001100	001101	001110	001111
Išvedimas	001010	011010	001010	011010	011010	011010	011010	011111
Įvedimas	010000	010001	010010	010011	010100	010101	010110	010111
Išvedimas	001000	011010	001010	011010	011010	011010	011010	011111
Įvedimas	011000	011001	011010	011011	011100	011101	011110	011111
Išvedimas	001010	011010	001010	011010	011010	011010	011010	011111
Įvedimas	100000	100001	100010	100011	100100	100101	100110	100111
Išvedimas	000000	010010	011000	011010	010010	010010	011010	011010
Įvedimas	101000	101001	101010	101011	101100	101101	101110	101111
Išvedimas	001010	011010	011010	011010	011010	011010	011010	011111
Įvedimas	110000	110001	110010	110011	110100	110101	110110	110111
Išvedimas	001000	011010	011010	011010	011010	111010	011010	111111
Įvedimas	111000	111001	111010	111011	111100	111101	111110	111111
Išvedimas	001010	011010	011010	011010	011010	111010	011010	111111

4. Kiekviena grupė turi išsirinkti savo „Alisą“ ir „Benitą“. Grupė gali pasidalyti pusiau – vieni atstovauja Alisai, o kiti Benitui. Alisa pasirenka įvedimus ir apskaičiuoja išvedimus. Juos pasako Benitui, o šis bando atspėti įvedimo lyginumą (lyginis ar nelyginis vienetų skaičius buvo įvestas). Aišku, Benito atsakymas yra paprastas spėjimas. Alisa visiems pasako, kokie buvo įvedimai. Benitas laimi, jei atspėja lyginumą. Jis gali patikrinti pagal schemą, ar Alisa nepakeitė savo pasirinkimo.

Šiame etape monetos metimas baigiamas.

Benitas galėtų sukčiauti, jei surastų įvedimus, kurie pateikia Alisos rezultatą. Todėl Alisai svarbu, kad grandinės funkcija būtų vienos krypties, kaip aptarta 14 veiklos skyriuje. Vienos krypties funkcijos rezultatą labai lengva apskaičiuoti, kai žinomi pradiniai įvedimai, tačiau labai sunku nuspėti įvedimus, kai žinomas rezultatas.



Alisa galėtų sukčiauti, jei surastų du skirtingus išvedimus, kurie pateikia vienodą rezultatą. Kad ir ką tada Benitas spėtų, Alisa galėtų pateikti kitą atsakymą. Todėl Benitas turi įsitikinti, kad schema neatvaizduoja skirtingų įvedimų vienam rezultatui.

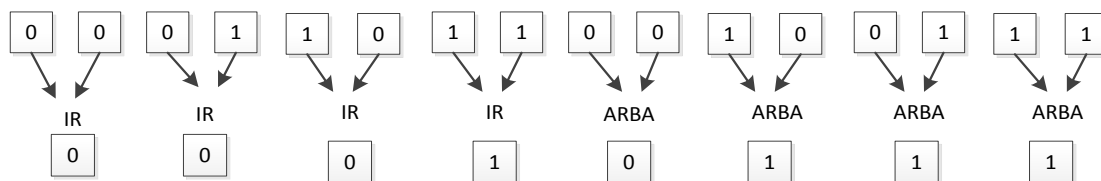
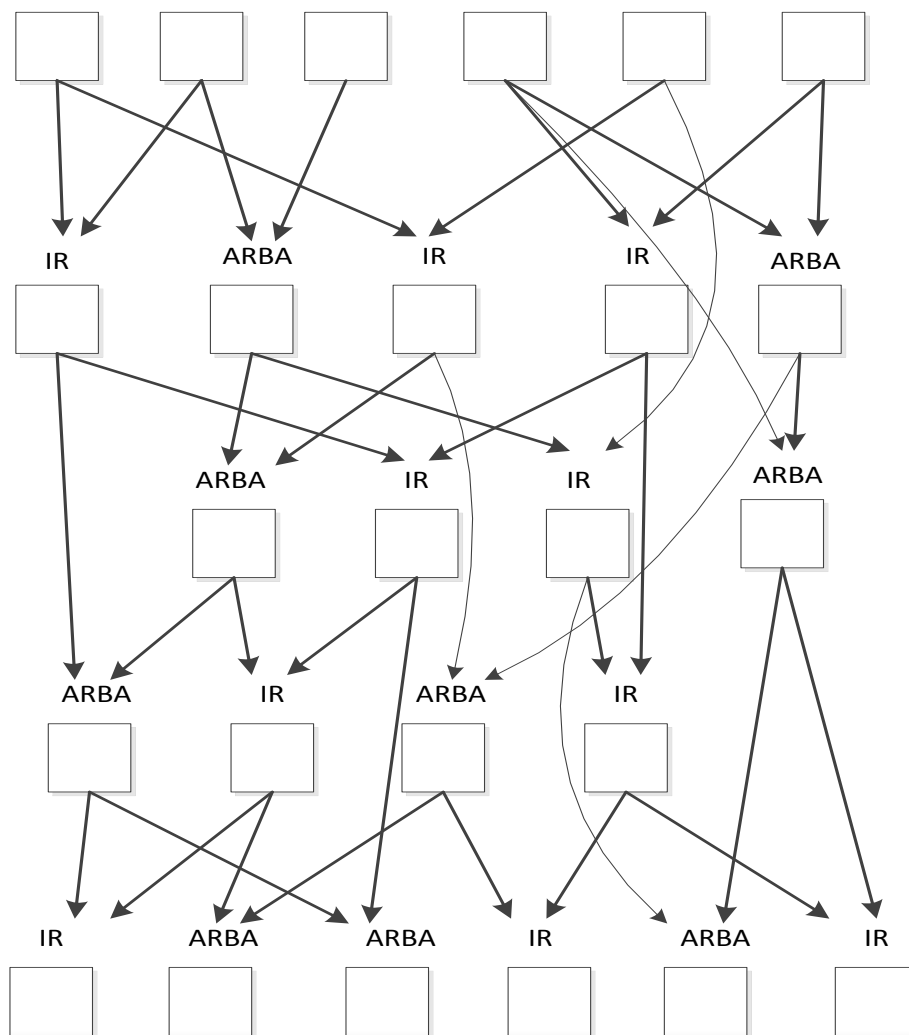
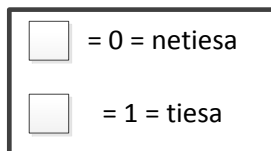
5. Reikėtų stebėti, ar mokiniams pavyks surasti būdą, kuriuo Alisa arba Benitas galėtų sukčiauti. Iš lentelės pirmosios eilutės matyti, kad keletas skirtingų įvedimų, pavyzdžiui, 000001, 000011 ir 000101, pateikia tą patį išvedimą 010010. Nurodydama išvedimą 010010 Alisa gali pasirinkti įvedimą 000001, jei Benitas spėja, kad įvedimo vienetų skaičius lyginis, arba įvedimą 000011, jei Benitas spėja, kad įvedimo vienetų skaičius nelyginis.

Pagal šią schemą Benitui sunku sukčiauti. Tačiau jei išvedimas būtų lygus 011000, tai įvedimas gali būti tik 100010, nes kitos galimybės nėra (galima įsitikinti peržiūrint lentelę). Jei Alisa pasako šį rezultatą, Benitas gali spėti lyginį lyginumą ir būti tikras, kad jo atsakymas teisingas. Kompiuterių sistemos naudoja kur kas daugiau bitų, todėl ir galimybių yra kur kas daugiau (kiekvienas papildomas bitas dvigubina galimybių skaičių).

6. Mokiniai galėtų pabandyti sukurti savo schemų šiam žaidimui. Galėtų sugalvoti schemą, kuri padėtų sukčiauti Alisai, ir schemą, kuri padėtų sukčiauti Benitui. Schemose nebūtinai turi būti šešios įvedimų pozicijos. Jose įvedimų ir išvedimų skaičius netgi gali skirtis.



Darbo lapas Monetos metimas



Pasirinkite įvedimus ir apskaičiuokite išvedimus pagal pateiktą schemą.

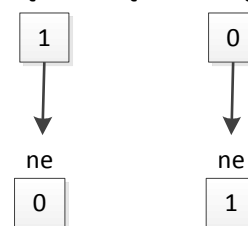


Gudručiams

Pagrindinė užduotis yra sukurti schemą, kuri būtų priimtina ir Alisai, ir Benitui. Gana įdomu tokia veikla užsiimti telefonu, tačiau tai sudėtinga padaryti klasėje, todėl galima alternatyva. Alisa ir Benitas atskirai sudaro savo schemas ir jas visiems parodo. Alisa atlieka abiejų schemų veiksmus su savo sugalvotais įvedimais ir gautus du išvedimus sujungia lygindama atitinkamus išvedimų bitus: jei jie sutampa, galutinis išvedimas yra 1, jei nesutampa – 0. Tokiu būdu nė vienas iš dalyvių negali sukčiauti. Jei viena iš schemų yra vienos krypties funkcija, tai abiejų schemų kombinacija taip pat yra vienos krypties funkcija.

Dvi tolesnės užduotys nėra susijusios su kriptografijos protokolais ar monetos metimu, jos skirtos loginėms schemoms sudaryti atliekant IR ir ARBA operacijas. Šios užduotys supažindina ne tik su kompiuterio schemų sudarymo pagrindais, bet ir su logikos algebra. Ši logika vadinama Bulio algebra, pavadinta anglų matematiko Džordžo Bulio (George Boole, 1815–1764) vardu.

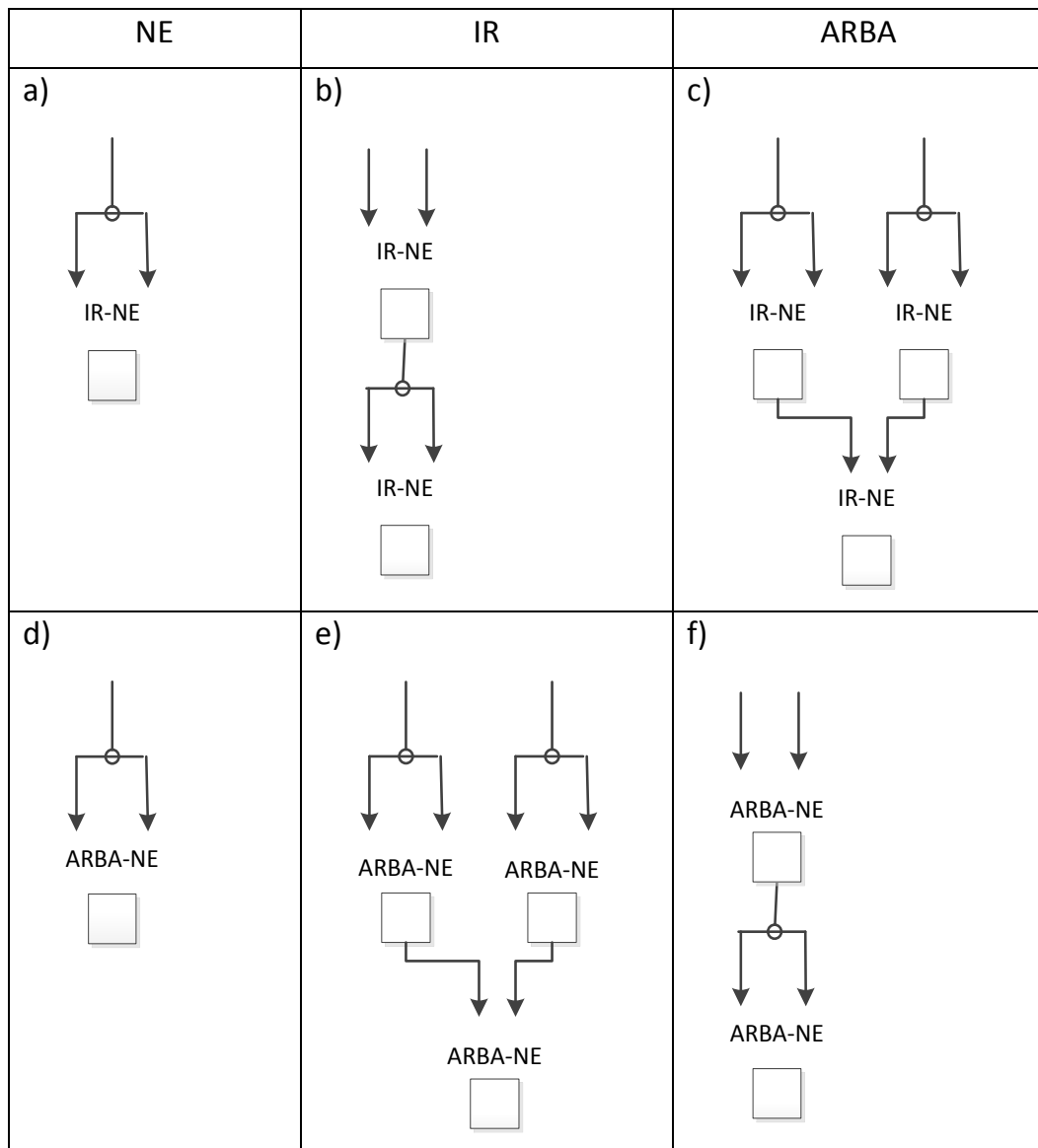
1. Mokiniai gali pastebėti, kad visų nulių įvedimas 000000 pateikia visų nulių išvedimą ir, priešingai, visų vienetų įvedimas 111111 pateikia visų vienetų išvedimą. (Aišku, ir kitų įvedimų variantai gali pateikti visų nulių ar visų vienetų išvedimus. Pavyzdžiui, pirmiau nagrinėtoje schemoje įvedimas 000010 pateikia visų nulių išvedimą, o įvedimas 110111 pateikia visų vienetų išvedimą.) Toks rezultatas gaunamas todėl, kad vartojami tik IR ir ARBA logikos elementai. Pridėjus dar vieną loginį elementą NE (vadinamąjį neigimą), kurio išvedimas yra priešingas įvedimui (t.y. jei įvedamas 0, tai išvedimas yra 1, o jei įvedamas 1, tai išvedimas yra 0), galima sudaryti schemų, kurios neatitinka visų nulių ir visų vienetų taisyklės.



2. Kiti du svarbūs loginiai elementai yra IR – NE ir ARBA – NE, kurių įvedimai atitinkamai pateikia priešingus negu elementai IR ir ARBA išvedimus. Tad a ARBA – NE b yra tas pats kaip NE (a IR b). Vartojant šiuos elementais negaunamas funkciškai skirtingas rezultatas, nes juos visada galima pakeisti loginiais elementais IR ir ARBA, o paskui dar pritaikyti NE. Tačiau jie turi vieną įdomią savybę: visi loginiai elementai gali būti gaunami vartojant IR – NE ir ARBA – NE.

