

Pakartok tai dar kartą!  
Teksto suspaudimas

Purvinas miestas.  
Minimalūs jungiantieji medžiai

Ledo kelias.  
Šteinerio medžiai

Šokolado fabrikas.  
Vartotojo sasajos  
projektavimas

Laivų mūšis.  
Paieškos algoritmai

# INFORMATIKA

## BE KOMPIUTERIO

Turininga informatikos mokymosi medžiaga pradinukams ir vyresniems

Kortelių keitimo magija.  
Klaidų radimas ir taisymas

Vaikai šnipai.  
Viešojo rako  
kriptografija

Tašķų skaičiavimas.  
Dvejetainiai skaičiai

Monetos metimas.  
Kriptografijos protokolai

Dialogas su kompiuteriu.  
Tiuringo testas

Apelsinų žaidimas.  
Maršruto parinkimai  
ir tinklo aklavietė

# Informatika

# BE KOMPIUTERIO

**Turininga informatikos mokymosi medžiaga  
pradinukams ir vyresniems**

Parašė Timas Bellas, Ianas H. Wittenas ir Mike'as Fellowsas

Darbui klasėje pritaikė Robynas Adamsas ir Jane McKenzie

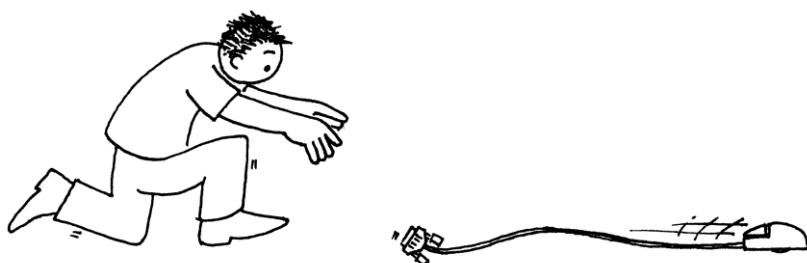
Iliustravo Mattas Powellas

2015 m. atnaujino Samas Jarmanas

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

Redagavo Audra Ivanauskienė

Viršelį kûrė Vaidotas Kinčius



<http://csunplugged.org/>



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)



Licensed under Creative Commons  
© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# Turinys

Padėka .....	6
I dalis Duomenys <i>Informacijos atvaizdavimas</i> .....	7
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> .....	57
Laivų mūšis. Paieškos algoritmai .....	59
Lengviausias ir sunkiausias. Rikiavimo algoritmai .....	78
Dar greičiau! Tinklinis rikiavimas .....	85
Purvinas miestas. Minimalūs jungiantieji medžiai.....	90
Apelsinų žaidimas. Maršruto parinkimas ir tinklo aklavietė .....	96
Akmeninės plokštelynės. Tinklo komunikacijų protokolas .....	99
III dalis Kaip pasakyti kompiuteriui, ką jis turi daryti? <i>Algoritmų užrašymas</i> .	107
Lobio ieškojimas. Baigtiniai automatai.....	109
Nurodymų eilė. Programavimo kalbos.....	125
IV dalis Ko kompiuteris negali? <i>Sunkiai sprendžiami uždaviniai</i> .....	131
Vargšas kartografas. Grafo spalvinimas.....	133
Turistų miestas. Dominuojančios sekos .....	145
Ledo kelias. Šteinerio medžiai .....	151
V dalis Informacijos šifravimas <i>Kriptografija</i> .....	161
Dalijimasis paslaptimis. Informacijos slėpimo protokolai.....	165
Monetos metimas. Kriptografijos protokolai .....	169
Vaikai šnipai. Viešojo rakto kriptografija .....	180
VI dalis Žmogiškasis kompiuterio veidas <i>Bendravimas su kompiuteriu</i> .....	193
Šokolado fabrikas. Vartotojo sąsajos projektavimas .....	196
Dialogas su kompiuteriu. Tiuringo testas.....	207



# Ivadas

Kompiuteriai mus supa visur. Visiems reikia mokytis jais naudotis, o dauguma iš mūsų be jų neapsieiname nė dienos. Kaipgi veikia kompiuteriai? Kaip jie „masto“? Kaip žmonės sukuria greitai veikiančias ir lengvai naudojamas kompiuterių programas? Informatika – nepaprastai įdomus dalykas, nagrinėjantis šiuos klaušimus. Šioje knygoje aprašomas 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 informatikos pagrindų drauge su mokiniais. Knygoje lengva ir suprantama kalba aprašoma daug veiklų, siejamų su informatikos pagrindais. Pateikiami visų užduočių atsakymai, kiekviena veikla baigiamą skyreliu „Apie ką visa tai?“, kuriame aiškinama atlirkos veiklos teorija.

Daugelis veiklų grindžiamos matematika, pavyzdžiui, dvejetainiai skaičiai, žemėlapiai ir grafa, 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, atlkti kūrybinį darbą ir mąstyti. Šios veiklos ypač lavina mokinį informatinį mąstymą, kurį skatinti siekiama moderniomis mokyklų programomis.

Daug papildomos ir laisvai prieinamos medžiagos, įskaitant animaciją ir paveikslus, pateikiama projekto *Unplugged* svetainėje [csunplugged.org](http://csunplugged.org). 2015 m. knyga atnaujinta, perdaryta ir svetainė, joje pateikta daug atvirosios medžiagos, sukurta patogesnė prieiga, labiau siejant su informatika ir informatiniu mąstymu mokykloje aiškiau išdėstyta ugdymo programas 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 šią medžiagą naudojančių mokytojų. Įsitikinta, kad dauguma svarbių informatikos konceptų gali būti mokoma be kompiuterio. Iš tiesų kartais kompiuteris pernelyg atitraukia mokinį dėmesj nuo mokymosi. Dažnai informatikos pamokos pradedamos programavimo mokymu, tačiau tai motyvuojant ne visus mokinius, todėl tokia pradžia gali tapti reikšminga kliūtimi norint sudominti mokinius šiuolaikiniaiš 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 knyga galima laisvai dalytis: ją kopijuoti, platinti ar persiųsti. Taip pat



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

leidžiama knygą atnaujinti. Visa tai galima daryti laikantis šių sąlygų: būtina nurodyti knygos autorius, nenaudoti jos komerciniais tikslais, pakeista ar pertvarkyta knyga arba šios medžiagos pagrindu sukurta nauja knyga turi būti platinama pagal panašią arba tą pačią licenciją. Daugiau informacijos apie šią licenciją galite rasti interne, 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](http://csunplugged.org)).

Knyga išversta į daugelį kalbų. Apie vertimus į kitas kalbas galite sužinoti sveitainėje [csunplugged.org](http://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. Viršelį kûrė Vaidotas Kinčius.



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://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 pasiūlymų, kaip tobulinti veiklas. Gwenda Bensemann specialiai mums parengė keleto veiklų pamoką ir pasiūlę, kaip ir ką keisti. Richardas Lyndersas ir Sumantas Murugeshas 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 Hango 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. Bellas<sup>1</sup>, Mike'as Hallettas ir Haroldas Thimbleby taip pat pasidalijo savo mintimis.

Esame dėkingi savo šeimų nariams: Bruce'ui, Fran, Grantui, Judith ir Pam už paramą ir Andrew, Annai, Hannah'i, Maxui, Michaelui<sup>2</sup> 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](http://csunplugged.org).

---

<sup>1</sup> Nesusijęs su pirmuoju autoriumi.

<sup>2</sup> Teksto suspaudimo veiklos autorius.



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# I dalis

## Duomenys

### *Informacijos atvaizdavimas*



Licensed under Creative Commons  
© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# Duomenys. Pradžių pradžia

---

## Kaip informacija laikoma kompiuteryje?

Žodis „kompiuteris“ kilęs iš lotynų kalbos žodžio „computare“, reiškiančio „skaičiuoti“ arba „sudėti“. Šiuolaikiniai kompiuteriai gali daug daugiau, nei tik suskaičiuoti. Juose gali būti kaupama ir laikoma visa biblioteka, jie gali padėti rašyti tekstus, ieškoti informacijos, groti muziką ar rodyti vaizdo įrašus. Kaipgi juose laikoma visa ši informacija? Neįtikėtina, bet kompiuteriui reikia tik dviejų ženklų: nulio ir vieneto!

## Kuo skiriasi duomenys ir informacija?

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

## 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.



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# 1 veikla

---

## Taškų skaičiavimas. Dvejetainiai skaičiai

### Santrauka

Duomenys kompiuteryje laikomi ir perduodami skaitmenų 0 ir 1 sekomis. Kaip pavaizduoti žodžius ir skaičius tik šiais dviem ženklais?

### Ryšiai su ugdymo programomis

- ✓ Matematika: skaičiai, skaičių vaizdavimas dvejetainėje sistemoje
- ✓ Matematika: algebra, begalinė seka, sekos taisyklės, dvejeto 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 dvejeto 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“

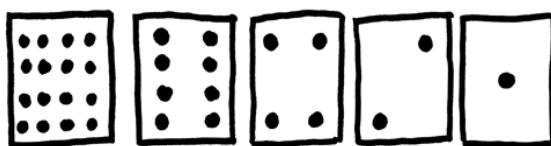


# Dvejetainiai skaičiai

---

## Jvadas

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 rodomas tokia tvarka:



## Diskusija

Iš pradžių kortelės rodamos tuščiąja puse. Atverčiama pirmoji kortelė iš dešinės stebinčių mokinį atžvilgiu. Mokinį 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)

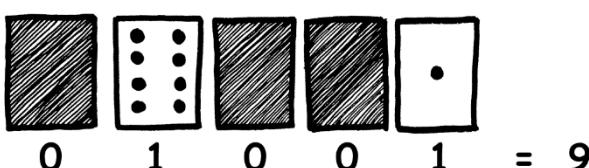
Stovintys mokiniai korteles vėl atverčia tuščiomis pusėmis į priekį. Prašoma kortelėmis pavaizduoti jvairius skaičius: 6 (atverčiamos 4 ir 2 taškų kortelės), 15 (atverčiamos 8, 4, 2 ir 1 taško kortelės). Taisyklė: nereikalingos kortelės turi būti užverstos, o reikalingos – atverstos.

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

Toliau verčiant korteles vaizduojami skaičiai iš eilės nuo nulio.

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

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



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

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

# Darbo lapas *Dvejetainiai skaičiai*

---

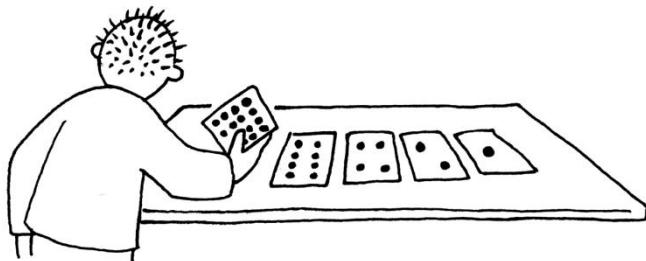
## Mokytis skaičiuoti

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 siuštį koduotas žinutes draugams taip, kaip daro kompiuteris.

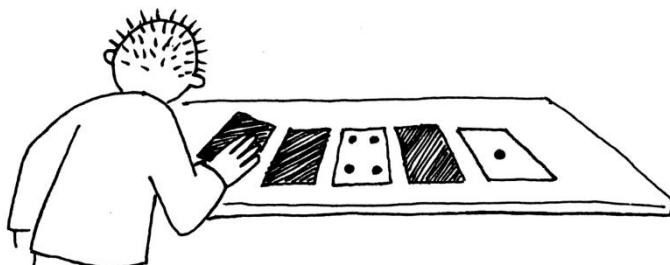
## Ivadas

Sukarpykite taškų korteles (tolesnis lapas).



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

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



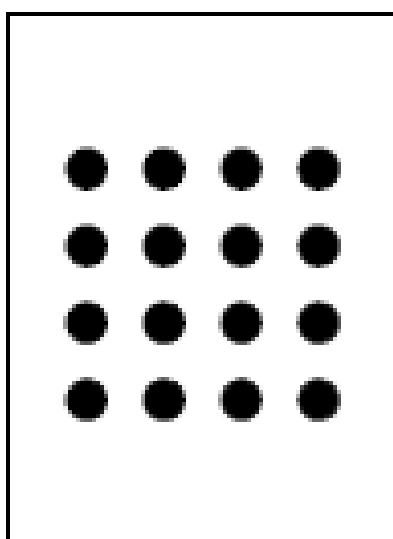
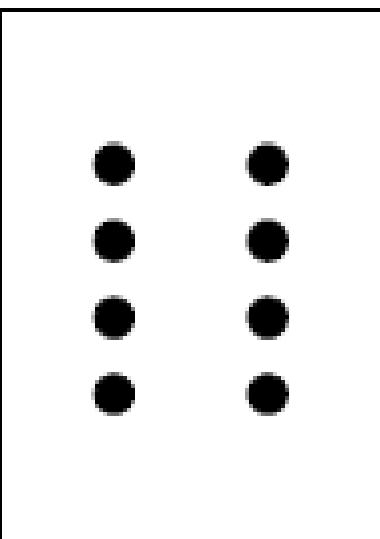
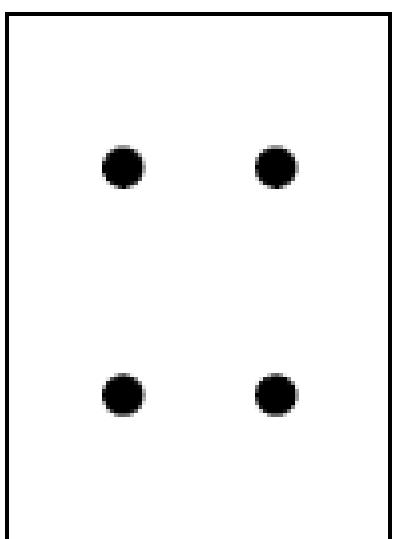
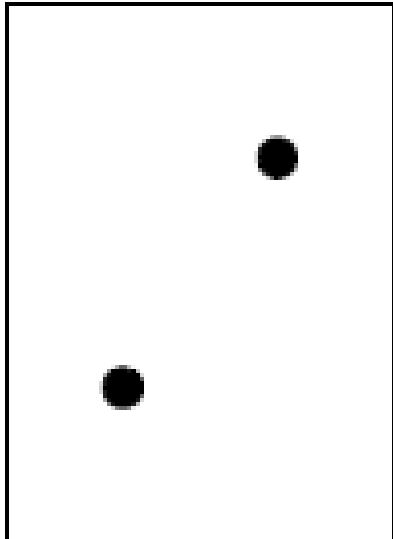
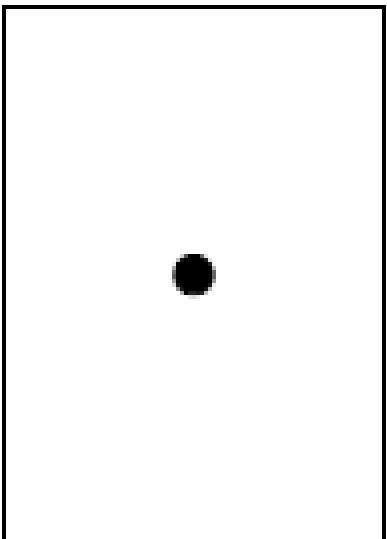
Atverskite korteles, kurių taškų suma būtų 3, 12, 19. Ar galite daugiau nei vienu būdu atversti korteles, kad gautumėte tą patį skaičių? Kokį didžiausią taškų skaičių galite atversti? Kokį mažiausią skaičių galite atversti? Ar tarp mažiausio ir didžiausio taškų skaičiaus yra skaičius, kurio negalima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



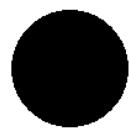
## Dvejetainiai skaičiai

---

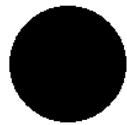
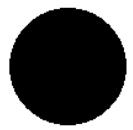


Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))



Licensed under Creative Commons  
© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

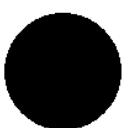
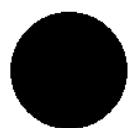
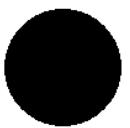
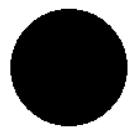
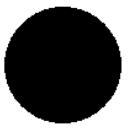
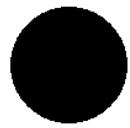


Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

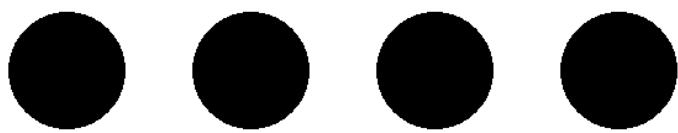
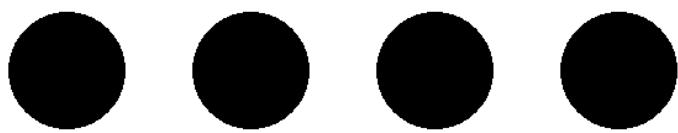
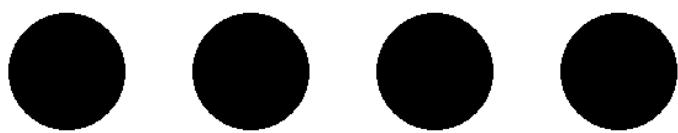
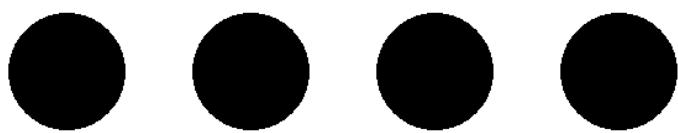


Licensed under Creative Commons  
© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))



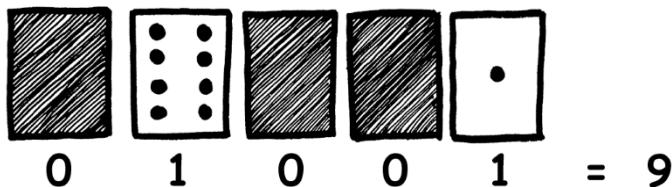
Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))