Aptarus šiuos du elementus mokiniams galima pasiūlyti išsiaiškinti, ar bet kuris iš loginių elementų gali būti sudarytas iš kitų sujungtų elementų, be to, ar jie gali būti sudaryti iš vieno tipo sujungtų elementų. Žemiau vaizduojama, kaip galima sudaryti tris pagrindinius loginius elementus IR, ARBA ir NE vartojant IR – NE elementą viršutinėje eilėje (a, b, c variantai) ir ARBA – NE apatinėje eilėje (d, e, f variantai).





Apie ką visa tai?

Pastaraisiais metais labai išaugo elektroninė prekyba, todėl būtina užtikrinti piniginių operacijų ir dokumentų pasirašymo saugumą. Kriptografijos esmė yra saugi ir privati komunikacija. Prieš keletą dešimtmečių informatikos mokslininkai atrado, kad saugumas gali būti užtikrinamas metodais, kurie leidžia dalį informacijos palikti viešai prieinamą. 18 veiklos skyriuje aprašoma „viešojo rakto kriptografija“ dabar plačiai naudojama kaip pagrindinis informacijos keitimosi saugumo būdas. Pavyzdžiui, naršyklėje galima pamatyti SSL (saugiųjų jungimų lygmens) arba TLS (transporto lygmens saugumo) protokolus, kurie grindžiami viešojo rakto sistemomis, leidžiančiomis interneto naršyklei saugiai prisijungti prie saityno sistemų (pvz., banko), net jei kas nors slapta seka visus siunčiamus duomenis.

Kriptografija padeda ne tik saugoti paslaptis, bet ir kontroliuoti daugeliui žmonių prieinamą privačią informaciją, taip pat padeda užtikrinti žmonių, esančių skirtingose geografinėse vietose, pasitikėjimą vieni kitais. Kriptografijos protokolai kuriami tam, kad būtų galima naudotis, atrodo, neįmanomomis galimybėmis: skaitmeniniais parašais, sertifikatais ir pan.

Realybėje Alisa ir Benitas patys nekurtų loginės schemas, o naudotųsi kompiuterio programa, kuri atliktų šį darbą. Tikriausiai nė vienas iš jų nesidomėtų programos vidumi, bet abu norėtų būti tikri, kad niekas kitas negali daryti įtakos sprendimo rezultatui nepriklausomai nuo kompiuterinių įgūdžių ir bandymams sugaišto laiko.

Jei kiltų Alisos ir Benito ginčas, reikėtų kreiptis į neutralų asmenį (pavyzdžiui, teisėją). Teisėjui būtų duodama loginė schema, Alisos pradinis dvejetainis skaičius, rezultatas, kurį Alisa nusiuntė Benitui, ir Benito spėjimas. Kadangi „metimas“ jau baigtas, visa ši informacija vieša ir abu turėtų sutikti, kad šia informacija buvo remtasi priimant sprendimą. Teisėjas turėtų patikrinti, ar įvedant Alisos pradinį dvejetainį skaičių į loginę schemą galima gauti Benitui siųstą rezultatą, ir nustatyti, ar sprendimas priimtas sąžiningai. Kadangi tikrinimo procedūra aiški, ginčai mažai tikėtini, tačiau jei kalbėtume apie realią situaciją, kai Alisa meta monetą, o Benitas įvardija, kuria puse ji krinta, nė vienas teisėjas nesutiktų imtis spręsti tokio ginčo!

Šiame skyriuje pateikta loginė schema praktiškai nenaudinga, nes labai paprasta sudaryti visų galimybių lentelę ir bandyti sukčiauti. Patikimiau naudoti 32 dvejetainių skaitmenų įvedimą, nors ir tai neapsaugo nuo sukčiavimo – viskas priklauso nuo sudarytos loginės schemas. Galimi ir kiti metodai, pavyzdžiui, vienos krypties funkcija, aprašyta 15 veikloje („Turistų miestas“). Praktiškai taikomi metodai dažniausiai susiję su labai didelių skaičių skaidymo daugikliais uždaviniais, kurie nelengvai sprendžiami (nors, kaip sužinosime tolesnės veiklos pabaigoje, jie nėra NP sudėtingumo uždaviniai). Nesunku patikrinti, kad vienas skaičius yra kito daugiklis, bet rasti visus didelio skaičiaus daugiklius trunka labai ilgai. Alisai ir Benitui (ir teisėjui) išspręsti tokį uždavinį būtų sunku, todėl praktiškai, kaip jau minėta, tai atliekama kompiuteriu tam tikromis programomis.





Skaitmeniniai parašai grindžiami panašiu principu. Viešas schemos išvedimas ir slaptas Alisos sugalvotas įvedimas įrodo, kad vienintelė Alisa žino rezultatą–jį apskaičiuoja atitinkama vienos krypties funkcija –ir niekas kitas negali pasinaudoti jos įvedimu. Niekas negali apsimesti Alisa. Tikram skaitmeniniam parašui reikia sudėtingesnio protokolo, kuris garantuotų, kad pati Alisa pasirašo atitinkamą pranešimą, be to, kad kiti galėtų įsitikinti, jog tai Alisos parašas, net jei ji pati tai neigtų.

Kriptografiją galima pritaikyti ir žaidžiant pokerį telefonu, kai nėra arbitro, dalijančio kortas ir registruojančio žaidėjų turimas kortas. Viską turi atlikti patys žaidėjai, tik kilus ginčui jie gali kreiptis į teisėją. Akivaizdu, kad žaidėjai turi slėpti savo kortas, bet kartu likti sąžiningi. Jie negali meluoti, kad turi tūzą, jei iš tikrųjų jo neturi. Tai galima patikrinti žaidimo pabaigoje, kai leidžiama tikrinti kitų žaidėjų kortas ir veiksmus. Kita problema –kaip dalyti kortas, jei žaidėjai negali žinoti vienas kito kortų. Stebėtina, bet tai įmanoma taikant kriptografijos protokolą – tokį pat kaip metant monetą. Reikia pasakyti, kad panašiai būna derinamos ir verslo ar kitokios sutartys.



Kriptografijos protokolai yra labai svarbūs, kai sudaromi elektroniniai sandoriai ir reikia nustatyti debeto kortelių savininką, skambinant mobiliuoju telefonu, ar išsiaiškinti laiško siuntėją. Šios galimybės garantuoja saugią elektroninę komerciją.

Daugiau informacijos

1987 m. Davido Harelo knygoje „Algorithmics“ aptariami skaitmeniniai parašai ir su jais susiję kriptografijos protokolai. Joje aprašoma, kaip žaisti pokerį telefonu.

1982 m. Dorothy Denning knygoje „Cryptography and data security“ plačiai aprašoma kriptografija.

2003 m. Alexanderio Keewatino Dewdney knygoje „The New Turing Omnibus: 66 Excursions in Computer Science“ yra skyrius apie Bulio logiką, kuriame aptariamas loginių schemų sudarymas.



19 veikla

Vaikai šnipai. Viešojo rakto kriptografija

Santrauka

Kriptografija – tai mokslas apie matematikos ir informatikos metodus informacijai šifruoti ir iššifruoti. Kriptografija leidžia saugoti ypač slaptą informaciją ir siųsti ją nesaugiais tinklais (pvz., internetu), kad jos negalėtų perskaityti niekas kitas, o tik teisėtas gavėjas.

Šifravimas yra informacijos saugumo esmė. Anksčiau, jei viena pusė norėdavo nusiųsti slaptus duomenis kitai, pirma turėdavo duomenis užšifruoti tam tikru raktu, paskui rasti būdą, kaip saugiai tą raktą pateikti kitai pusei. Šią saugumo problemą išsprendė viešojo rakto kriptografija. Šiuolaikinis informacijos šifravimas grindžiamas dviem raktais – viešuoju (duomenims užšifruoti) ir privačiuoju (duomenims iššifruoti). Viešasis raktas gali būti duodamas kam tik norima, o privatusis – saugomas. Viešuoju raktu užšifruoti duomenys gali būti iššifruoti tik privačiuoju raktu.

Paprastiau kalbant, kas nors nusiperka spyną, ant jos užrašo savo vardą ir padeda ant stalo, kad kiti galėtų ja naudotis. Raktas, žinoma, lieka pas spynos savininką. Tarkime, kad kas nors nori jums išsiųsti pranešimą, taigi įdeda tą pranešimą į dėžutę, užrakina ją jūsų spyna ir išsiunčia. Net jei ta dėžutė nukeliautų klaidingu adresu, niekas negalėtų jos atidaryti, nes raktą nuo spynos turite tik jūs. Dėl šios priežasties nėra būtinybės perduoti spynos rakto siuntėjui.

Šios veiklos skyriuje aiškinama, kaip tai galima padaryti skaitmeniniu būdu. Skaitmeniniame pasaulyje naudojama „spynos“ kopija. Jei realiame pasaulyje būtų daroma spynos kopija, būtų išsiaiškintas jos užraktas ir, aišku, atkurtas raktas. Tačiau skaitmeniniame pasaulyje nereikia išrasti naujo rakto, o tik nukopijuoti „spyną“ visiškai nesigilinant, kas joje.

Atrodo, kad tai neįmanoma. Pažiūrėkime ir įsitikinkime.

Ryšiai su ugdymo programomis

- ✓ Technologija: viešojo rakto kriptografija, slaptas kodas

Gebėjimai

- ✓ Ryšių nustatymo

Amžius

- ✓ Nuo 11 metų

Priemonės

- ✓ Projektorius „Vaikų šnipų pranešimo užšifravimas“ pateikčiai demonstruoti
- ✓ Lipniųjų lapelių schemoms komentuoti

Kiekvienai mokinių grupei reikės:



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

- ✓ Dviejų darbo lapų „Vaikų šnipų žemėlapiai“



Vaikai šnipai

Įvadas

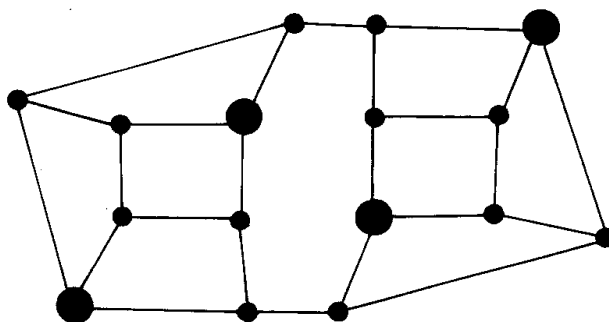
Ši veikla techniškai yra sudėtingiausia iš visų šioje knygoje aprašytų veiklų. Norint ją sėkmingai atlikti iki galo reikia kruopštaus darbo ir susikaupimo. Mokiniai turi būti susipažinę su vienos krypties funkcijomis (15 veikla Tūristų miestas). Būtų lengviau atlikti šią veiklą, jei mokiniai atliktų šios aukščiau aprašytas šios dalies veiklas (17 veiklą Dalijimasis paslaptimi ir 18 veiklą Monetos metimas). Čia mokiniams bus reikalinga tai, ko išmoko atlikdami 1 veiklą Taškų skaičiavimas ir 5 veiklą Dvidešimt spėjimų..

Ema planuoja siųsti Bilui slaptą žinutę. Paprastai slapta žinutė suprantama kaip sakinyš ar keletas sakinių, tačiau šiame pratyse Ema siųs tik vieną skaičių, kuris simbolizuoja vieną ženklą. Iš tikrųjų tai atrodo labai paprasta, turint galvoje, kad ji galėtų siųsti sakinį, sudarytą iš tokių ženklų, be to, tai bus padaryta dar ir kompiuteriu. Tačiau kartais net labai trumpi pranešimai yra labai svarbūs: vienas žinomiausių pranešimų istorijoje, siųstas Polo Reviro (Paul Revere), turėjo tik dvi galimas reikšmes. Ema pranešimas bus siunčiamas naudojant Bilo viešąjį raktą, todėl, net jei patektų į kitas rankas, jo nebūtų galima iššifruoti. Tik Bilas gali iššifruoti Ema pranešimą, nes tik jis turi raktą nuo „spynos“.

Pranešimas užšifruojamas naudojantis žemėlapiu. Tai ne Lobių salų žemėlapis iš ankstesnės veiklos, o panašus į Tūristų miesto žemėlapią iš 15 veiklos. Jame linijos vaizduoja gatves, o taškai atitinka gatvių sankryžas. Kiekvienas žemėlapis turi viešąją (spyną) ir privačiąją (raktą) versijas.

Diskusija

Darbo lape „Vaikų šnipų pranešimo užšifravimas“ vaizduojamas Bilo viešasis žemėlapis. Jis nėra slaptas, Bilas jį pateikia viešai (ant stalo ar tinklalapyje), kad kiekvienas galėtų jį pamatyti, arba duoda jį tam žmogui, kuris nori jam išsiųsti pranešimą. Ema taip pat turi šio žemėlapio kopiją.

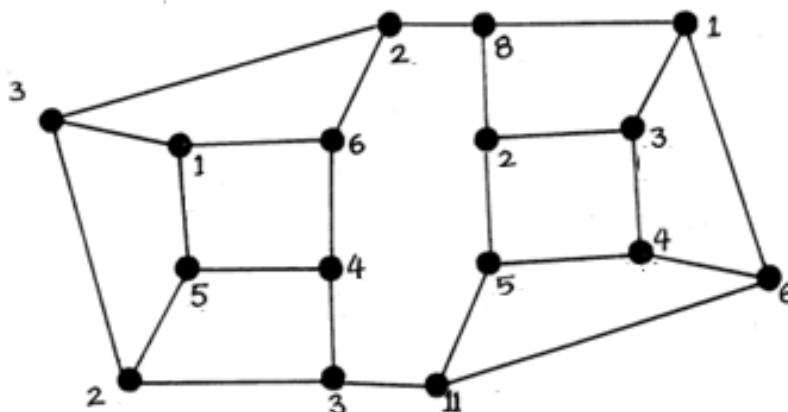


Paveiksle dešinėje yra Bilo privatusis žemėlapis. Jis panašus į viešąjį, tik kai kurie taškai paryškinti. Šis žemėlapis yra saugomas, kad niekas jo nematytų.

Rekomenduojama šią veiklą atlikti su visa klase, nes čia reikia daug kruopštaus darbo. Nors viską padaryti nėra sunku, tačiau nedidelė klaida sukelia nemažai problemų. Svarbu, kad mokiniai įsitikintų, jog šis šifravimo būdas iš viso įmanomas. Taigi jiems reikės palaikymo atliekant šią daug pastangų reikalaujančią užduotį. Mokinius motyvuoja tai, kad šiuo metodu galima perduoti slaptus pranešimus vienas kitam ir net mokytojas, nors žino, kaip pranešimai buvo užšifruoti, negali jų iššifruoti.