## Darbo lapas *Dvejetainiai skaičiai*

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



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

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

**Užrašykite, kuriuos dvejetainius skaičius atitinka šie ženklai:**

$$\boxed{\text{X} \ \checkmark \ \text{X} \ \text{X} \ \checkmark} = \\ (\checkmark=1, \text{X}=0)$$

$$\boxed{\text{👍} \ \text{👎} \ \text{👍} \ \text{👎}} = \\ (\text{👍}=1, \text{👎}=0)$$

$$\boxed{\uparrow \downarrow \uparrow} = \\ (\uparrow=1, \downarrow=0)$$

$$\boxed{+ + \times +} = \\ (+=1, \times=0)$$

$$\boxed{\circ \circ \circ \circ \circ} = \\ (\odot=1, \circ=0)$$

$$\boxed{\circlearrowleft \circlearrowright \circlearrowleft \circlearrowright} = \\ (\circlearrowleft=1, \circlearrowright=0)$$

$$\boxed{\square \ \square} = \\ (\square=1, \ \square=0)$$

$$\boxed{\blacktriangle \ \blacktriangledown \ \blacktriangle \ \blacktriangledown \ \blacktriangle} = \\ (\blacktriangle=1, \ \blacktriangledown=0)$$

$$\boxed{\text{😊}} = \\ (\text{😊}=1, \text{😊}=0)$$

$$\boxed{\spadesuit \ \clubsuit \ \heartsuit \ \diamondsuit} = \\ (\spadesuit=1, \clubsuit=0)$$

**Gudručiams.** Juostelėmis, kurių ilgis 1, 2, 4, 8 ir 16 matavimo vienetu, 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!



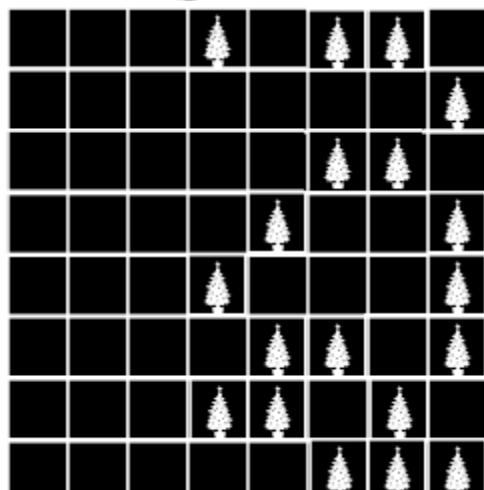
Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Darbo lapas *Slapy žinučių siuntimas*

---

Prieš pat Kalėdas Tomą atsitiktinai užrakino vienoje parduotuvėje viršutiniame pastato aukšte. Aišku, jis norėjo kuo greičiau grjžti 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?

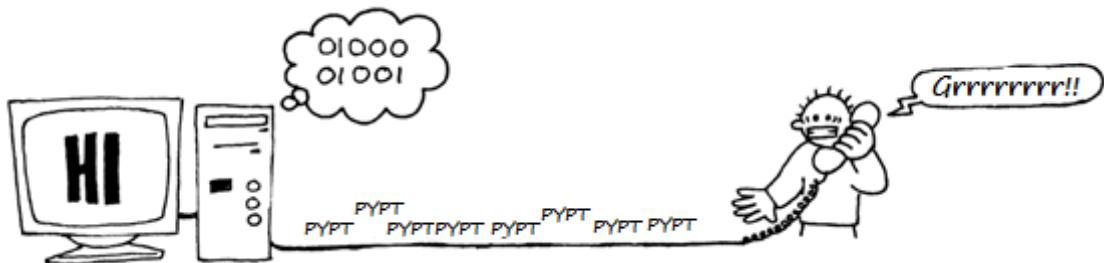


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	ą	b	c	č	d	e	ę	ė	f	g	h	i	į	γ	j
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
k	l	m	n	o	p	r	s	š	t	u	ų	ū	v	z	ž

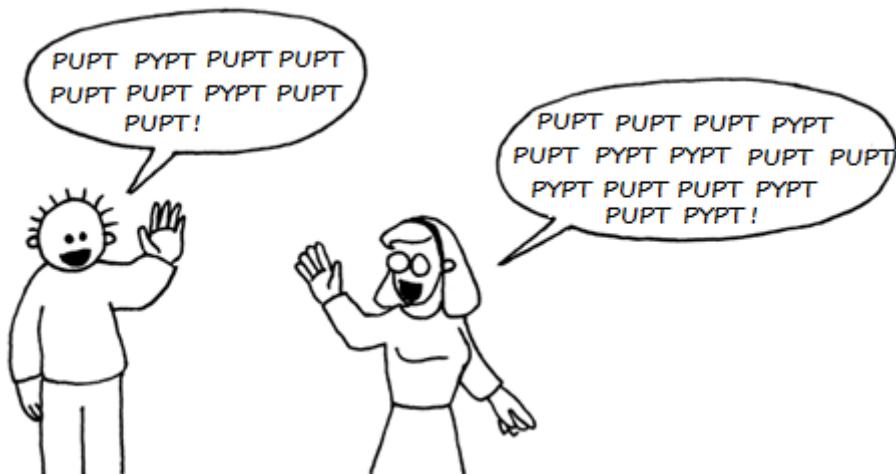
## Darbo lapas *Elektroninis paštas ir modemai*

---

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 įsklausykite 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 dvejetainės kortelės 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 jų 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 sudėjė visus skaičius nuo pirmosios iki penktosios kortelės?

Ar teko kada girdėti apie skaičiavimą „pirštais“? Išmèginkite 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 –  $32 \times 32 = 1024$  skaičius, kiek skaičių galėtų pavaizduoti keturrankis pabaisiukas?



## Darbo lapas *Dar daugiau apie dvejetainius skaičius*

---

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

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

$$\begin{array}{r} 1001 \\ (9) \end{array} \rightarrow \begin{array}{r} 10010 \\ (?) \end{array}$$

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

2. Ankstesnėse užduotyse naudojome kortèles su taškais. Kiekviena kortelė atitinka bitą kompiuteryje. Žodis „bitas“ kilęs iš anglų k. frazės „binary digit“ (liet. *dvejetainis skaitmuo*). Lietuvių kalbos abécélės raidėms (32 didžiosioms ir antra tiek mažosioms) koduoti pakaktų 6 kortelių arba, kitaip sakant, bitų. Tačiau kompiuteris turi ne tik atpažinti raides, bet skirti didžiasias ir mažiasias 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 skirtoma į 8 bitų grupes, vadinamas baitais. Vieno baimo pakanka kelių kalbų abécélėms ir dar daugeliui kitų reikalingų ženklų užkoduoti. Daugiau paskaityti apie ženklų kodavimą galite V. Dagienės, G. Grigo ir T. Jevsikovos knygoje „Programinės įrangos lokalizavimas“ (<http://www.mii.lt/files/lokalizavimas.pdf>).

Suskaičiuokite, kiek ženklų telpa į vieną baitą.

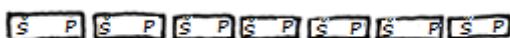


## Apie ką visa tai?

Šiuolaikiniai kompiuteriai naudoja dvejetainę sistemą informacijai kaupti ir apdoroti. Sistema vadinama dvejetaine, 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 jungtas 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 kryptimi: šiaurė → pietūs arba pietūs → šiaurė.



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



Kai skiriomas dvi reikšmės, daug paprasčiau kurti 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 tame 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 po 8 į sekas, 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, tekstu ir kitokiai informacijai laikyti ir apdoroti kompiuteris naudoja tik bitus ir baitus.

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# 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: PADÉKITE

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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: jvairių duomenų atvaizdavimas skaičiais; atminties taupymas – pasikartojančių duomenų taupesnis kodavimas

#### Gebėjimai

- ✓ Skaičiavimo
- ✓ Vaizdavimo

#### Amžius

- ✓ Nuo 7 metų

#### Priemonės

- ✓ Pateiktis „Spalvinimas skaičiais“

Kiekvienam mokinui reikia:

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# Spalvinimas skaičiais

## Ivadas

### 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 programą.)
3. Kaip kompiuteriai gali koduoti paveikslus vien skaičiais?  
(Norėdami geriau pasirengti šiai veiklai ir supažindinti su ja mokinius, galite suorganizuoti faksogramos siuntimą ar gavimą.)

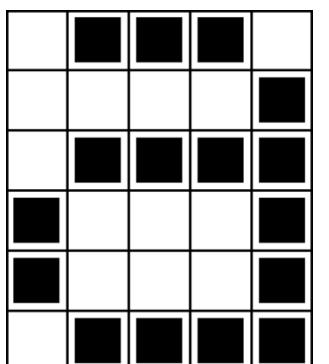
### Išsamus paaiškinimas



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

Kiekvienas nespalvoto paveikslo pikselis yra arba juodas, arba baltas.