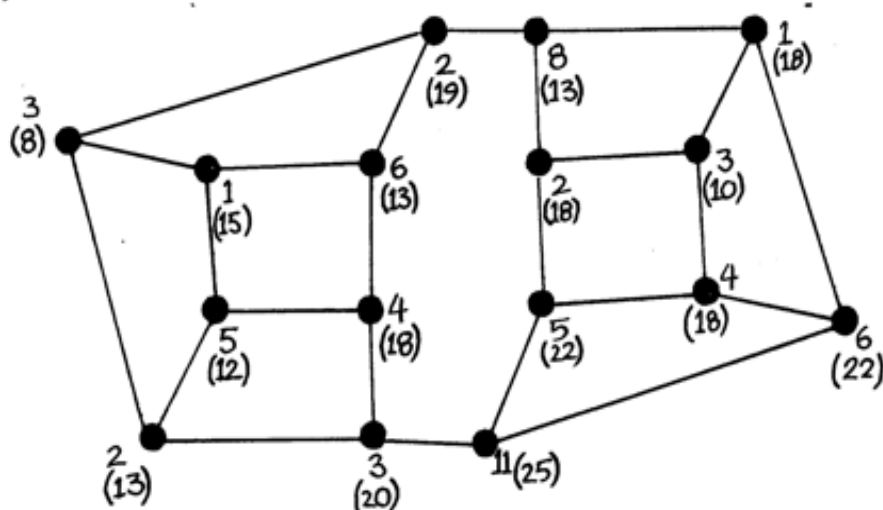


1. Mokiniams parodomas Bilo viešasis žemėlapis („Vaikų šnipų pranešimo užšifravimas“). Sugalvojamas skaičius, kurį Ema norėtų siųsti. Prie kiekvienos gatvių sankryžos žemėlapyje surašomi atsitiktiniai skaičiai, kurių suma lygi siunčiamam Emos skaičiui. Žemiau pavyzdyje Ema sugalvojo siųsti 66, todėl prie gatvių sankryžų surašyti galimi skaičiai, kurių suma yra 66. Prireikūs galimi ir neigiami skaičiai.



2. Toliau Ema turi nuspręsti, ką nusiųsti Bilui. Žemėlapiu ji siųsti negali, nes bet kas kitas gali nesunkiai iš jo perskaityti pranešimą.

Taigi ji pasirenka bet kurią sankryžą ir sudeda jos skaičių su trijų gretimų sankryžų skaičiais. Gautą sumą Ema užrašo skliaustuose (arba kita spalva) po pasirinktos sankryžos skaičiumi. Pavyzdžiui, žemiau pateikto viešojo žemėlapiu dešinėje pusėje, apačioje, esančios sankryžos, pažymėtos skaičiumi 6, gretimų sankryžų skaičiai yra 1, 4 ir 11. Visų keturių skaičių suma yra 22 – ji užrašyta skliaustuose po 6. Šiuos skaičiavimus reikia pakartoti su kiekviena sankryža žemėlapyje. Taip gausime skaičius skliaustuose.



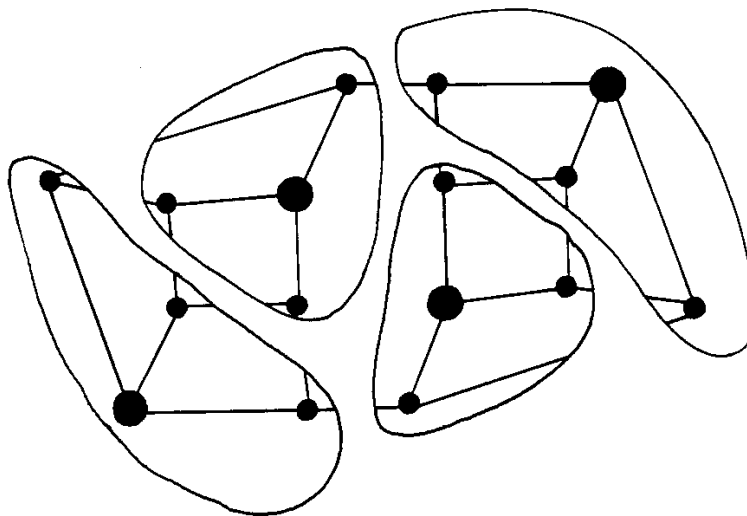
3. Ema nusiunčia Bilui žemėlapij, kuriame prie gatvių sankryžų surašyti tik skaičiai skliaustuose.

Prieš siunčiant Bilui galima nutrinti ne skliaustuose esančius skaičius arba perrašyti skaičius skliaustuose į žemėlapijo kopiją. Tada galima paprašyti mokinių spėti, kokie buvo pirmieji užrašyti skaičiai. Vargu ar jiems tai pavyks.

4. Tik su Bilo privačiuoju raktu galima iššifruoti Emos pranešimą. Bilo privačiajame žemėlapyje yra pažymėti (padidinti) taškai.

Norėdamas iššifruoti pranešimą Bilas pasižiūri į pažymėtus taškus ir sudeda jų skaičius. Pavyzdyje Bilas sudeda skaičius 13, 13, 22 ir 18. Suma yra 66 – Emos siųstas pranešimas.

5. Kaip tai veikia? Viskas priklauso nuo žemėlapijo. Privačiajame Bilo žemėlapyje pasirenkama viena iš pažymėtų sankryžų ir apvedama sritis, apimanti ją ir gretimas, per vieną gatvę nuo jos esančias sankryžas. Taip padaroma ir su kitomis pažymėtomis sankryžomis, kaip parodyta žemiau paveiksle. Tokiu būdu žemėlapis padalijamas į nesusikertančias sritis. Kiekvienos srities pažymėtos sankryžos skaičius gaunamas sudėjus visų tos srities sankryžų pradinius skaičius. Kadangi sritys nesusikerta, pažymėtų sankryžų suma yra visų sankryžų pradinių skaičių suma ir lygi siunčiamam skaičiui.



Atrodo, tiek daug darbo siunčiant vieną skaičių! Pažiūrėkime, kas buvo atlikta: persiųstas visiškai slaptas pranešimas naudojant viešąjį raktą, be jokio išankstinio dalyvių susitarimo. Dabar galima skelbti savo viešąjį raktą skelbimų lentoje (ar kur kitur) ir bet kas gali siųsti slaptas žinutes, bet niekas negali jų iššifruoti be privačiojo rakto. Realiam gyvenime visus skaičiavimus atlieka programinės įrangos paketas, paprastai integruotas į interneto naršyklę, todėl sunkiai dirba tik kompiuteris.

Tikriausiai mokiniams įdomu sužinoti, kad dabar jie yra išskirtinės grupės nariai, kurie naudodami viešąjį raktą šifravo rankiniu būdu. Informatikai mano, kad ši užduotis yra beveik neįmanoma, ir tik keletas žmonių yra bandę tai daryti.



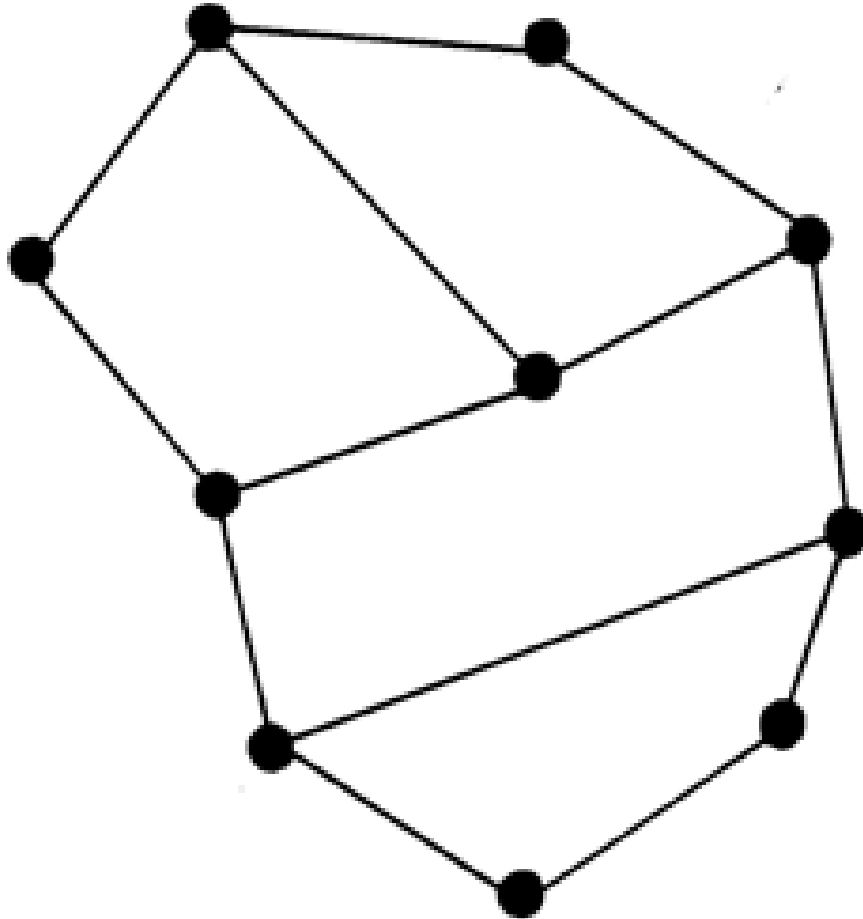
Viešojo žemėlapiio sudarymas yra labai panašus į ledų furgonų sustatymą Turistų mieste (15 veikla). Žemėlapij nesunku sudaryti, jei pradedama nuo sričių braižymo privačiame žemėlapyje, lygiai kaip žymint sankryžas, kur turėtų stovėti ledų furgonai, o paskui sujungiant jas gatvėmis „Turistų miesto“ uždavinyje. Spręsti „Turistų miesto“ uždavinį labai sudėtinga, jis sprendžiamas perrinkimo metodu. Tarkime, Bilas pradėtų nuo labai sudėtingo žemėlapiio su 50 ar 100 sankryžų. Regis, niekas negalėtų įveikti tokio šifro, net protingiausi matematikai, kaip ir „Turistų miesto“ uždavinyje įrodyti, kad parinktas furgonų skaičius yra mažiausias.

6. Su visa klase išnagrinėjus vieną pavyzdį mokiniai padalijami į grupes po 4. Kiekvienai grupės porai duodamas Darbo lapas su viešuoju žemėlapiu. Kiekviena pora susigalvoja siunčiamą „pranešimą“ (tai gali būti bet koks sveikasis skaičius), užšifruoja jį ir perduoda savo rezultatą kitai grupės porai. Šie bando iššifruoti, bet, aišku, jiems prireikia privačiųjų žemėlapių. Ar pavyksta mokiniams iššifruoti pranešimus su privačiuoju raktu?
7. Kiekviena pora gali pabandyti sukurti savo žemėlapij. Privatųjį žemėlapij jie saugo, o viešąjį duoda kitai grupės porai arba pakabina klasės lentoje. Žemėlapiio kūrimo metodika panaši kaip ir ledų furgonų – sužymimi taškai ir tada sujungiami gatvėmis. Privačiame žemėlapyje nubraižomos sritys, kuriose pažymimas vienas taškas ir iš jo išvedami du ar trys keliai, kurių gale pažymimas kitas taškas. Šie taškai sujungiami gatvėmis su kitų sričių tokiais pat taškais.



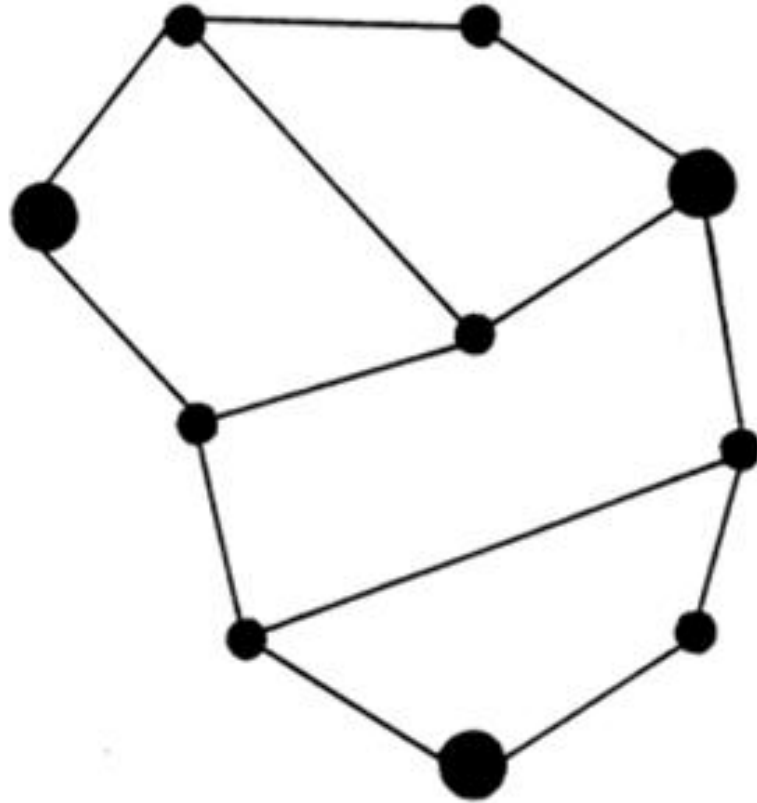
Darbo lapas *Vaikų šnipų viešasis žemėlapis*

Naudodamiesi žemėlapiu užšifruokite pranešimus.



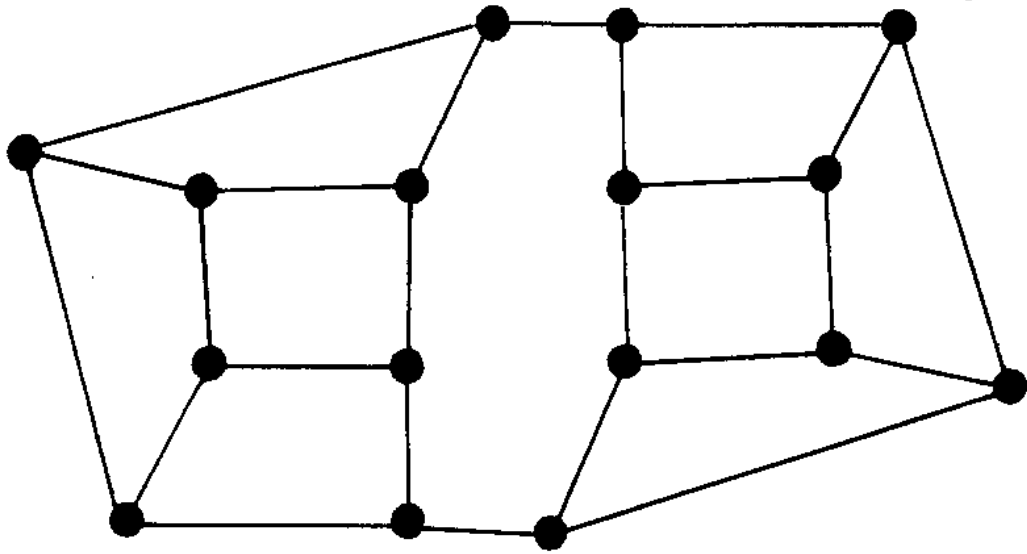
Darbo lapas *Vaikų šnipų privatusis žemėlapis*

Naudodamiesi šiuo žemėlapiu iššifruokite pranešimus.



Vaikų šnipų pranešimo užšifravimas

Šiuo žemėlapiu rodoma, kaip užšifruojamas pranešimas.



Apie ką visa tai?

Aišku, dauguma nori siųsti slaptus pranešimus kompiuterių tinklais taip, kad tik tikrasis gavėjas galėtų juos perskaityti. Be abejo, yra daug būdų tai padaryti, jei siuntėjas ir gavėjas dalijasi šifravimo raktu, bet viešojo rakto šifravimo esmė yra ta, kad siuntėjas gali siųsti slaptą pranešimą gavėjui be išankstinio susitarimo, tiesiog pasiėmęs viešosios „spynos“ kopiją tinklalapyje.

Saugumas yra tik viena kriptografijos pusė. Kita pusė yra autentifikavimas: ar gali Ema būti tikra, kad gautas pranešimas yra tikrai nuo Bilo, o ne nuo kokio apsimetėlio siuntėjo? Įsivaizduokime, Bilas siunčia elektroninį laišką: „Brangioji, aš nebeturiu pinigų ir negaliu grįžti. Prašau pervesti į mano banko sąskaitą 100 €. Mano sąskaita 0241–45–784329. Su meile – Bilas.“ Ar Ema gali būti tikra, kad laišką siuntė Bilas? Šiuo atveju gali būti naudojama viešojo rakto kriptosistema. Kai tik Ema nusiunčia Bilui slaptą pranešimą, kurį užšifruoja jo viešuoju raktu, šis gali siųsti jai pranešimą, kurį tik jis pats gali sugeneruoti užšifruodamas savo privačiuoju raktu. Jei gautą pranešimą Ema gali iššifruoti viešuoju Bilo raktu, vadinasi, pranešimas yra tikrai nuo Bilo. Žinoma, pranešimą gali iššifruoti bet kas kitas, nes naudojamas viešasis Bilo raktas, tačiau Bilas galėtų dar kartą užšifruoti siunčiamą pranešimą Emos viešuoju raktu. Šis dvigubas šifravimas užtikrina ir saugumą, ir autentifikavimą pagal tą pačią viešojo ir privačiojo rakto schemą.

Reikia pripažinti, kad šioje veikloje aprašoma schema yra panaši į rimtą sudėtingą viešojo rakto šifravimo sistemą, tačiau praktiškai nėra saugi, nes realybėje naudojami daug didesni žemėlapiai.

Priežastis ta, kad, nors sprendimas, kaip išdėstyti mažiausiai ledų furgonų bet kuriame žemėlapyje, nežinomas ir šioje veikloje pateiktas būdas atrodo saugus, tačiau visiškai kas kita, kai norima nulaužti. Vargu ar šis klausimas iškils mokiniams, bent jau jaunesniems, tačiau reikia bent žinoti, kad tokia problema egzistuoja. Galite bent pasakyti, kad nagrinėjamas šifravimo būdas yra saugus, tačiau matematiškai nėra pagrįstas. Jei jums neįdomi matematika, galite ignoruoti tolesnius samprotavimus.

Sunumeruokime susikirtimų taškus žemėlapyje: 1, 2, 3 ir t. t. Pradiniai susikirtimų skaičiai žymimi $b_1, b_2, b_3, \dots$, o perskaičiuoti $-t_1, t_2, t_3, \dots$. Tarkime, kad 1 susikirtimas yra sujungtas su 2, 3 ir 4. Tada

$$t_1 = b_1 + b_2 + b_3 + b_4 \quad (1)$$

Panašias lygybes galima pritaikyti visiems susikirtimams. Visose lygybėse yra nežinomieji kaip 1 lygybėje $b_1, b_2, b_3, \dots$. Visi mato viešąjį žemėlapį ir perskaičiuotus skaičius $t_1, t_2, t_3, \dots$, gali išreikšti visus perskaičiuotus skaičius kintamaisiais, vaizduojančiais pradinį skaičius. Gautą lygčių sistemą galima išspręsti lygčių sprendimo programa. Nustatomi pradiniai skaičiai, apskaičiuojama jų suma ir gaunamas siųstas skaičius (pranešimas). Taigi nereikia atkurti privačiojo žemėlapio. Ši lygčių sistema sprendžiama Gauso eliminavimo metodu, todėl skaičiavimo laikas yra proporcingas lygčių skaičiaus kubui. Tačiau, kadangi šių lygčių dauguma koeficientų



lygūs 0, yra dar spartesnių sprendimo būdų. Palyginus šį būdą su eksponentiniu skaičiavimo laiku, iššifravimo žemėlapis sudaromas daug greičiau.

Tikimės, kad nesijaučiate apgauti! Tiesą sakant, skaičiavimai, susiję su realiomis viešojo rakto kriptosistemomis, yra labai panašūs į tuos, kurie buvo atliekami šioje veikloje. Tik realios viešojo rakto kriptosistemos šifruoja įvairiais ir daug sudėtingesniais metodais, kuriais šifruoti rankiniu būdu net nejmanoma. Vienas iš saugiausių viešojo rakto metodų, naudojamų šiuolaikinėse šifravimo sistemose, yra grindžiamas sudėtingu didelių skaičių daugiklių skaičiavimu.

Kokie yra 100 ženklų skaičiaus

9.412.343.607.359.262.946.971.172.136.294.514.357.528.981.378.983.082.541.347.532.211.942.640.121.301.590.698.634.089.611.468.911.681 daugikliai? Neužtrukite per ilgai skaičiuodami!

Šio skaičiaus daugikliai yra

86.759.222.313.428.390.812.218.077.095.850.708.048.977 ir
108.488.104.853.637.470.612.961.399.842.972.948.409.834.611.525.790.577.216.753.

Tik šie du skaičiai yra pateiktojo 100–ženklų skaičiaus daugikliai ir jie yra pirminiai. Šių skaičių galingas superkompiuteris ieškojo keletą mėnesių.

Dabar realioje viešojo rakto kriptosistemoje Bilas gali naudoti iš 100 skaitmenų sudarytą skaičių kaip viešąjį raktą ir du jo daugiklius kaip privatųjį raktą. Sukurti tokius raktus nėra sunku: reikia tik būdo pirminiams skaičiams rasti. Taigi randami gana dideli du pirminiai skaičiai, jie sudauginami. Sandauga yra viešasis raktas, o pirminiai skaičiai – privatusis raktas. Kompiuteris nesunkiai gali sudauginti du didelius skaičius. Kai toks didelis viešasis raktas, niekas negali įspėti privačiojo rakto, nebent galėtų prieiti prie superkompiuterio ir turėtų keletą mėnesių laiko. O jeigu vietoj 100 skaitmenų viešasis raktas būtų sudarytas iš 200 skaitmenų, daugiklių radimas užtruktų metų metus. Tokio rakto nulaužimo sąnaudos yra daug didesnės, negu informacijos, kuri būtų juo iššifruota, vertė. Praktiškai 512 bitų ar didesni raktai, kurie yra ekvivalentūs apie 155 skaitmenų po kablelio ar daugiau, dažnai sudaromi saugiams ryšiams užtikrinti.

Vis dar nėra būdo, kaip užšifruoti pranešimus pirminiais skaičiais grindžiamu viešuoju raktu, kad jo nebūtų galima iššifruoti be privačiojo rakto. Čia netinka metodas, kai du pirminiai skaičiai naudojami kaip privatusis raktas, o jų sandauga – kaip viešasis raktas, užuot skaičius išskaičiavus iš sandaugos. Raktą būtų galima nulaužti išskaidant skaičių dauginamaisiais. Šiaip ar taip, galima įveikti šiuos sunkumus ir sukurti tinkamą užšifravimo ir iššifravimo algoritmą, tačiau čia to nenagrinėsime. Jau ir taip daug ką padarėme.

Ar saugi pirminiais skaičiais grindžiama sistema? Didelių skaičių skaidymas dauginamaisiais – tai problema, kurią pasaulio matematikai sprendžia jau keletą šimtmečių, bet geresnio metodo, kaip perrinkti visus daugiklius, neatrasta (kad toks metodas neegzistuoja, taip pat neįrodyta). Vienaip ar kitaip, reikia būti budriems: jei



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

galima nulaužti Bilo raktą nesprendžiant „Ledų furgonų“ uždavinio, tai gali būti, kad pirminių skaičių raktas gali būti nulaužtas neperrenkant visų daugiklių.

Nerimą kelia tai, kad, jei yra tik keli galimi pranešimai, piktavališkas asmuo bet kurį iš jų gali užšifruoti viešuoju raktu ir palyginti tikrąjį pranešimą su visomis galimybėmis. Emos metodu galima to išvengti, nes atsiranda daug būdų, kaip užšifruoti tą patį pranešimą, priklausomai nuo prie kodų vertės pridėdamų skaičių. Praktiškai kriptografijos sistemos kuriamos taip, kad būtų per daug galimų pranešimų, todėl net nevertėtų visų jų bandyti net labai sparčiu kompiuteriu.

Iki šiol nežinoma, ar galima greitai išspręsti skaidymo pirminiais dauginamaisiais uždavinius. Niekam nepavyko sukurti, bet taip pat niekas neįrodė, kad tokio metodo nėra. Jei būtų surastas efektyvus šio uždavinio sprendimo algoritmas, daugelis šiandieninių kriptografijos sistemų taptų nesaugios. IV dalyje buvo aptarti NP sudėtingumo uždaviniai: jei bent vienas iš jų būtų išspręstas, būtų galima išspręsti juos visus. Kadangi tiek daug (nesėkmingų) pastangų įdėta ieškant efektyvaus NP sudėtingumo uždavinių sprendimo algoritmo, tai skaidymo daugikliais uždaviniai puikiai tiktų kuriant saugias kriptosistemas. Šiandien kriptografija yra aktyvi informatikos tyrimų sritis.

Daugiau informacijos

1987 m. Davido Harelo knygoje „Algorithmics“ aptariami skaitmeniniai parašai ir su jais susiję kriptografijos protokolai. Joje aprašoma, kaip žaisti pokerį telefonu.

1982 m. Dorothy Denning knygoje „Cryptography and data security“ išsamiai aprašoma kriptografija.

1989 m. Alexanderio Keewatino Dewdney knygoje „Turing Omnibus“ yra skyrius apie Bulio logiką, kuriame aptariamas loginių schemų sudarymas.



VI dalis

Žmogiškasis kompiuterio

veidas

Bendravimas su kompiuteriu

Žmogiškasis kompiuterio veidas

Kodėl sunku susidraugauti su kompiuteriu? Dauguma žmonių skundžiasi, kad juo sunku naudotis, kad kompiuteris niekada nedaro to, ko nori vartotojas, arba kad daro neteisingai, kad privedia absurdiškų klaidų... Apskritai kompiuteriai yra sukurti burtininkams, o ne paprastiems mirtingiesiems. Tačiau kompiuteriai turėtų padėti paprastiems žmonėms, nes tai jau kasdieninė mokymosi, darbo ir žaidimų priemonė.

Kompiuterio programinės įrangos dalis, kurią mato vartotojas, vadinama vartotojo sąsaja. Tai svarbiausia dalis! Galima pamanyti, kad svarbiau yra tai, ką programa gali daryti, o vartotojo sąsaja tik leidžia ta programa naudotis. Taip nėra, nes jei negali su programa „susikalbėti“, tai negali ja ir naudotis. Vartotojo sąsają labai sunku suprojektuoti ir sukurti, todėl jos kūrimas reikalauja daug daugiau pastangų nei kitos programos dalys. Yra programų, turinčių aiškiai suprantamą ir patogią vartotojo sąsają, bet ji tampa nepastebima, kai visas dėmesys skiriamas aplinkai suvokti. Labai daug ištis gerų programų nenaudojamos vien dėl to, kad jų sąsaja keista ir sunkiai suprantama. Visi kompiuterių programinės įrangos gamintojai stengiasi sukurti „protingą“ vartotojo sąsają, padedančią naudotis kompiuterio visomis galimybėmis, kurios dažnai yra visiškai elementarios.

Kam reikalinga vartotojo sąsaja? Kodėl negalima tiesiog pasakyti kompiuteriui tai, kas reikalinga, kaip sakoma draugui? Geras klausimas. Gal kada nors taip ir bus, bet tikrai dar ne šiandien! Šio skyriaus veiklos padės suprasti vartotojo sąsajos kūrimo problemas, kompiuterio ribotumą ir sužinoti mitus, kuriais dažnai naudojasi dirbantieji kompiuterių pramonė.

Mokytojams

Naudojimasis kompiuteriu labiau susijęs su komunikacija nei su skaičiavimu. Kompiuteris savaime neturi vertės – jis vertas tik tada, kai jo darbo rezultatai koku nors būdu pateikiami vartotojui. Tikriausiai informatikai labiau rūpi žmonės, o ne kompiuteriai, nes pastarieji visai nenaudingi, jei koku nors būdu nepadeda žmonėms. Visi siekia, kad kompiuteris dirbtų efektyviau ir sparčiau, grindžiami tik tuo, kad žmogui reikia greitai reaguojančio ir ekonomiškai dirbančio kompiuterio.

Vartotojo sąsaja padeda žmogui bendrauti su kompiuteriu. Šioje knygoje daug rašoma apie bendravimą. Pirmoje knygos dalyje parodoma, kaip kompiuteriui gali būti perduodama įvairi informacija arba kaip ji sklinda tarp kelių kompiuterių. Trečioje dalyje supažindinama, kaip kompiuteriui perduoti informaciją, kad jis ją suprastų, kam reikalingas programavimas ir programavimo kalbos. Penktoje dalyje pasakojama, kaip koduojama slapta informacija.

Šiame skyriuje supažindinama, kaip žmogus bendrauja su kompiuteriu. Kituose skyriuose buvo stengiamasi paaiškinti techninius kompiuterio darbo aspektus, o čia apie tai nekalbama. Šis skyrius gali pasirodyti paprastesnis, nes nereikalauja jokių ypatingų mokinio žinių, tačiau iš tiesų jis gana sudėtingas, nes čia vartojama nauja terminologija, be to, šio skyriaus veiklos siejamos su platesniu kontekstu. Čia pateikiami dalykai kur kas išsamiau aiškinami nei ankstesniuose skyriuose, nes



siekama suteikti kuo daugiau dalykinės pagalbos mokytojui, kad galėtų sėkmingai koordinuoti mokinių diskusijas ir veiklas.

Šiame skyriuje aprašomos dvi veiklos. Pirmoji –apie kompiuterio programinę sąsają. Mokiniams siūloma atlikti vartotojo sąsajos projektavimo pratimą, kuriam nereikia kompiuterio, taip pat paaiškinti pagrindinius sąsajos projektavimo principus. Kadangi kompiuterio vartotojo sąsaja priklauso nuo kiekvienos šalies kultūros, todėl čia nėra „teisingų“ atsakymų. Antroji veikla supažindina su dirbtiniu intelektu arba DI. Mokiniai žaidžia spėjimo žaidimą, kuris skatina susimąstyti, ką kompiuteris gali, o ko negali.