Aukščiau pateikta raidė „a“ padidinta tam, kad įsivaizduotume pikselius. Kompiuterui 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

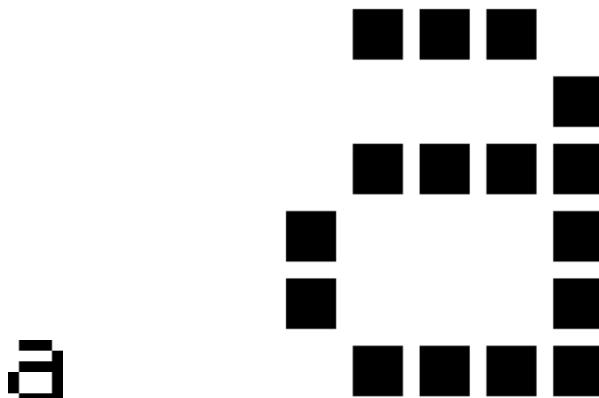
Šis piešinys rodo, kaip galima koduoti paveikslus skaičiais. Pirmoje eilutėje yra vienas baltas pikselis, trys juodi ir vėl vienas baltas, todėl pirmoji eilutė užrašoma skaičių 1, 3, 1 seka.

Pirmasis skaičius visada nusako baltujų 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

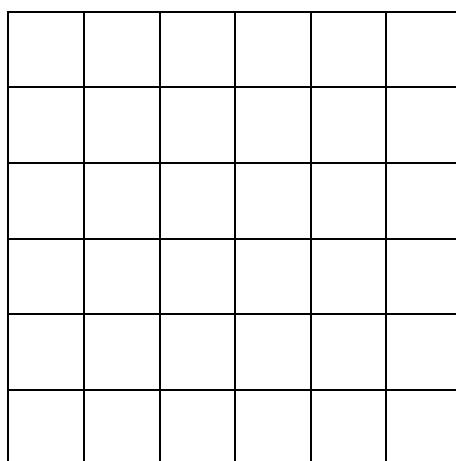
---



▲ Raidė „a“, matoma kompiuterio ekrane, ir ta pati tik 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.



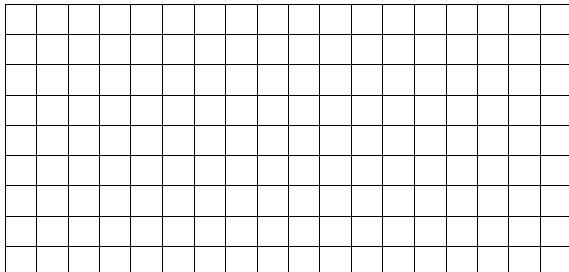
▲ Tuščias tinklelis (naudojamas mokyti)



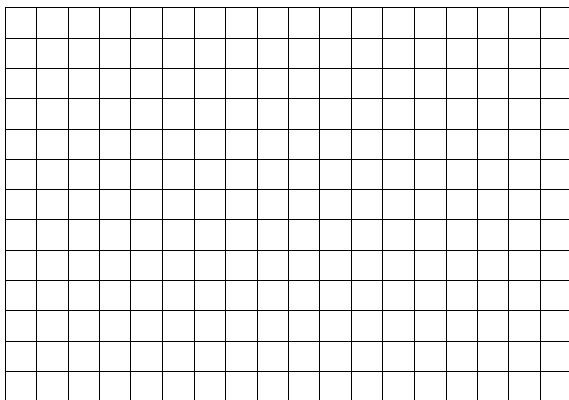
## Darbo lapas *Vaikiškas faksas*

---

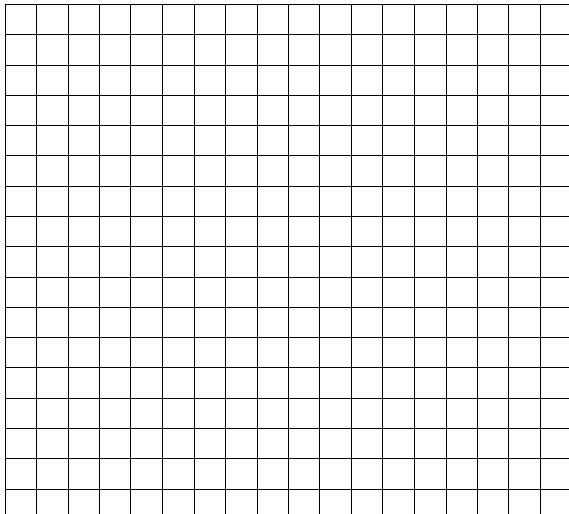
Pirmasis paveikslas yra paprasčiausias, paskutinis – sudėtingiausias. Atliekant užduotį lengva padaryti klaidą, tad patariama piešti pieštuku ir po ranka turėti trintuką!



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



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

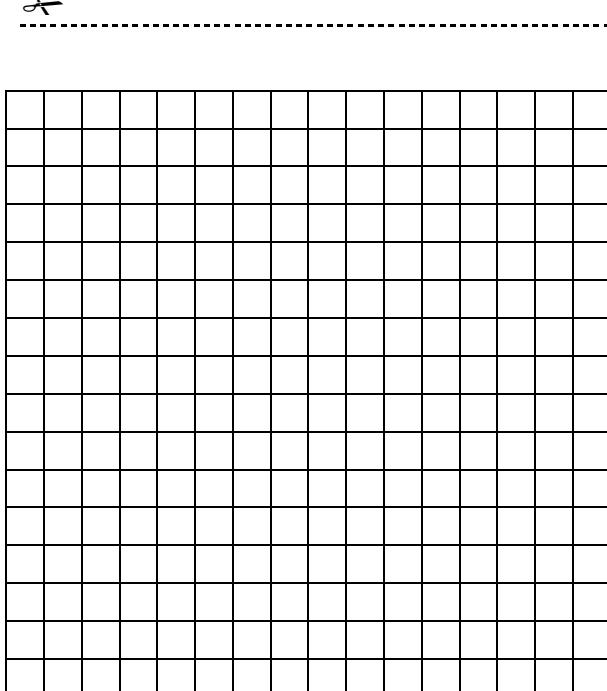
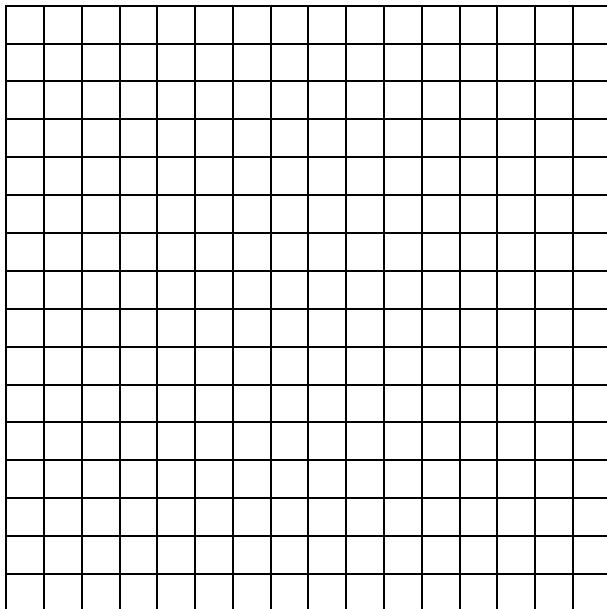
# **Darbo lapas *Nupiešk* savo paveikslą**

Dabar, kai jau žinome, kaip skaičiais galime koduoti paveikslus, kodėl nepabandžius patiemis 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 (pavyzdžiu, 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 – reikiama 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. Užuot spalvinę tinklelio langelius, mokiniai ant didesnio tinklelio gali klijuoti ar dėti lipnius kvadratinius lapelius ar kartono kvadratelius.

## **Diskusija**

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



## Apie ką visa tai?

---

Fakso aparatai yra paprasčiausi kompiuteriai, kurie nuskenuoja nespalvotą popieriaus lapą į maždaug  $1000 \times 2000$  pikselių tinklėlį. Tada tą nuskenuotą tinklėlį fakso aparatas per modemą siunčia į kitą fakso aparatą, kuris išspausdina pikselius ant popieriauslapo. Dažnai faksu siunčiamuose paveiksluose yra didelių baltų (pavyzdžiu, paraščių) ir juodų pikselių blokų (pavyzdžiu, linijų). Spalvoti paveikslai dažniausiai turi labai daug pasikartojančių dalių. Jvairiais suspaudimo metodais programuotojai siekia suraupyti atminties tokiems paveikslams laikyti. Metodas, aprašomas šiame skyriuje, vadinamas *vienodu eilėje kodavimu*. Tai labai efektyvus būdas paveikslui suspausti. Jei paveikslų nesuspaustumė, jų siuntimas užtruktų kur kas ilgiau, be to, jiems laikyti reikėtų daugiau atminties. Tada siųsti paveikslus faksu ar jidéti juos į tinklalapį būtų beveik neįmanoma. Pavyzdžiu, 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, pavyzdžiu, JPEG, GIF ir PNG). Tokių nuotraukų diske galima laikyti daugiau, o jų peržiūra tinklalapyje trunka tik dalį sekundės.