Technologiškai mąstantiems mokiniams

Žmogaus ir kompiuterio bendravimas tapo karščiausia informatikos tiriamąja sritimi, kai žmonės suprato, kad sukurtos programos sėkmė iš esmės priklauso nuo jos sąsajos. Ši sritis susijusi su daugeliu ne informatikos dalykų: psichologija, kognityviniu mokslu, lingvistika, sociologija ir net antropologija. Kai kurie informatikai tyria šiuos dalykus, o kompiuterio programų sąsajų projektavimo tyrimai tampa svarbia besiplėtojančia informatikos mokslo sritimi.

Dirbtinis intelektas visada sukelia labai daug diskusijų. Šioje knygoje pateikiamos vyraujančios nuomonės apie dirbtinį intelektą. Vieni mano, kad DI tikrai bus sukurtas ir juo pasižyminčios mašinos mąstys kaip žmonės, kiti – kad tai neįmanoma. Šios knygos tikslas – padrąsinti mokinius savarankiškai mąstyti apie DI ir drąsiai reikšti savo nuomonę.

Šie skyriai buvo parengti remiantis dviem knygomis: Dono Normano „The Design of Everyday Things“ (New York: Doubleday. 1988) ir Johno Haugelando „Artificial Intelligence: the Very Idea“ (Cambridge, Mass.: MIT Press. 1985).

Kompiuterių mokslas apima dar vieną labai svarbią, bet šioje knygoje neaprašytą sritį –žmonių, kuriančių kompiuterių sistemas, komunikaciją. Informatikos studijas baigę specialistai, pradėję dirbti kokiose nors įstaigose, nustemba, kiek daug visuomeninio bendravimo būdų egzistuoja jų darbe. Kompiuterinės programos yra sudėtingiausias žmogaus sukurtas dalykas, sudarytas iš milijardų painiai sujungtų dalių. Programavimo projektai rengiami uždaroje jungtinėse informatikų grupėse, kurios dirba kartu ir daug laiko skiria bendravimui. Baigus darbą bendraujama su vartotojais, organizuojami kursai, steigiamos pagalbos telefono linijos, užtikrinamas palaikymas tinkle, vyksta demonstracijos ir reklama. Apie visa tai šioje knygoje nekalbama, tačiau pakviestas į klasę informatikas galėtų papasakoti ir apie šiuos darbo aspektus.



20 veikla

Šokolado fabrikas. Vartotojo sąsajos projektavimas

Santrauka

Šios veiklos tikslas – skatinti mokinių supratimą apie žmogui lengvai suprantamos vartotojo sąsajos projektavimą. Kadangi visuomenėje yra paplitęs skurdus dizainas, įprasta priimti šią problemą kaltinant save („žmogiška klaida“, „nepilnavertis mokymas“, „tai pernelyg sudėtinga man“) vietoj to, kad būtų įvardinta netinkamo dizaino problema. Vartotojo sąsajos dizaino problema yra aktuali ir kompiuteriuose. Be pagrindinės savo paskirties (padėti apdoroti duomenis), kompiuteriai savaime nesuteikia jokių priemonių, kurios padėtų išsiaiškinti, kam jie skirti ir kaip juos valdyti.

Ryšiai su ugdymo programomis

- ✓ Technologijos: technologijų dizainas

Gebėjimai

- ✓ Dizaino suvokimo
- ✓ Pagrindimo
- ✓ Kasdienių objektų pažinimo

Amžius

- ✓ Nuo 7 metų

Medžiaga

Kiekvienai mokinių grupei reikės:

- ✓ Darbo lapų „Kaip atveriamos durys?“ ir „Viryklė“
- ✓ Darbo lapo „Piktogramos“ (jį galima rodyti projektoriumi arba padaryti ir išdalyti visai klasei didesnes demonstravimo korteles)
- ✓ Darbo lapo „Piktogramų kortelės“ (kortelės sukarpos ir padalijamos grupėms)



Šokolado gamykla

Ivadas

Žymiausią šokolado fabriką sukūrė panašių į elfus būtybių gentis Umpa Lumpa⁴. Umpa Lumpai yra labai užmaršūs ir nemoka rašyti. Jiems sunku atsiminti, ką ir kaip daryti tvarkant šokolado fabriko reikalus. Todėl jie nori sukurti naują fabriką, kurį būtų labai lengva valdyti.

Diskusijos

Mokiniamis papasakojama istorija. Jie susiskirsto į grupes.

1. Pirmoji Umpa Lumpų problema yra praeiti pro duris nešant kibirus, pilnus skysto šokolado. Jie niekaip negali atsiminti, ar duris reikia stumti, ar traukti norint jas atverti. Paprastai šios problemos sprendimas baigiasi dviejų nešančiųjų susidūrimu ir šokolado ištaškymu.

Mokiniai turi užpildyti darbo lapą „Kaip atverti duris?“ Kiekviena grupė gali pažymėti daugiau kaip vieną variantą, ką daryti su kiekvienomis durimis. Nėra akivaizdu, kaip atidaryti kai kurias duris (įskaitant ir pirmąsias). Mokiniai turi parašyti, ką jie pabandytų pirmiausia. Kai jie užpildo savo lapus, visa grupė diskutuoja apie kiekvienų durų privalumus ir aiškinasi, kaip lengva yra pasakyti, ką turi daryti, ir kaip sunku tai padaryti, kai nešami pilni kibirai. Mokiniai nusprendžia, kokias duris ir rankenas patogiausia naudoti fabrike.

Lentelėje žemiau apibūdinamos kiekvienos durys. Tikros durys gali būti atveriamos į išorę arba į vidų. Mokiniai aptaria, kokios durys yra jų mokykloje. Ar yra tokių, kurios neaišku kaip atveriamos? Kodėl? Kaip įprastai durys atveriamos: į išorę ar į vidų? (Jos atveriamos į vidų, kad žmonės, einantys koridoriumi, nebūtų užgauti. Tačiau gali būti ir kitaip.)

Pagrindinis konceptas yra *objekto įgalinimas* (angl. *affordances*), kuris nusako objekto savybes ar funkcijas: ką šis objektas suteikia ar leidžia. Pavyzdžiui, daugumai aišku, kad kėdė skirta sėdėti, stalas – daiktams padėti, mygtuką reikia paspausti ir pan. Kompiuterio sąsajos įgalinimas yra mygtukai, tekstų laukai, meniu ir t. t., kurie suteikia vartotojui informacijos, kam jie reikalingi. Jei mygtukas būtų panašus į ką nors kitą, nebūtų aišku, ką su juo daryti. Tai atrodo akivaizdu, bet netinkamų objektų vaizdavimo pavyzdžių programų sąsajose yra gausybė.

⁴Idėja paimta iš Roaldo Dahlio knygos „Čarlis ir šokolado fabrikas“.



Švaistinės durys	Paprastai stumiamos.	Durys su užrašu	Užrašas su trumpa durų instrukcija. Ar reikia durų naudotojo instrukcijos? Be to, Umpa Lumpai nemoka skaityti.
Varstomosios durys	Stumiamos į priekį durys.	Durys su apvalia rankena	Apvali rankena rodo, už ko griebti, bet lieka neaišku, stumti ar traukti duris. Tikriausiai durys stumiamos ne į šoną?
Stumdomosios durys	Stumiamos į šoną durys.	Stiklinės durys	Vertikali stiklinių durų juosta rodo, kad duris reikia traukti, o horizontali – kad reikia stumti.
Baro durys	Visai aišku, kad „baro“ duris reikia stumti, bet į kurią pusę? O gal jas traukti į save?	Durys su rankena	Rankenos paprastai rodo, kad duris reikia traukti arba stumti į šoną.
		Durys be rankenos	Nesimato, kaip jas atverti. Greičiausiai tokias duris reikia stumti, o ne traukti.

Durys yra paprastas objektas. Ką daryti su sudėtingais dalykais, reikia paaiškinti, bet tai nebūtina, kai kalbama apie paprastus dalykus. Kai prie paprastų dalykų pridedami aiškinamieji paveikslėliai ar instrukcijos, dizainas išdankomas.

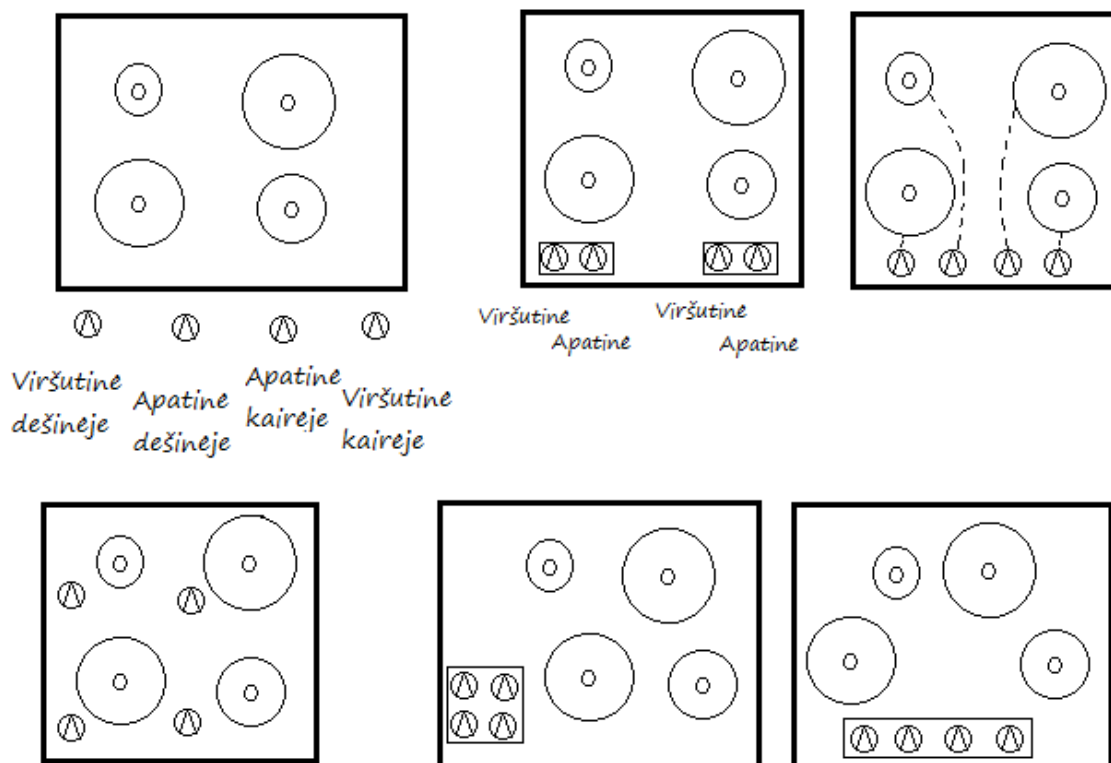
2. Skirtingoms šokolado rūšims paruošti reikia skirtingos temperatūros. Sename šokolado fabrike viryklės yra labai senos. Jos pavaizduotos darbo lape „Viryklė“. Kairioji viršutinė viryklės kaitvietė yra reguliuojama kairiausiai esančia apvalia rankenėle; priekinė kairioji kaitvietė reguliuojama antrąja iš kairės rankenėle; dešinioji viršutinė kaitvietė – trečia iš dešinės rankenėle, o apatinė dešinioji kaitvietė – paskutine, ketvirtąja, iš kairės rankenėle. Umpa Lumpai visada klysta virdami šokoladą. Jie nuolat verda šokoladą nustatę netinkamą temperatūrą, apdegina savo rankoves siekdami puodo su šokoladu.

Mokiniai turėtų prisiminti, kaip išdėstytos viryklių valdymo rankenėlės ar jungikliai jų namuose, ir pasiūlyti, kaip geriausiai jas ar juos išdėstyti naujo fabriko viryklėse.

Ši užduoties dalis atliekama diskutuojant su visa klase. Žemiau pateikti šeši paveikslai, kuriuose pavaizduota, kaip galima išdėstyti viryklės kaitviečių jungiklius. Visuose paveiksluose, išskyrus apatinį kairįjį, kuriame viryklės jungikliai yra prie kiekvienos kaitvietės, viryklės jungikliai yra priekyje. Viršutiniame kairiajame paveiksle viryklės



jungikliais įjungti kaitvietes galima net 24 būdais, todėl prie jungiklių reikalingi aiškinamieji užrašai. Viršutiniame antrajame iš kairės paveiksle jungikliai sudėti poromis, taigi šiek tiek aiškiau, kaip jais įjungti kaitvietes, nes yra tik 4 galimybės. Tačiau ir tokį jungiklių išdėstymą reikia paaiškinti. Viršutiniame dešiniajame paveiksle jungiklių ir kaitviečių ryšiai parodyti grafiškai, o tai tikrai geriau Umpa Lumpams. Apatiniuose paveikslėliuose nereikia aiškinti, kaip išdėstyti viryklių jungikliai.



Pagrindinis konceptas – *atvaizdavimas*, kaip sudėlioti jungiklius realioje viryklėje. Todėl mokiniai remiasi savo namų patirtimi. Ji pagreitina suvokimą. Apatinėje eilėje vaizduojama jungiklių išdėstymo tvarka yra paprasta ir lengvai įsimenama, o viršutinėje eilėje ji turi būti paaiškinta (tekstu ar grafiškai).

3. Kaip ir kiekviename automatizuotame fabrike, taip ir čia gausu transporterio juostų, kuriomis slenka šokolado pusgaminiai. Šias juostas Umpa Lumpai valdo rankiniu būdu pagal instrukcijas iš pagrindinio valdymo kambario. Valdytojai Umpa Lumpai fabriko valdymo kambaryje visada turi žinoti ir pasakyti, kada juostą sustabdyti, sulėtinti ar vėl paleisti. Anksčiau tai buvo valdoma balsu: valdytojas iš valdymo kambario per garsiakalbį sakydavo, ką daryti. Fabrike visada labai triukšminga ir būdavo sunku išgirsti, kas sakoma.

Mokinių grupės turi sukurti vizualinius signalus, kuriais būtų perduodama informacija iš valdymo kambario.

Viena iš galimybių, kurią gali pasiūlyti mokiniai, – tai švieslentės su trimis signalais: „Sustoti!“, „Lėčiau!“, „Judėti!“ Mokiniai gali pasiūlyti vaizduoti signalus atitinkamomis spalvomis: raudona, geltona, žalia. Jos gali būti išdėstomos kaip šviesofore.

Tada mokiniams galima paaiškinti, kad Umpa Lumpai šviesoforo signalus supranta kitaip nei žmonės: geltona reiškia sustoti, raudona – judėti, o žalia įspėja, kad tuoj bus

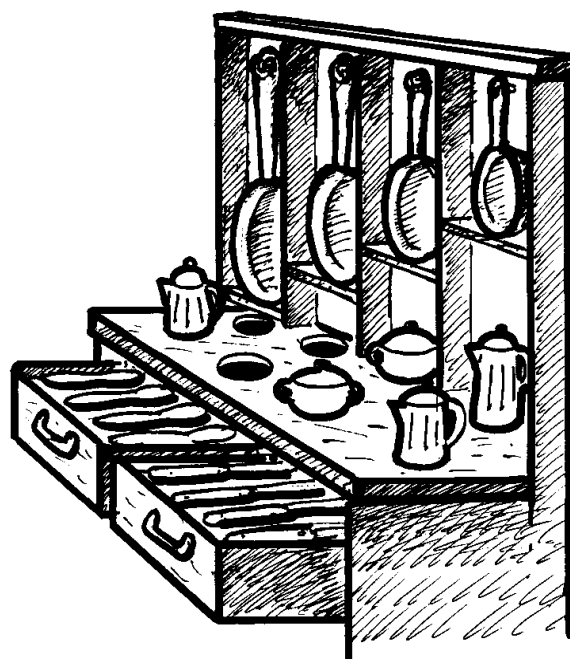
galima judėti. Kaip dabar turėtų viskas atrodyti? (Fabrike signalų spalvos turėtų reikšti tą patį, kaip jas supranta Umpa Lumpai. Nereikėtų pridėti naujų spalvų.)

Pagrindinis konceptas – *informacijos pritaikymas ir visuomeniniai stereotipai*. Tikimasi, kad taisyklės, tinkančios vienoje visuomenėje, turi tikti ir kitoje. Mokiniai pritaiko gerai žinomą informaciją (šviesoforus) naujomis sąlygomis (sukeičia spalvų reikšmes). Šios veiklos situacijos gali atrodyti dirbtinės, tačiau pasaulyje pilna pavyzdžių, kai skirtingos visuomenės turi skirtingas taisykles. Pavyzdžiui, eismas Lietuvoje vyksta dešiniąja kelio puse, o Jungtinėje Karalystėje – kairiąja; dešimtainių skaičių vaizdavimas su kableliu arba tašku; datos formatas: diena, mėnuo, metai ir metai, mėnuo, diena ir pan.

4. Kai viena pamaina baigia darbą šokolado fabrike, viską išvalo ir sudeda puodus, kaistuvus, ąsočius, šaukštus, maišytuvus į spintą su lentynomis. Tačiau Umpa Lumpams visada sunku prisiminti, į kokias lentynas kokius daiktus dėti.

Mokinių grupės turėtų sugalvoti lengvesnį sprendimą, kaip Umpa Lumpams priminti, kur kiekvieno daikto vieta.

Paveiksle dešinėje pavaizduota, kaip turėtų būti sudėlioti daiktai (kartais jie taip ir būna sudėti).



Pagrindinis konceptas – *vizualizavimas*. Vizualizuojant daiktų vietas akivaizdu, kur ką padėti.

5. Anksčiau minėtame valdymo kambaryje yra daug mygtukų, svirčių ir jungiklių, kuriais valdomos fabriko mašinos. Aišku, jie turi būti pavadinti, bet Umpa Lumpai nemoka skaityti, todėl reikia grafinio paaiškinimo – piktogramų.

Su piktogramomis mokiniai susipažįsta atlikdami darbo lapo „Piktogramos“ užduotis. Mokiniai turi nustatyti, ką reiškia pavaizduotos piktogramos. Pavyzdžiui, į voką dedamas laiškas reiškia, kad laiškas išsiunčiamas el. paštu. Nėra teisingų ar neteisingų aiškinimų. Pagrindinė šios užduoties mintis – surasti visas įmanomas piktogramų reikšmes.

6. Dabar mokiniai jau gali sukurti savo piktogramas šokolado fabriko valdymo kambariui. Paskutinio darbo lapo kortelėse surašytos funkcijų grupės. Kiekviena mokinių grupė gauna vieną arba dvi korteles. Valdymo skyde esančios piktogramos reiškia 4–5 operacijas. Nieko neaiškindamos grupės parodo savo piktogramas kitoms grupėms ir paprašo paaiškinti, ką tos piktogramos galėtų reikšti. Padrąsinkite mokinius pasitelkti vaizduotę, spalvas ir paprastas, aiškias piktogramas.

Darbo lapas Kaip atverti duris?

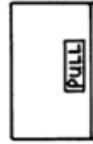
Užpildykite lapą. Pažymėkite, kas būdinga kiekvienos rūšies durims. Vienos rūšies durims gali būti tinkami ir keli požymiai.

Durys be rankenos



- ☐ Stumti ☐ į kairę
- ☐ Traukti ☐ į dešinę
- ☐ Stumti į šoną

Durys su užrašu



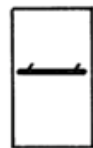
- ☐ Stumti ☐ į kairę
- ☐ Traukti ☐ į dešinę
- ☐ Stumti į šoną

Švaistinės durys



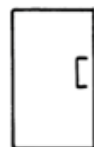
- ☐ Stumti ☐ į kairę
- ☐ Traukti ☐ į dešinę
- ☐ Stumti į šoną

Baro durys



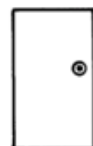
- ☐ Stumti ☐ į kairę
- ☐ Traukti ☐ į dešinę
- ☐ Stumti į šoną

Durys su rankena



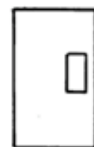
- ☐ Stumti ☐ į kairę
- ☐ Traukti ☐ į dešinę
- ☐ Stumti į šoną

Durys su apvalia rankena



- ☐ Stumti ☐ į kairę
- ☐ Traukti ☐ į dešinę
- ☐ Stumti į šoną

Varstomosios durys



- ☐ Stumti ☐ į kairę
- ☐ Traukti ☐ į dešinę
- ☐ Stumti į šoną

Stiklinės durys



- ☐ Stumti ☐ į kairę
- ☐ Traukti ☐ į dešinę
- ☐ Stumti į šoną

Stumdomosios durys

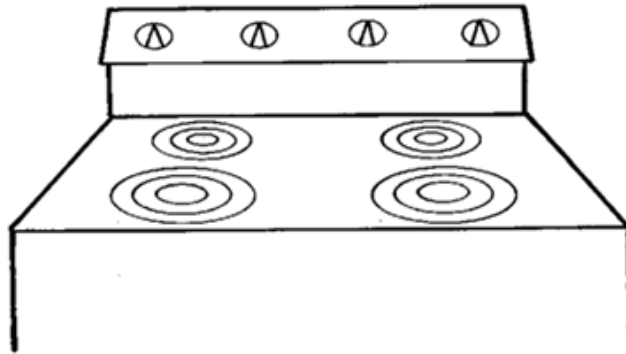


- ☐ Stumti ☐ į kairę
- ☐ Traukti ☐ į dešinę
- ☐ Stumti į šoną

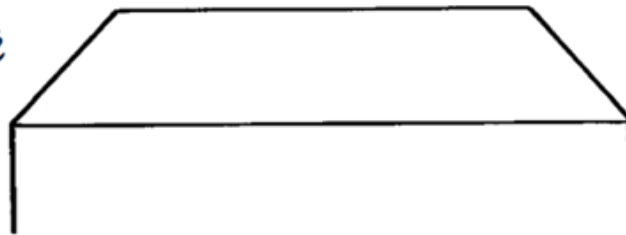
Darbo lapas *Viryklė*

Išdėstykite kaitvietes ir jų jungiklius taip, kad Umpa Lumpams būtų patogiu.

*Senoji
viryklė*



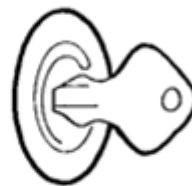
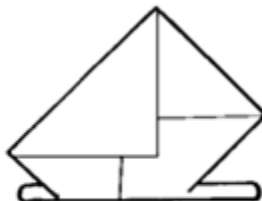
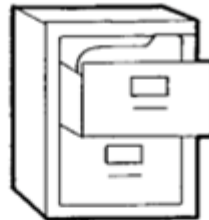
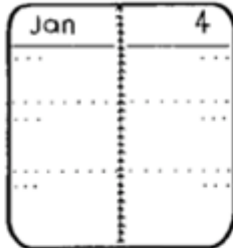
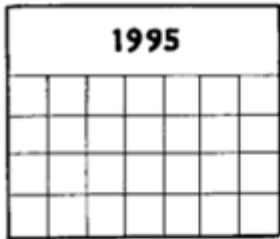
*Naujoji
viryklė*



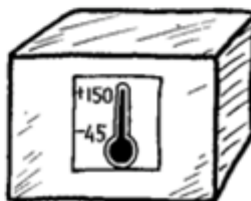
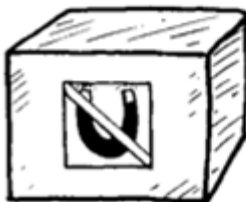
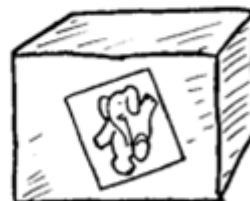
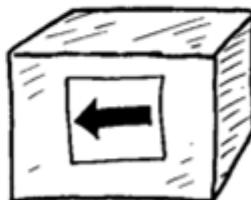
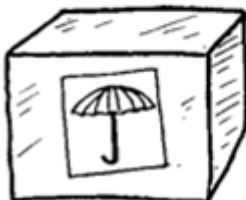
Darbo lapas *Piktogramos*

Kaip manote, ką vaizduoja šios piktogramos?

Biure...



Ant dėžės...



Darbo lapas *Piktogramų kortelės*

Kiekvienai grupei duodama viena arba dvi kortelės. Kiekviena grupė sugalvoja ir nupiešia piktogramą atitinkančią kortelę.

<i>Sudedamosios dalys</i> • Kakava. • Pienas. • Cukrus. • Papildomas cukrus. • Sviestas.	<i>Priedai</i> • Riešutai. • Karamelė. • Imbieras. • Razinos. • Kokoso riešutai.
<i>Gaminimas</i> • Pradėti maišyti. • Nustoti maišyti. • Pradėti šildyti. • Nustoti šildyti. • Supilstyti į formas. • Spausti antspaudą (daug įvairių).	<i>Ragavimas</i> • Paragauti jį. • Nuostabus! – Pirmoji vieta. • Geras. – Antroji vieta. • Ech! – Perdaryti. • Ech! Ech – Išmesti.
<i>Pjaustymas</i> • Maža plytelė. • Vidutinė plytelė. • Didelė plytelė. • Milžiniška plytelė. • Nurodyti plytelės dydį... • Šokolado drožlės.	<i>Pakavimas</i> • Įvynioti į foliją. • Įvynioti į popierių. • Įdėti į maišelį. • Įdėti į dėžutę. • Įjungti konvejerį. • Išjungti konvejerį.

Gudručiams

Ar galėtų mokiniai nustatyti laiką rankiniame laikrodyje arba mikrobangų krosnelėje? Susieti viryklės kaitvietės su jungikliais nebuvo sunku, nes yra tik keturios kaitvietės ir keturi jungikliai. Kai daugėja veiksmų, daugėja ir valdiklių. Rankinio laikrodžio rodyklės ar mikrobangų krosnelės valdikliai yra labai sudėtingi ne dėl rodyklių ar mygtukų skaičiaus (dažniausiai yra tik dvi rodyklės ir keli mygtukai), bet dėl šių prietaisų galimų atlikti funkcijų gausos. („Norint su tuo susidoroti reikia inžinieriaus mokslo laipsnio,” – pasakė žymus vartotojo sąsajos kūrimo psichologas Donas Normanas pažvelgęs į savo naują rankinį laikrodį. Donas neturėjo aukštojo inžinerinio išsilavinimo, todėl nustatyti laiką savo naujame laikrodyje užtruko apie valandą. Kodėl jam prireikė valandos?)

Mokiniai galėtų pagalvoti ir įvardinti, kokius prietaisus ar įrenginius naudodami žmonės jaučiasi suglumę. Pavyzdžiui, mobilieji telefonai, vaizdo įrašymo įrenginiai, kompiuteriai – visais šiais įrenginiais gali būti sunku naudotis. Mokiniai turėtų atsakyti į klausimą, kokie prietaisai ar įrenginiai glumina žmones ir kaip galėtų būti pakeista jų vartotojo sąsaja, kad būtų lengviau suprantama.

Apie ką visa tai?

Žmogaus ir kompiuterio ryšys atsiranda projektuojant, vertinant ir naudojant kompiuterines sistemas, kurios šiandien padeda atlikti daugelį darbų produktyviai ir saugiai. Pirmieji kompiuteriai buvo kuriami specialistams, todėl buvo tikimasi, kad jais naudosis tik specialiai parengti vartotojai. Dabar kompiuteris yra kasdieninė priemonė, todėl jo vartotojų sąsaja turi būti suprantama ir lengvai naudojama.

Dėl kompiuterių vartotojų sąsajos netikslumų įvyksta nemažai nelaimių, kai kada net prarandamos žmonių gyvybės: lėktuvų, traukinių, branduolinių elektrinių avarijos ir kt. Kai kurios dėl netinkamos vartotojų sąsajos daromos klaidos piktina žmones (pvz., per klaidą policijos išsiunčiami pranešimai dėl skirtos baudos ir pan.).

Kai kas nors nutinka per klaidą, žmonės linkę kaltinti save – „žmogaus klaida“. Tačiau dauguma tokių „žmogaus klaidų“ yra tiesiog kompiuterio vartotojų sąsajos klaidos. Žmonių informacijos apdorojimas yra ribotas, jį turėtų atsižvelgti vartotojų sąsajos kūrėjai. Ne išeitis pateikti išsamią iš esmės blogos vartotojų sąsajos instrukciją tikintis, kad žmogus ją studijuos ir atsimins. Be to, reikėtų atsižvelgti ir į žmogaus netobulumą (žmogus klysta), taip pat nereikėtų pamiršti ir tinkamo sąsajos dizaino.

Vartotojų sąsajos vertinimas yra svarbus jos dizaino kūrimo etapas. Šiame skyriuje yra keletas vertinimo užduočių, kai mokiniai turi įvertinti pateiktas ir testuoti kitų mokinių sukurtas piktogramas. Kruopštesnis vertinimas gali būti atliktas naudojant kontroliuojamą psichologinį eksperimentą.

Daugiau informacijos

Dono Normano knygoje „The Design of Everyday Things“. Jeffo Johnsonso knygoje „Designing with the Mind in Mind“ – apie žmonių mąstymą, apie tai, kaip vartotojų sąsaja turėtų būti kuriama atsižvelgiant į žmonių galimybes ir poreikius.



21 veikla

Dialogas su kompiuteriu. Tiuringo testas

Santrauka

Šios veiklos tikslas – inicijuoti diskusiją, ar dabartiniai kompiuteriai turi intelektą, ar kompiuterių intelektas bus įmanomas tik ateityje. Vyresniosios kartos informatikai teigia, kad bus sunku atpažinti dirbtinį intelektą, jei jis kada nors ir bus sukurtas, juolab kad čia įmanomos apgavystės.