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



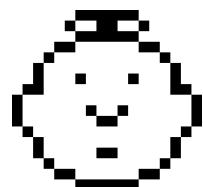
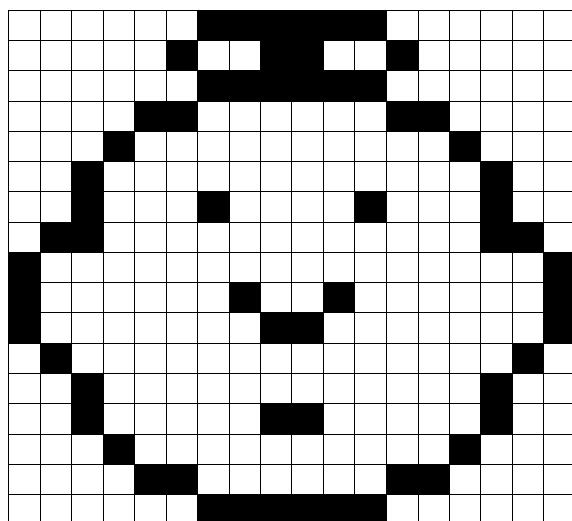
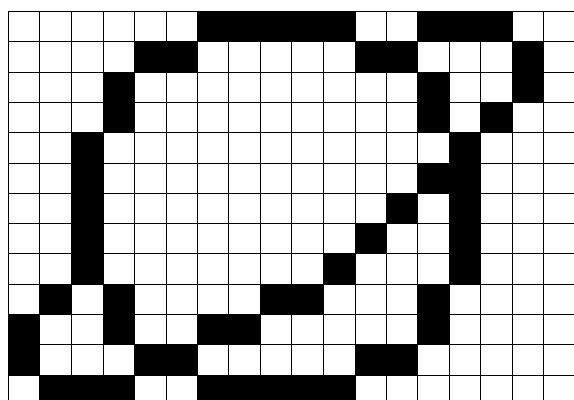
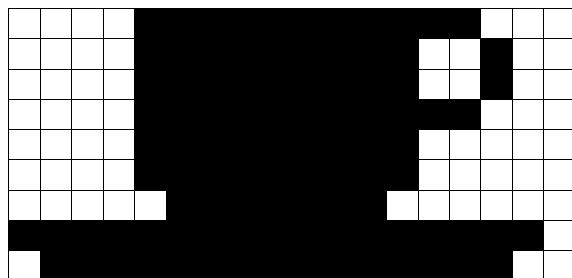
Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## **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. Užkoduotų (t. y. suglaudintų) norimų saugoti duomenų, kurie išskleidžiami, kai prieikia, kompiuteris 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

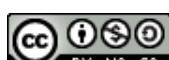
- ✓ Nuo 9 metų

#### Priemonės

- ✓ Pateiktis „Pakartok tai dar kartą!“

Kiekvienam mokinui būtinės:

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# Pakartok tai dar kartą!

---

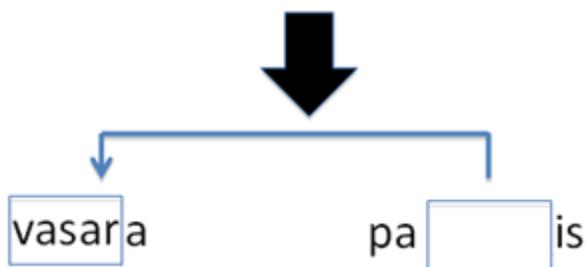
## Ivadas

Kompiuteriai laiko ir persiunčia daugybę duomenų. Kad jiems nereikėtų labai daug atminties arba kad pernelyg ilgai neužstruktūr informacijos siuntimas, kompiuteriai suspaudžia tekštą panašiai, kaip aprašoma šioje veikloje.

## Diskusija ir pavyzdžiai

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

**vasara pavasaris**



**Pakartok tai dar kartą!**

---

**Saulė**

**Vasara pavasaris**

**Vasara pavasaris**

**Saulė šypsosi**

**Vasara pavasaris**

**Vasara pavasaris**

**I laukelį kviečia**

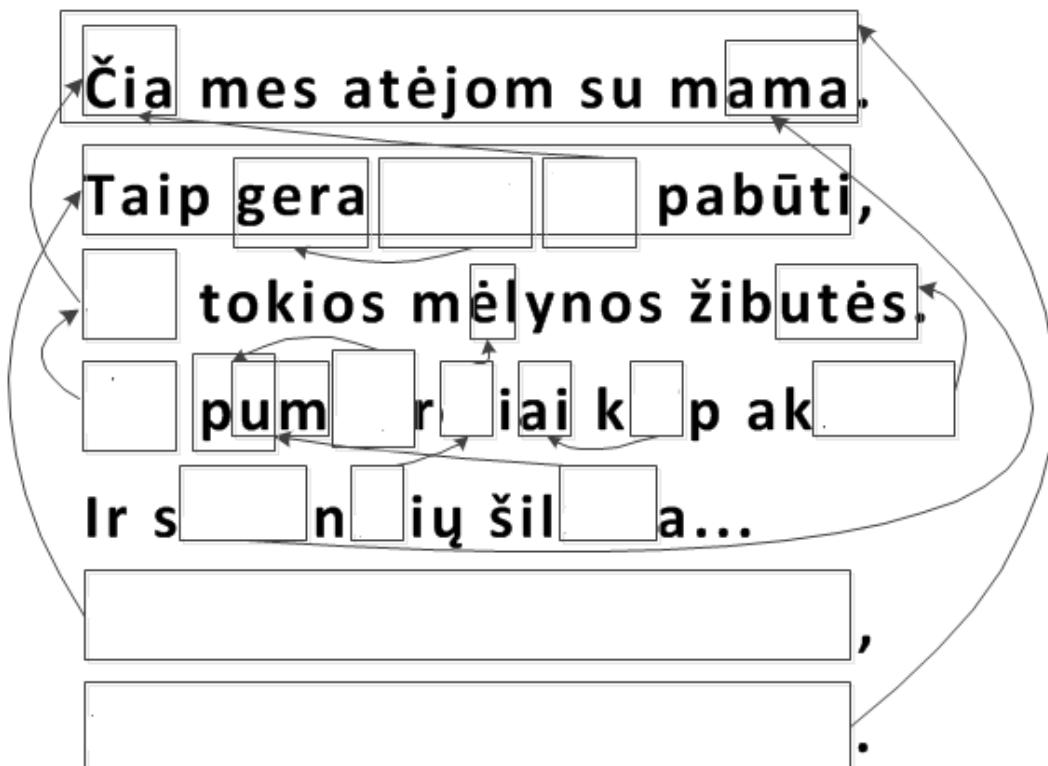


Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

## Darbo lapas Pakartok tai dar kartą!

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



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

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

**Užuominė:** stenkiteis neapkrauti teksto rodyklėmis. Rašydami eileraštį palikite daugiau laisvos vietos aplink žodžius ir tarp raidžių, kad galėtumėte piešti stačiakampius ir rodykles.

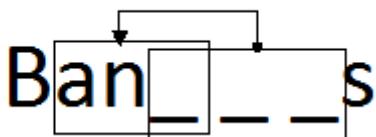
Suspaustą tekštą lengviau užkoduoti, kai pasirašote atskirai sau visą eileraštį ir tada nupiešiate kvadratelius ir rodykles.



## Darbo lapas *Gudručiams*

---

Kaip išspręstumėte šį galvosūkį?



Kartais trūkstamos teksto dalys rodo pačios į save. Šiuo atveju žodis gali būti iškoduotas teisingai, jeigu kopijuojame raides iš kairės į dešinę. Tada kiekviena raidė gali būti kopijuojama prieš tai, kai jrašoma. 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 vienos pradžią

**Pas—**

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

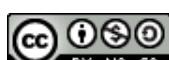
**Pasa—**

**Pasas**



Kadangi žodžiui užkoduoti naudojami du skaičiai, verta suspausti dviejų ar daugiau raidžių grupeles, kitaip nebus sutauroma vietas. 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 iškoduoti?



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Darbo lapas *Trumpas ir saldus*

---

**Kiek žodžių čia reikia iš tiesų?**

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

**Apvali, saldi gražuolė...**

**Šmurkšt, po suolu jau nupuolė.**

**„Kurgi bėgi? Grjžk į puodą...“**

**Bet, pakėlus savo kuodą,**

**Bulvė lekia per laukus:**

**„Nebegrįšiu pas vaikus.“**



# Darbo lapas *Tikriems gudručiams*

---

## Ar pasiruošę *tikrai „kietai“ suspausti?*

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

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

*Tuoj pat po jkurtuvių vakarėlio pirmasis paršiukas suposi kėdėje ir skaitė knygą. Staiga kažkas pabeldė j duris. Tai buvo didelis piktas vilkas.*

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

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

– Tada aš supyksiu, giliai jkvépsi ir nupūsiu tavo namą! – užriaumojos vilkas ir, labai supykęs, giliai jkvépę, stipriai papūtė ir suniokojo namą. Išsigandęs pirmasis paršiukas kiek jkabindamas nubėgo pas savo brolį j iš šakų pastatytą namą. Vos tik pasijuto saugus, vilkas vėl buvo prie antrojo paršiuko slenksčio.

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

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

– Tada aš supyksiu, giliai jkvépsi ir nupūsiu tavo namą! – užriaumojos vilkas ir, labai supykęs, giliai jkvépę, stipriai papūtė... Abu išsigandę paršiukai nubėgo pas trečiąjį brolį j plytų namą, bet vilkas lipo jiems ant kulnų ir tuo pat jau buvo prie durų slenksčio.

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

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

– Tada aš supyksiu, giliai jkvépsi ir nupūsiu tavo namą! – užriaumojos vilkas ir, labai supykęs, giliai jkvépę, 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 ir tik tada pamatė, kad namas – be kokino, nes trečiasis paršiukas rūpinosi aplinka ir jrengė elektrinį šildymą. Labai jniršęs vilkas paslydo ir nukrito nuo stogo. Jis susilaužė kairę koją, bet dar blogiau, kad prarado savo orumą. Paršiukai juokėsi, kai vilkas nušlubavo atgal. Jie suprato, kaip protinė gyventi mieste, kur vieninteliai vilkai yra zoologijos sode. Paršiukai persikraustė į miestą ir ten gyveno ilgai ir laimingai.“



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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 laikoma knygų ar net ištisa biblioteka, muzikos įrašų ir filmų, jei tik yra laisvos vietos. Didelės apimties dokumentai internete yra labai didelė problema, nes ilgai užtrunka juos atsisuisti. Dabar stengiamasi sumažinti kompiuterius – net telefonai ir rankiniai laikrodžiai gali laikyti labai daug informacijos.

Vienas šios problemos sprendimų štai tokis: užuot pirkę kompiuterius, turinčius daugiau atminties, ar įsigiję spartesnį interneto ryšį, galime duomenis taip suspausti, kad jie užimtų mažiau vietos. Dažniausiai duomenys koduojami ir iškoduojami kompiuteriu automatiškai. Mes to proceso beveik nepastebime. Nors iš tiesų kompiuteris atlieka daugiau veiksmų, matome tik tai, kad diske telpa daugiau informacijos ar sparčiau atsiunčiamas tinklalapis.

Sugalvota daug duomenų suspaudimo metodų. Atliekant šią veiklą, buvo naudojamas metodu, kai nuorodomis rodoma į prieš tai buvusius pasikartojančius simbolius tekste. Šis metodas vadinamas tiesiog „zip“, ji 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 modeamuose, juo suspaudžiami GIF ir PNG paveikslai, sumažinamas telefono linija siunčiamų duomenų kiekis, tad jų siuntimas kur kas spartesnis.



## **Sprendimai ir užuominos**

---

Pakartok tai dar kartą<sup>3</sup>!

**Čia mes atėjom su mama.**

**Taip gera gera čia pabūti,**

**Čia tokios mėlynos žibutės.**

**Čia pumpurėliai kaip akutės**

**Ir samanelių šiluma...**

**Taip gera gera čia pabūti,**

**Čia mes atėjom su mama.**

---

<sup>3</sup>Parenkant eiles talkino Vilniaus universiteto studentė Indrė Tamošauskaitė.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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, nežinomas 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 nuspalvinta
- ✓ Stalas (arba metalinė lenta) demonstracijai

Kiekvienai mokinui porai reikia:

- ✓ 36 vienodų kortelių, kurių viena pusė nuspalvinta

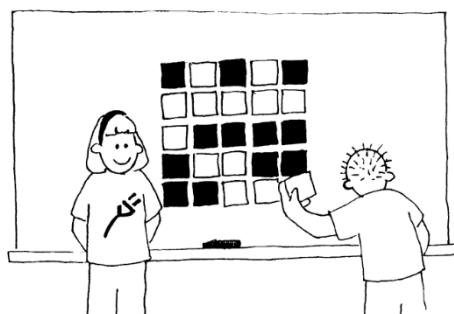


# Magiškas triukas

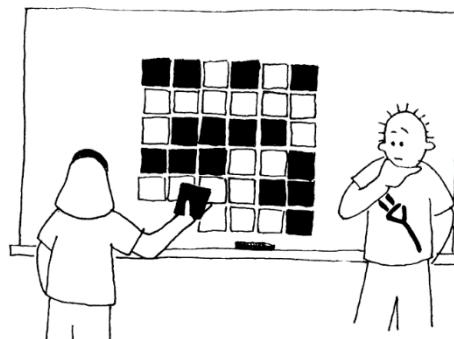
## Demonstracija

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 patogu 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 kortelės galima suklijuoti puse, kurioje nėra magneto, tada ant vienos iš „magnetinių“ naujos kortelės pusią nupiešti baltą tašką ar tą pusę nuspavinti.)

1. Mokinio paprašoma išdėlioti kortelės kvadratu  $5 \times 5$  atsitiktinė puse.



Mokytojas išdėlioja dar vieną eilutę ir stulpelį, kad šiek tiek pasunkintų užduotį.



Papildomos kortelės ir yra šio uždavinio – triuko pagrindas. Papildomų kortelių reikia pridėti tiek, kad kiekvienoje eilutėje ir stulpelyje būtų lyginis nuspavintų kortelių skaičius.

2. Tegul mokinys vieną kortelę apverčia, kai mokytojas nežiūri. Apverstos kortelės eilutėje ir stulpelyje bus nelyginis nuspavintų 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 kortelės ant stalo kvadratu  $5 \times 5$ .
2. Po kiek nuspavintų kortelių yra kiekvienoje eilutėje ir stulpelyje? Ar jų skaičius yra lyginis, ar nelyginis? Primenama, kad nulis yra lyginis skaičius.

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

**Papildomos veiklos:**

1. Šią 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 lyginimo kortelės neparodys klaidos.)
3. Galima išbandyti triuką su daugiau kortelių, pavyzdžiui, išdėstyti iš jų didelį kvadratą  $9 \times 9$ . (Šis triukas gali būti daromas su bet kiek kortelių, be to, jų nebūtina išdėstyti kvadratu.)
4. Kitas jdomus pratimas – nagrinėti apatinę dešiniają 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 nuspalvintų kortelių skaičius eilutėse ir stulpeliuose. Ar galima tą patį padaryti su lyginiu lyginumu? (Tai įmanoma. Tačiau apatinės dešiniosios lyginumo kortelės pratimas tinka tik tada, kai yra lyginis (arba nelyginis) ir eilučių, ir stulpelių skaičius išdėstytyame plote. Pavyzdžiui, korteles išdėsčius  $5 \times 9$  arba  $4 \times 6$  pratimą galima atlikti, o išdėsčius  $4 \times 7$  pratimas netinka.)



## **Realaus gyvenimo pavyzdys gudručiams!**

---

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

Vadinasi, jei knygai suteiktas *tarptautinis standartinis knygos numeris ISBN* (angl. *International Standard Book Number*), 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

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

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, 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.



Traškios duonos riekelių „Joré“ déžutės 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 kitur, kur žmonėms reikia įsitikinti, kad naudojamas numeris yra teisingas.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# Patikrink šią knygą!

Detektyvai

Knygų tikrinimo paslaugos



Randame ir patikriname ISBN kontrolines sumas už nedidelį mokesčių.

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 numerj;
- ✗ skaitmuo praleistas numeryje.

Ar galėtumėte surasti knygą su X kontrolinei sumai 10? Neturėtų būti labai sunku, nes kiekviena 11 knyga ji 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)?



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

## Apie ką visa tai?

---

Tarkime, reikia jnešti 10 € į banko sąskaitą. Banko kasininkas jveda sumą ir siunčia duomenis į centrinj kompiuterj. Tačiau atsiranda trikdžių ir į centrinj kompiuterj 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 siunčiant nebuvu sugadinti kokių elektros trikdžių. Kartais klaudingai perduodamų duomenų siuntimas pakartojamas, tačiau kai kada tai neįmanoma, pavyzdžiu, 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 kaida. (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 lyginimo bitai. Tokiu būdu galima ne tik nustatyti klaidą, bet ir pasakyti, kur ji yra. Pažeista bitas atkuriamas. Kaida ištaisoma.

Kompiuteriuose naudojamos daug sudėtingesnės klaidų kontroliavimo sistemos, kurios gali nustatyti ir ištaisyti sudėtines 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 lyginumu.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## **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 dažniausiuų klaidų (vienas neteisingai surinktas skaitmuo arba vietomis sukeisti du skaitmenys) dažniausiai aptinkama.



## 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. Tolkino (J. R. R. Tolkien) knygoje „Žiedų valdovas“? Jei galima išmatuoti informaciją, galima ir įvertinti, kiek vienos reikia šiai informacijai laikyti. Pavyzdžiui, ar galite perskaityti toliau esantį sakinį?