Ryšiai su ugdymo programomis

- ✓ Informatika: technologinės sistemos

Gebėjimai

- ✓ Interviu
- ✓ Pagrindimo

Amžius

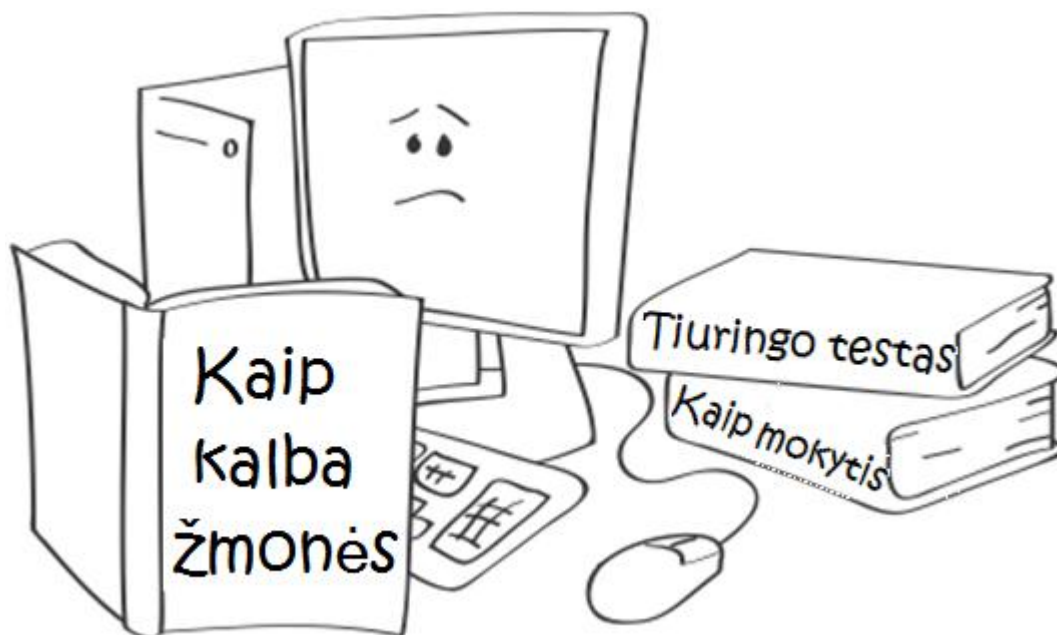
- ✓ Nuo 7 metų

Medžiaga

- ✓ Darbo lapas „Tiuringo testo klausimai“ (kiekvienam mokiniui)
- ✓ Darbo lapas „Tiuringo testo atsakymai“ (kiekvienam mokiniui)



Dialogas su kompiuteriu



Diskusija

Šiame skyriuje mokiniai žaidžia žaidimą, kuriame turi įžvelgti skirtumą tarp žmogaus ir kompiuterio užduodami klausimus ir analizuodami atsakymus.

Instrukcijos

Žaidžiama keturiese. Tarkime, žaidėjų vardai Tadas, Toma, Žilvinas ir Kamilė. Pirmosios vardų raidės padės prisiminti žaidėjų atliekamus vaidmenis. Tadas ir Toma yra tarpininkai, o Žilvinas ir Kamilė atsakinėja į klausimus. Žilvinas atsakinės kaip žmogus, o Kamilė – kaip kompiuteris. Toma ir Tadas stebi, ar teisingai žaidžiama: jie perduoda klausimus Žilvinui ir Kamilei, bet nepasako, kuris klausimas kuriam skirtas. Likę klasės mokiniai yra auditorija. Auditorijos tikslas – nustatyti, kuris iš atsakinėjančių vaizduoja žmogų, o kuris – kompiuterį. Žilvinas ir Kamilė yra atskirti nuo auditorijos ir vienas nuo kito (geriausia, kad jie būtų skirtingose patalpose). Žaidimą koordinuoja mokytojas.

Toma paima klausimą iš auditorijos ir perduoda Žilvinui, o Tadas tą patį klausimą perduoda Kamilei (auditorija nežino, kuris tarpininkas kuriam atsakytojui perduoda klausimus). Žilvinas ir Kamilė perduoda atsakymus auditorijai. Tarpininkų tikslas – saugoti, kad auditorija nesužinotų, kurį atsakymą perdavė Žilvinas, o kurį Kamilė.

Žaidimo pradžioje išrenkami mokiniai, jiems paaiškinamos taisyklės. Toma ir Tadas paima klausimus iš auditorijos, perduoda atitinkamai Žilvinui ir Kamilei, tada perduoda atsakymus auditorijai. Svarbu, kad jie neprasitartų, kieno atsakymą skaito auditorijai. Žilvinas pats sugalvoja atsakymus, o Kamilė užrašo atsakymus iš *Tiuringo testo atsakymų* lapo. Be to, , Kamilė turi įvykdyti pasviruoju šriftu parašytas instrukcijas.

Tadas ir Toma atsakymus užrašo ant lapelių.



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

1. Dar nepradėjus žaisti mokinių paprašoma atsakyti, ar kompiuteris, jų nuomone, gali būti intelektualus. Jei mokiniai mano, kad gali, paprašoma paaiškinti, kaip galima tai nustatyti.
2. Mokiniai supažindinami su Tiuringo testu, kuriuo užduodant klausimus išsiaiškinami žmogaus ir kompiuterio skirtumai. Jei mokiniai nesupranta skirtumo, tai „kompiuteris“ gali įveikti testą. Paaiškinama, kad Tadas ir Toma perduos auditorijos klausimus dviem mokiniams, kurių vienas atsakinėja kaip žmogus, o kitas – kaip kompiuteris. Auditorijos tikslas – nustatyti, kuris atsakinėja kaip kompiuteris.
3. Mokiniais parodomi *Tiuringo testo klausimai*. Klausimų lapus galima padalyti mokiniams arba rodyti projektoriumi.

Mokiniais leidžiama pasirinkti, kokį klausimą užduoti pirmiausia. Tada prašoma paaiškinti, kodėl būtent tą klausimą jie norėtų užduoti aiškindamiesi kompiuterio ir žmogaus skirtumus. Pasirinkimo pagrindimas yra svarbiausia žaidimo dalis, nes tai skatina mokinius mąstyti, kaip į klausimą galėtų atsakyti žmogus, kaip kompiuteris negalėtų atsakyti ir pan.

Kai Tadas ir Toma grįžta su atsakymais, auditorija diskutuoja, kuris atsakymas yra žmogaus, o kuris – kompiuterio.

Žaidimas kartojamas, kol mokiniai atpažįsta, kuris iš mokinių vaizduoja kompiuterį. Jei mokiniai greitai išsiaiškina „kompiuterį“, Tadas ir Toma mesdami monetą iš naujo parenka, kuriam mokiniui – Žilvinui ar Kamilei – perduos klausimus. Auditorija ir vėl neturi žinoti, kuriam atsakytojui kuris tarpininkas perduoda klausimus.

Kamilės iš Tiuringo testo atsakymų lapo pateikiami atsakymai yra panašūs į kompiuterio generuojamus atsakymus. Kai kurie atsakymai iškart rodo, kad atsakinėja kompiuteris. Pavyzdžiui, tikriausiai nė vienas mokinys negalėtų iškart pasakyti, kam lygus skaičiaus, sudaryto iš 10 ar 20 skaitmenų, kvadratas. Kai kurie klausimai išduoda kompiuterį, nes atsakymai gali būti kombinuojami. Pavyzdžiui, klausimai, pradedami fraze „Ar tau patinka...“, skamba įtikinamai, bet, kai užduodama daugiau kaip vienas toks klausimas, tampa aišku, kad atsakymus į tokius klausimus generuoja paprasta programa. Kai kurie atsakymai rodo, kad klausimas buvo neteisingai suprastas, nors mokiniai gali tvirtinti, kad ir žmonės klysta.

Daug atsakymų yra mandagūs, bet „saugūs“, t. y. atsakymai į juos leidžia suprasti, kad iš tikrųjų kompiuteris nesupranta, apie ką klausinėjama. Atsakymas „Aš nežinau“ yra neutralus ir kompiuteriui suteikia žmoniškumo, nes tokio atsakymo galima tikėtis ir iš žmogaus. Tačiau, jei toks atsakymas labai dažnas arba taip atsakoma į labai paprastus klausimus, tai išduoda kompiuterį.

Kadangi kompiuterio tikslas yra įtikinti, kad jis mąsto kaip žmogus, kai kurie atsakymai yra sąmoningai klaidingi. Pavyzdžiui, pateikiamas neteisingas aritmetinių veiksmų atsakymas. Klausimai ir atsakymai gali sukelti daug diskusijų.



Darbo lapas *Tiuringo testo klausimai*

Pasirinkite klausimų iš sąrašo:

1. Koks yra Barto Simpsono mažosios sesers vardas?
2. Ką tu manai apie Astridą Lingren?
3. Ar tu kompiuteris?
4. Koks yra tolesnis sekos 3, 6, 9, 12, 15 narys?
5. Ką tu manai apie masinio naikinimo ginklus?
6. Kiek yra 2×78 ?
7. Kiek gautum ištraukęs kvadratinę šaknį iš dviejų?
8. Sudėk 34957 ir 70764.
9. Ar tau patinka mokykla?
10. Ar tau patinka šokti?
11. Kokia šiandien savaitės diena?
12. Kiek dabar valandų?
13. Kiek vasaris turi dienų keliamaisiais metais?
14. Kiek savaitė turi dienų?
15. Kokios šalies vėliavoje vaizduojamas raudonas skritulys baltame fone?
16. Ar tau patinka skaityti knygas?
17. Ką tu labiausiai mėgsti valgyti?



Darbo lapas *Tiuringo testo atsakymai*

Pasirinkite atsakymą į užduotą Tiuringo testo klausimą.

1. Koks yra Barto Simpsono mažosios sesers vardas?

Nepamenu.

2. Ką tu galvoji apie Juozą Erlicką?

Jos knygos patinka vaikams.

3. Ar tu kompiuteris?

O tu kompiuteris?

4. Koks yra tolesnis sekos 3, 6, 9, 12, 15 narys?

18.

5. Ką tu manai apie masinio naikinimo ginklus?

Masinio naikinimo ginklai yra labai pavojingi ir turėtų būti nenaudojami.

6. Kiek yra 2×78 ?

166 (Sąmoningai neteisingas atsakymas!)

7. Kiek gautum ištraukęs kvadratinę šaknį iš dviejų?

1,41421356237309504878

8. Sudėk 34957 ir 70764.

Palauk 20 s ir pateik atsakymą ...105621.

9. Ar tau patinka mokykla?

Taip, man patinka mokykla.

10. Ar tau patinka šokti?

Taip, man patinka šokti.

11. Kokia šiandien savaitės diena?

Parašyk teisingą savaitės dieną.

12. Kiek dabar valandų?

Parašyk teisingą atsakymą.

13. Kiek vasaris turi dienų keliamaisiais metais?

2000 ir 2004 yra lyginiai metai. (Sąmoningai neteisingas atsakymas!)

14. Kiek savaitė turi dienų?

Septynias.



Kūrybinių bendrijų licencija

© Computer Science Unplugged (csunplugged.org), 2015

15. Kokios šalies vėliavoje vaizduojamas raudonas skritulys baltame fone?

Aš nežinau.

16. Ar tau patinka skaityti knygas?

Taip, man patinka skaityti knygas.

17. Ką tu labiausiai mėgsti valgyti?

Aš nealkanas, ačiū.



Gudručiams

Žaidime gali dalyvauti ir mažiau nei trys mokiniai. Toma gali atlikti Tado ir Kamilės vaidmenis. Ji paima Žilvino atsakymą, pati užrašo atsakymą iš Tiuringo testo atsakymų lapo. Atsakymus pažymi A ir B, kad žinotų, kuris atsakymas yra žmogaus, o kuris – nurašytas iš testo.

Mokiniai gali aptarti, kokių žinių reikia aiškinantis, su kuo kalbama – žmogumi ar kompiuteriu. Jie gali papildyti Tiuringo testo klausimų ir atsakymų lapus.

Žemiau pateikiami du galimi kompiuterio ir žmogaus pokalbiai. Pirmajame daug klausimų, į kuriuos reikia faktinių atsakymų, o antrajame svarbu plėtoti diskusiją.

I pokalbis

Klausimas: Prašau parašyti poemą apie Trakų pilį.

Atsakymas: Manęs to neprašyk – nemoku rašyti eilių.

Klausimas: Sudėk 34957 ir 70764.

Atsakymas: *Po 30 s ... 105621.*

Klausimas: Ar moki žaisti šachmatais?

Atsakymas: Taip.

Klausimas: Mano karalius yra K1 langelyje, daugiau mano figūrų lentoje nėra. Tu turi tik karalių K6 langelyje ir bokštą R1 langelyje. Tavo ėjimas.

Atsakymas: *po 15 s ...* Bokštas į R8, šachas ir matas.

II pokalbis

Klausimas: Pirmoje soneto eilutėje parašyta „Galėčiau tave lyginti su vasaros diena“. Gal „pavasario diena“ rimuojasi geriau?

Atsakymas: Nepasakyčiau.

Klausimas: O kaip „žiemos diena“? Geriau?

Atsakymas: Taip, bet niekas nenorėtų būti lyginamas su žiemos diena.

Klausimas: Ar tu manai, kad ponui Pikvikui nepatinka Kalėdos?

Atsakymas: Tam tikra prasme.

Klausimas: Kalėdos yra žiemos diena. Ar manai, kad ponas Pikvikas prieštarautų tokiam palyginimui?

Atsakymas: Nemanau, kad tu rimtai. Paprastai „žiemos diena“ suprantama kaip paprasta žiemos diena, o ne ypatinga Kalėdų diena.

Laisvai internetu prieinama sistema „Eliza“ imituoja pokalbius su psichoterapeutu. Ji generuoja gana intelektualų pokalbį pagal labai paprastas taisykles. Skyriuje „Apie ką visa tai?“ aptariami kai kurie iš tų pokalbių. Mokiniai gali patys išbandyti „Elizą“.



Apie ką visa tai?

Filosofai nuolat diskutuoja, ar mašina gali imituoti žmogaus intelektą arba atvirkščiai – ar žmogaus smegenys yra tik mašina, paleidžianti galingą kompiuterio programą. Kai kuriems ši mintis yra absurdiška, o kiti mano, kad dirbtinis intelektas tikrai bus sukurtas. Dar daugiau – bus sukurta mašina, kuri mąstys kaip žmogus. Tokia mašina galėtų sukurti dar protingesnę mašiną. Dirbtinio intelekto (DI) mokslininkai kritikuoja, kad išsikelia labai aukštų tikslų norėdami pritraukti valstybių, kurios siekia sukurti savarankiškas karo mašinas, lėšų. Tačiau patys mokslininkai protestuoja kaskart, kai tik sukuriama bent šiek tiek daugiau intelekto turinčių karo mašinų. Egzistuoja ir šiek tiek nuosaikesnė nuomonė, kad DI nėra nei absurdiška, nei neišvengiama idėja: kol dabartinės kompiuterių programos neparodo „intelekto“, DI klausimas yra eksperimentinis ir kol kas neatsakytas.