Šs skns ntr bIls.

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

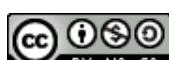
- ✓ Nuo 10 metų

#### Priemonės

- ✓ Pirmai veiklai atliliki papildomos priemonės nereikalingos

Papildomai veiklai atliliki kiekvienam mokinui reikia:

- ✓ Darbo lapo „Sprendimų medžiai“



Licensed under Creative Commons

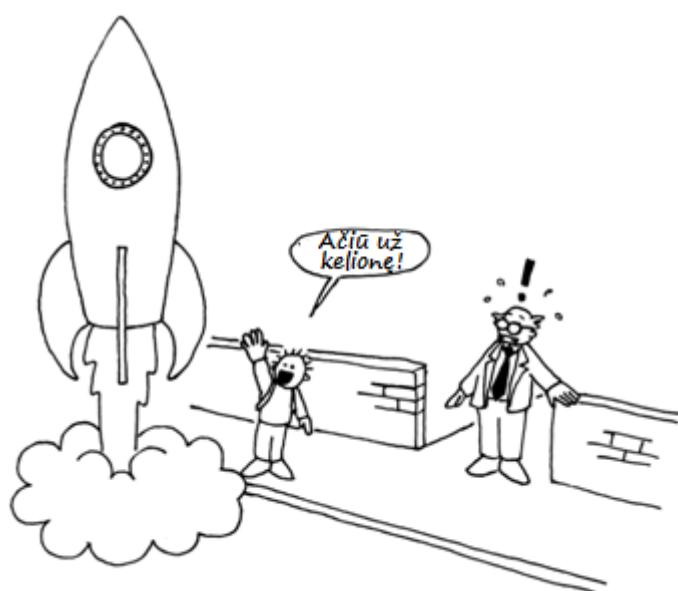
© 2015 Computer Science Unplugged (csunplugged.org)

## Dvidešimt spējimų

---

### 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 jdomi? 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 pasakytu: „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 pasakytu: „Atspék, kaip aš šiandien atvykau į mokyklą“, tai pirmasis spējimas tikriausiai būtų, kad jis atėjo pėsčiomis. Tačiau reikėtų dar kelių spējimų, jei jis atskrido sraigtasparniu, ar dar daugiau spējimų, jei atskrido erdvėlaiviu.
6. Informacijos kiekis, kurį turi žinia, priklauso nuo to, kaip lengva ar sunku ją atspėti. Toliau aprašytas žaidimas padės tai išsiaiškinti.



## Dvidešimt klausimų

---

Tikriausiai dauguma žino 20 klausimų žaidimą: pasirinktam mokinui kiti mokiniai užduoda klausimus, j 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 taisykłę (pvz., lyginiai skaičiai, skaičių kvadratai ir pan.). Sekos nariai spēliojami iš eilės nuo pirmojo iki paskutinio (pvz., 2, 4, 6, 8, 10).

Suskaičiuojami užduoti klausimai. Gautas skaičius yra informacijos matavimo vertė.

### Diskusija

Kokias būdais buvo spėjamas atsakymas? Kuris būdas buvo geriausias?

Atkreipiama 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 tiketina žodyje arba sakinyje.

Siūloma pabandyti spėjimo žaidimą paėmus trumpą 4–6 žodžių sakinį. Raidė turi būti spėjama iš eilės nuo pirmos iki paskutinės pagal abécélę. Vienas mokinys galėtų surašyti raides pagal abécélę. 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 abécélėje?“ Tarpas tarp žodžių taip pat laikomas raide ir turi būti spėjamas. Perklausomas įrašas ir nustatoma, kurių žinios dalį buvo lengviau atspėti.



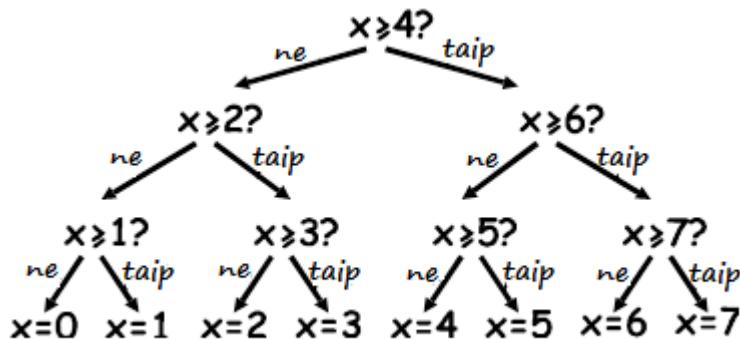
Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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 pavyzdyste po skaičiais 0, 1, 2, 3, ... paskutinėje eilėje užrašykite jų reikšmes dvejetainė sistemoje (žr. 1 veiklą).

Atidžiau pažvelkite 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 spėjamų skaičių dvejetainė sistema.

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

**Gudručiams.** Kokį medį naudotumėte spėdami kieno nors amžių?

Kokiui medžiu pavaizduotumėte sakinio raidžių spėjimo medį?



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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 kiekj 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?“ I 80 proc. tokį klausimą, jei metama netaisyklinga moneta, atsakymas būtų „taip“ ir tik 20 proc. klausimų atsakymas būtų „ne“. Vidutiniškai užduodamas mažiausiai vienas klausimas apie vieną monetos metimą.

K. Šenonas jvedė 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 suaupo 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 sutiki arba atmesti siūlymus. Kompiuteriui užtenka tik vieno atsakymo „taip“ arba „ne“ kiekvienam ženklui, bet 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 liečiamuose telefonuose.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## **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 seką yra spėjamo skaičiaus reikšmė dvejetainėje sistemoje. Trys dvejetainėje sistemoje yra 110, atsakymų apie šį skaičių seką 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.





Licensed under Creative Commons  
© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# 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 tokį algoritmų, 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 atliliki siekiant užsibrėžto tikslo arba išspręsti suformuluotą uždavinį. Informatikoje algoritmo savoka 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  $\pi$  (pi) milijoną skaitmenų po kablelio, visus tinklalapius, kuriuose minimas tam tikras žodis, talpiausią déjimo į konteinerį būdą, didžiausią pirminį skaičių ir kt.

Algoritmo savoka 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.



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# 6 veikla

---

## Laivų mūšis. Paieškos algoritmai

### Santrauka

Dažnai kompiuteris turi surasti tam tikrą informaciją iš didelio jos kieko. Tuo tikslu turi būti sukurtas greitas ir efektyvus būdas. Čia pateikiami trys skirtini 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 reikia:

- ✓ 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

---

## Ivadas

1. Pasirenkama 15 mokiniai, kurie sustoja viena eile prieš klasę. Kiekvienam mokinui duodama kortelė su skaičiumi (atsitiktine tvarka). Skaičiai nerodomi likusiems mokiniams. Pasakoma, jų 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 jų 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 mokinį, pasirinkę atversti vidurinio mokinio kortelę. Kartodami šį veiksmą jie gali išsiaškinti ieškomą skaičių „sumokėję“ tris saldainius. Šio metodo efektyvumas akivaizdus.

## Veikla

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# **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 pasakojo 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 žaidimo 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. Dvejetainė 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 pasakojo 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 žaidimo 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 veikiama šiuo metodu, kiek prireikė šūvių pataikyti į laivą? (Daugiausia 5 šūviai.)

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# Laivų mūšis. Maiša

---

## Instrukcija

1. Mokiniai susiskirsto poromis. Vienas paima 3A lapą, kitas – 3B. Abu vienas kitam pasako savo pasirinkto laivo numerj.
2. Žaidžiant šį žaidimą reikia nustatyti, kuriame iš dešimties stulpelių (nuo 0 iki 9) yra laivas. Tuo tikslu sudedami laivo numero 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 abiejuose lapuose laivai turi būti išdėstyti stulpeliuose vienodai.)

## Diskusija

1. Kokie žaidimo 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 išrikuoti 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 antrajį 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 nutiktu, jei ieškomo laivo lape nebūtų? (Atlikdami nuosekliajā paiešką tuo įsitikintume padarę 26 žingsnius. Jei paieška dvejetainė, reikėtų tik 5 žingsnių. Jei naudojamasi 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 logaritmu.)