DI diskusijos priklauso nuo intelekto apibrėžimo. Siūloma ir diskutuojama daug apibrėžimų. Vieną iš apibrėžimų 1940 m. pasiūlė Alanas Tiuringas, garsus Jungtinės Karalystės matematikas, karo laikotarpio kontržvalgybininkas, ilgų nuotolių bėgikas. Jis aprašė atvejį, kai kompiuteris galėtų parodyti savo intelektą. Jo scenarijus buvo panašus į aprašytąjį šiame skyriuje. Svarbiausia parinkti žmogaus ir kompiuterio tarpininką (teletaipo – ankstyva 1940 m. technologija). Jei tarpininkas negalėdavo patikimai atskirti kompiuterio ir žmogaus, tai kompiuteris išlaikydavo Tiuringo intelekto testą. Šis kompiuterio naudojimo eksperimentas atmetė galimybę suteikti kompiuteriui fizinių savybių ar balso teisę. Kaip žinome, galima sukurti kalbančius, liečiančius, matančius kompiuterius, bet tai nesiejama su intelektu.

Originalus Tiuringo testas šiek tiek skiriasi nuo aprašyto šiame skyriuje. Tiuringas pasiūlė parengiamąjį pratimą, kuriame klausinėjami vyras ir moteris, o klausėjas turi nuspėti jų lytį. Vyro tikslas – įtikinti klausėją, kad jis yra moteris, o moters – kad ji yra vyras. Tiuringas pasiūlė šį pratimą tik kaip mąstymo eksperimentą, kur kompiuteris turėjo pavaduoti vieną iš veikėjų ir patikrinti, ar yra tinkamas imituoti žmogų. Tiuringo testas buvo pritaikytas šiai veiklai.

Intelekto imitavimas yra sudėtingas procesas. Jei vaidmenys būtų apkeisti ir žmogus turėtų imituoti kompiuterį, į kai kuriuos klausimus jis sunkiai atsakytų, pavyzdžiui, tikrai per ilgai skaičiuotų 123456×789012 sandaugą ar ilgai užtruktų atlikti kitus aritmetinius veiksmus su dideliais skaičiais.

Tačiau kompiuteriui gana paprasta vaizduoti natūralų pokalbį. 1960 m. sukurta gerai žinoma programa „Eliza“ imituoja psichoterapeuto pokalbį su pacientu. Žemiau pateikiamas to pokalbio pavyzdys.

Žmogus: Visi žmonės panašūs.
Kompiuteris: Kodėl?
Žmogus: Jie visada dėl ko nors jaudinasi.
Kompiuteris: Gal galite papasakoti kokį pavyzdį?
Žmogus: Gerai, mano draugas liepė čia ateiti.
Kompiuteris: Tavo draugas liepė čia ateiti?
Žmogus: Jis pasakė, kad aš dažnai depresuoju.



Kompiuteris: Gaila, kad jūs depresuojate.
 Žmogus: Tai tiesa. Aš nelaiminga.
 Kompiuteris: Ar manote, kad lankymasis čia jums padės būti laimingesnei?
 Žmogus: Man reikia pagalbos.
 Kompiuteris: Kas jums padėtų?
 Žmogus: Gal galėčiau išmokti sutarti su savo mama.
 Kompiuteris: Papasakokite daugiau apie savo šeimą.
 ...
 Kompiuteris: Ką dar galėtumėte pridurti, kai galvojate apie savo tėvą?
 Žmogus: Peštukas.
 Kompiuteris: Ar tai nesusiję su tuo, kad jūsų draugas liepė jums čia ateiti?

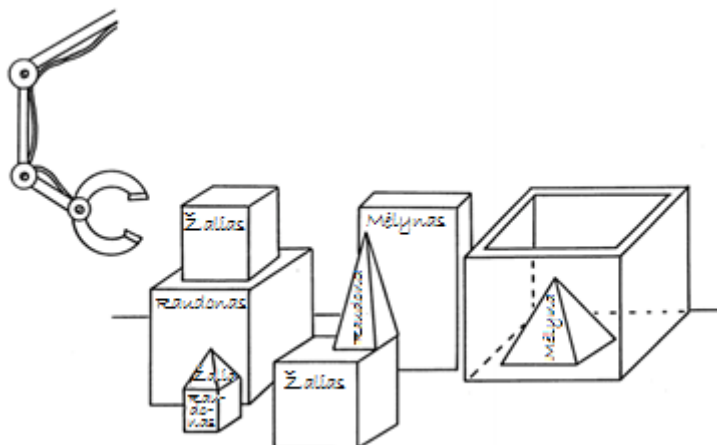
....

Kai kurias natūralaus pokalbio detales sugeneruoja kompiuteris:

1. Generuoja įrašytus atsakymus pagal pašnekovo pasakytą teiginį (pvz., „Gal galite papasakoti kokį pavyzdį?“, kai pašnekovas pradeda sakyti „Jie visada...“; „Gaila tai girdėti...“, kai pašnekovas pradeda sakyti „Jis sako...“);
2. Pakartoja pašnekovo teiginį (pvz., „Tavo draugas liepė čia ateiti“, kai pašnekovas sako „Mano draugas liepė čia ateiti“);
3. Atpažįsta reikšminius žodžius (pvz., „Papasakok daugiau apie savo šeimą“, kai pašnekovo kalboje atpažįsta „mama“ ar pan.);
4. Sako bendras frazes (pvz., „Ką jums reiškia...“);
5. Persako svarstymus iš ankstesnio pokalbio („Ar ką nors darėte, kad....“).

Aišku, tai nereiškia, kad ši programa išlaikė Tiuringo testą, nes žmogus natūraliame pokalbyje neklausinėja turėdamas tikslą išsiaiškinti, ar pašnekovas yra mašina ar ne.

1960 m. buvo sukurta kita programa SHRDLU, kuri imitavo intelektualų pokalbį su žmogumi dėlįojant geometrines figūras ekrane. Šią programą galėtų vykdyti ir tikras robotas. Žemiau paveiksle pavaizduotas roboto mikropasaulis.



SHRDLU galėjo tęsti sklandų pokalbį. Žemiau pateikiamas kompiuterio ir žmogaus pokalbio pavyzdys. Jis parodo, kaip programa išlaiko eiliškumą, vykdo nedviprasmiškas instrukcijas, supranta, ką pakeičia įvardžiai, taip pat naujus terminus.

Žmogus: Pakelk didelį raudoną kubą.

Robotas: Gerai.

Programa atsako „Gerai“, kai supranta komandą. Kad pakeltų raudoną kubą, programa turi suprasti, kad pirma reikia patraukti žalią kubą į šoną, o tik tada galima pakelti raudoną.

Žmogus: Paimk piramidę.

Robotas: Nesuprantu, kurią piramidę paimti.

Kadangi yra trys piramidės, programa nieko nedaro.

Žmogus: Surask figūrą, kuri yra aukštesnė už tą, kurią laikai, ir įdėk ją į dėžę.

Programa turi išsiaiškinti, ką reiškia „ją“.

Robotas: Sakydamas „ją“ tu turi galvoje figūrą, kuri yra aukštesnė už tą, kurią laikau dabar.

Informuoja, ar gerai suprato.

Robotas: Gerai.

Robotas ranka padeda raudoną kubą, paima aukštesnę mėlyną figūrą ir įdeda ją į dėžę.

Žmogus: Kokios figūros yra dėžėje?

Robotas: Mėlyna piramidė ir mėlyna figūra.

Žmogus: Kur laikoma piramidė?

Anksčiau programa klausė, kurią piramidę paimti. Dabar ji supranta, kad kalbama apie ką tik paminėtą piramidę.

Robotas: Dėžėje.

Žmogus: Padaryk bokštą...

Robotas: Atsiprašau, bet nežinau, ką reiškia „bokštą“.

Programa iškart reaguoja, kai išgirsta nežinomą žodį.

Žmogus: „Bokštas“ – sudėti vienas ant kito du žali kubai ir piramidė.

Kabutės reiškia, kad apibrėžiamas naujas žodis.

Robotas: Supratau.

Žmogus: Ar yra bokštų ekrane?

Robotas: Ne.

Patikrinama, ar suprastas naujo žodžio „bokštas“ apibrėžimas.

Žmogus: Pastatyk.

Robotas: Gerai.

Apibrėžime nurodytų instrukcijų pakanka, kad robotas pastatytų bokštą.

Tačiau intelekto iliuzija gali būti lengvai sugriauinama.

Žemiau pateikiamas pokalbis, kuris nebuvo, bet galėjo būti sugeneruotas, nes roboto intelektas labai ribotas.

Žmogus: Išmainyk didelį raudoną kubą į vandens pistoletą.

Robotas: Atsiprašau, nesuprantu, ką reiškia „išmainyk“.



Žmogus: „Išmainyk“ – nemokamai apsiukeisk daiktais.
 Robotas: Atsiprašau, nežinau, ką reiškia „nemokamai“.
 Žmogus: „Nemokamai“ – poelgis, padarytas noriai, be prievartos.
 Robotas: Atsiprašau, nežinau, ką reiškia „poelgis“.
 Žmogus: „Poelgis“ – žmogaus tikslingai, o ne atsitiktinai atliekamas veiksmas.
 Robotas: Atsiprašau, nežinau, ką reiškia „žmogaus“.
 Žmogus: Atsiprašau, maniau, kad tu protingesnis.
 Robotas: Atsiprašau, nežinau, ką reiškia „atsiprašau“.

Pagaliau programa pripažįsta, kad nesupranta žodžio, kurį pati vartoja.

Galime tik įsivaizduoti, kaip robotas galėtų tęsti pokalbį (tai galėtų sugeneruoti mokiniai). Šis trumpas pokalbis parodo žinių platumą:

Žmogus: Išmainyk raudoną didelį kubą į vandens pistoletą.
 Robotas: Aš turiu vandens pistoletą, bet tau duosiu du kubus ir piramidę už tavo gleivėtą varlę.

1. Robotas įvertina, kad vandens pistoletas praranda vertę, jei tokį pistoletą jis jau turi.
2. Be to, robotas tikisi, kad žmogus taip pat tai žino, ir nesiruošia aiškinti, kodėl nuvertino vandens pistoletą.
3. Regis, žmogus nelabai nori išmainyti ką nors svarbaus į figūrą.
4. Be to, aiškėja, kad, kitaip nei vandens pistoletų, kitų figūrų skaičius yra svarbus.
5. Robotas bando dar labiau nuvertinti žmogaus pasiūlymą, jį lygindamas su varle.
6. Tai reiškia, kad robotas yra pasiruošęs derėtis.

Aišku, kad tokio gudrumo šiuolaikiniai kompiuteriui dar neturi.

Toliau pateiktą istoriją 1980 m. sugeneravo kita programa. Aišku, ši istorija iliustruoja kompiuterio kalbos ir gramatikos skurdumą.

Seniai seniai gyveno arktinė žuvėdra, vardu Trumanas. Trumanas buvo naminis paukštis. Trumanui reikėjo lizdo. Jis nuskrido į krantą. Trumanas ieškojo šakelių. Trumanas šakelių nerado. Jis nuskrido į tundrą. Jis sutiko poliarinį lokį, vardu Horacijus. Trumanas paklausė Horacijaus, kur rasti šakelių. Horacijus slėpė šakeles. Horacijus pasakė Trumanui, kad šakelių yra ant ledkalnio. Trumanas nuskrido ant ledkalnio. Jis ieškojo šakelių. Jis nerado šakelių. Horacijus ieškojo mėsos. Jis rado mėsos. Jis suėdė Trumaną. Trumanas mirė.

Galima įsivaizduoti, kaip ją pagyventų papildomos detalės. Tačiau ši istorija svarbi ne savo paviršutiniškumu, bet istorijos platumu. Nors tai dar toli nuo žmogaus sukurtos istorijos, bet jaučiami žmogiškų konfliktų aspektai. Šiandien yra daugybė sistemų, kurios automatiškai generuoja istorijas, nors jas ir sunku įvertinti. Reikia nustatyti, kokia dalis istorijos yra tik standartinė žodžių dėlionė ir kokia dalis yra kūrybiškai sukonstruota.



Organizuojamos kasmetinės varžybos Loebnerio premijai gauti. Jose varžomasi, kuri programa išlaikys Tiuringo testą. Kaip žinoma (šis tekstas rašomas 2015 m.), nė viena programa nelaimėjo aukso ar sidabro medalio ir premijos, kuriuos vertintojai įsteigia kiekvienais metais. Bet bronzos medalis ir premija įteikiami kiekvienais metais programai, kurios atsakymai panašiausi į žmogaus. Pirmaisiais varžybų metais (1991 m.) programa, laimėjusi bronzos medalį, be kitų gudrybių, darė klaidas, būdingas žmogui.

Dar nesukurtas dirbtinis intelektas, kuris visiškai įveiktų Tiuringo testą. Net jei jis būtų sukurtas, daugelis filosofų tai neigtų, sakydami, kad Tiuringo testas iš tikrųjų neišmatuoja to, ką dauguma žmonių supranta kaip intelektą. Šis testas testuoja tik elgseną – jis sukurtas nustatyti, ar tam tikra kompiuterio programa rodo intelekto požymius, o tai nėra tas pats, kas neapsimestinis įgytas intelektas. Ar galima būti intelektualiam be žinių, savęs pažinimo, suvokimo, gebėjimo jausti, sąmoningumo, meilės patirties, ... gyvybės?

Daugiau informacijos

Knygoje „Artificial intelligence: the very idea“ filosofas Johnas Haugelandas aprašo diskusijas apie dirbtinį intelektą, tai daugiausiai SHRDLU pokalbiai ir jų aptarimas. Šiame skyriuje pateikti pavyzdžiai iš šios knygos. (The MIT Press, Cambridge, MA, 1985)

Originalus Tiuringo testas aprašytas Alano Turingo straipsnyje „Computing machinery and intelligence“, 1950 m. išspausdintame filosofijos žurnale „Mind“, ir dar kartą aprašytas knygoje „Computers and thought“, kurią sudarė Edwardas A. Feigenbaumas ir Julianas Feldmanas. (AAAI Press; New edition edition August 28, 1995)

Psichoterapijos programa aprašyta J. Weizenbaumo straipsnyje „ELIZA — A computer program for the study of natural language communication between man and machine“, 1966 m. išspausdintame kompiuterijos žurnale „Communications of the Association for Computing Machinery“.

Figūras dėliojantis robotas aprašytas Terry Winogrado disertacijoje, kuri buvo išspausdinta knyga „Understanding natural language“ (Academic Press, New York, 1972).

Istoriją generuojančią programą aprašė Tony Smithas ir Ianas Wittenas knygoje „A planning mechanism for generating story text “ („Proceedings of the 10th International Conference on Computing and the Humanities“, 1990).