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Mano laivai

Šūvių skaičius:									
A	B	C	D	E	F	G	H	I	J
N	O	P	Q	R	S	T	U	V	W
9058	7169	3214	5891	4917	2767	4715	674	8088	1790
8311	7621	3542	9264	450	8562	4191	4932	9462	8423

## Priešininko laivai

Šūvių skaičius:									
A	B	C	D	E	F	G	H	I	J
N	O	P	Q	R	S	T	U	V	W
9058	7169	3214	5891	4917	2767	4715	674	8088	1790
8311	7621	3542	9264	450	8562	4191	4932	9462	8423

1A



## Mano laivai

Šūvių skaičius:

A	B	C	D	E	F	G	H	I	J	K	L	M
1630	9263	4127	405	4429	7113	3176	4015	7976	88	3465	1571	8625
2587	7187	5258	8020	1919	141	4414	3056	9118	7117	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
1630	9263	4127	405	4429	7113	3176	4015	7976	88	3465	1571	8625
2587	7187	5258	8020	1919	141	4414	3056	9118	7117	7021	3076	3336
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1B



Mano laivai

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
-163	-445	-622	-1410	-1704	-2169	-2680	-2713	-2734	-3972	-4208	-4871	-5031
A	B	C	D	E	F	G	H	I	J	K	L	M
-5283	-5704	-6025	-6801	-7440	-7542	-7956	-8094	-8672	-9137	-9224	-9508	-9663

### Šūvių skaičius:

Priešininko laivai

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

Šūvių skaičius:

2A



## Mano laivai

### Šūvių skaičius:

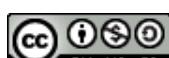
-33	-183	-730	-911	-1927	-1943	-2200	-2215	-3451	-3519	-4055	-5548	-5655
A	B	C	D	E	F	G	H	I	J	K	L	M
5785	5897	5905	6118	6296	6625	6771	6831	7151	7806	8077	9024	9328
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

## Priešininko laivai

### Šūvių skaičius:

-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

2B



# 3A

## Mano laivai

0	1	2	3	4	5	6	7	8	9
A  9047	C  3080	E  5125	H  8051	I  1481	L  7116	O  6000	R  9891	S  1989	W  1062
B  1829	D  9994	F  1480	G  8212	J  4712	M  8944	P  7432	T  2050	U  8199	X  2106

## Šūvių skaičius:

E  5125	F  1480	G  8212	H  8051	I  1481	J  4712	K  6422	L  7116	M  8944	N  4128
V  4392	W  1062	X  2106	Y  5842	Z  7057					

## Priešininko laivai

0	1	2	3	4	5	6	7	8	9
A	E	H	K	L	M	O	R	S	W
B	F	I	J			P	V	T	X

## Šūvių skaičius:

V	W	X	Y	Z



# 3B

## Mano laivai

0 A 9308	1 E 6519	2 H 1524	3 F 2469	4 K 4135	5 L 9050	6 O 4200	7 R 3121	8 V 2385	9 Y 1990
B 1478	C 8417	D 9434	E 5105	F 5105	G 5105	H 8112	I 2000	J 2000	K 1265
						M 5711	N 5711	O 6028	P 7153
						Q 6028	U 7019	V 5832	W 1917
								X 2502	Z 2502

## Šūvių skaičius:

## Priešininkų laivai

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

## Šūvių skaičius:

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Mano 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
-6123	-1519	-9024	-5164	-2038	-2142	-7156	-9974	-9375	-7104	-1004	-1023	-5108

## 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
-3541	-5251	-4840	-3289	-3654	-2480	-5602	-8965	-4053	-2405	-2304	-1959	-1959

1A'



1 B'

### Mano laivai

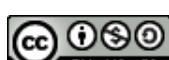
A	B	C	D	E	F	G	H	I	J	K	L	M
2387	9003	3951	5695	1284	4761	7118	1196	1741	3791	3405	3132	6682

### Šūvių skaičius:

### Priešininko laivai

A	B	C	D	E	F	G	H	I	J	K	L	M
-	-	-	-	-	-	-	-	-	-	-	-	-

### Šūvių skaičius:



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Mano laivai

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

2A'



# 2B'

## Mano 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
56 -56	194 -194	306 -306	1024 -1024	1510 -1510	1807 -1807	2500 -2500	2812 -2812	3011 -3011	3902 -3902	4178 -4178	5902 -5902	5915 -5915

## 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
-6102 -6102	-6526 -6526	-6818 -6818	-7020 -7020	-7155 -7155	-7913 -7913	-8016 -8016	-8230 -8230	-8599 -8599	-8902 -8902	-9090 -9090	-9526 -9526	-9812 -9812



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# 3A'

## Mano laivai

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

## Priešininko laivai

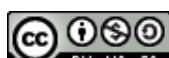
Šūvių skaičius:									
0	1	2	3	4	5	6	7	8	9
A	E	H	I	K	L	O	R	S	W
B	F	J		M		P	T	U	V
C	G			N		Q		X	Z
D									



# 3B'

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

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Apie ką visa tai?

---

Kompiuteriuose laikoma labai daug informacijos, ji turi būti greitai apdorojama. Daugiausia paieškos sunkumų kyla dėl būtinės 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 skirtinį produkty, išdėliotų skirtinio lentynose. Kai nuskaitomas prekės kodas, kompiuteris turi patikrinti 10000 kodų, kol randa reikiama prekę ir jos kainą. Jei vieną prekę kompiuteris tikrintų vieną tūkstantąjį sekundės, tai užtruktų 10 sekundžių. Kiek užtruktų surasti visas kieno nors perkamas prekes!

*Dvejetainė 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 šimtiasias sekundės, 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 svarstyklęs, kurios aprašomos 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 išrikuoti sąrašą. Pavyzdžiui, išrikuoti vardus pagal abécèle, elektroninius laiškus pagal datą ar kt. Išrikuotame 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 skirtinges rikiavimo metodus, kaip pasirinktas „gudrus“ rikiavimo metodas paspartina rikiavimą.

### Ryšiai su ugdymo programomis

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

### Gebėjimai

- ✓ Svarstyklų naudojimo
- ✓ Rikiavimo
- ✓ Lyginimo

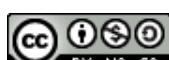
### Amžius

- ✓ Nuo 8 metų

### Priemonės

Kiekvienai mokinių grupei reikia:

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# **Lengviausias ir sunkiausias**

---

## **Diskusija**

Kompiuteriui dažnai reikia išrikiuoti duomenis. Kas nutiktu, 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 „Svorų rikiavimas“, sveriamų pakuočių rinkiniai ir svirtinės svarstyklės.
3. Mokiniai atlieka šią veiklą ir tada aptaria rezultatus.



## Darbo lapas *Svorų rikiavimas*

**Tikslas:** surasti geriausią nežinomų svorių rikiavimo metodą.

**Reikia:** smėlio arba vandens, 8 vienodų pakuocių, svirtinių svarstykliai.

**Ką daryti:**

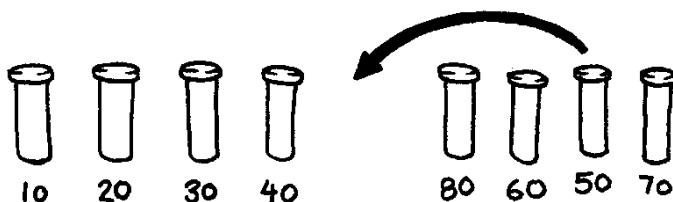
1. Į kiekvieną pakuotę įpilkite skirtingą kiekį smėlio arba vandens. Sandariai uždarykite.
2. Sumaišykite pakuotes, kad nežinotumėte, kiek juose smėlio ar vandens.
3. Suraskite lengviausią pakuotę. Kokiu paprasčiausiu būdu tai galima padaryti?

**Patarimai.** Naudokite svirtines svarstyklės. Vienu metu galite jomis palyginti tik du svorius.

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



Suskaičiuokite, kiek kartų lyginote svorius.

**Gudručiams.** Parodykite, kaip matematiškai galima suskaičiuoti, kiek kartų reikia lyginti 8 objektus norint juos išrikuoti pagal svorj. Kiek kartų norint išrikuoti reikia lyginti 9 objektus? 20 objektų?



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# Darbo lapas *Skaldyk ir valdyk!*

## Greitojo rikiavimo algoritmas

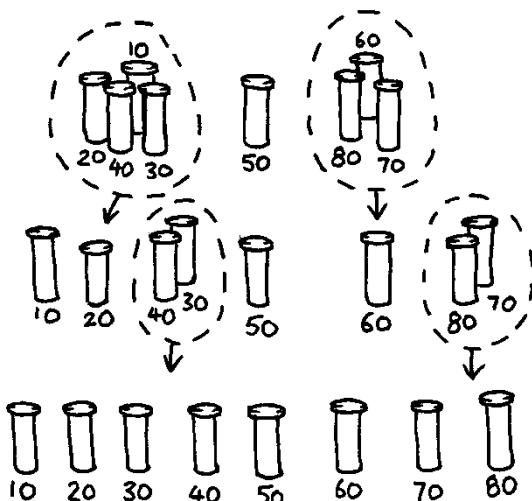
*Greitojo rikiavimo algoritmas* yra daug greitesnis ir paprastesnis už *išrenkamojo rikiavimo algoritmą*, todėl yra labai dažnai naudojamas.

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

Visus likusius objektus palyginkite su atsitiktinai pasirinktu objektu. Lengvesnius už jį padėkite kairėje svarstyklų 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.

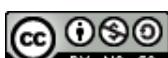
Kartokite šiuos veiksmus, kol grupėse liko tik po vieną objektą. Kadangi visose objektų grupėse liko tik po vieną objektą, vadinas, visi objektai yra išrikiuoti nuo lengviausio iki sunkiausio.



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 objekto, 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?



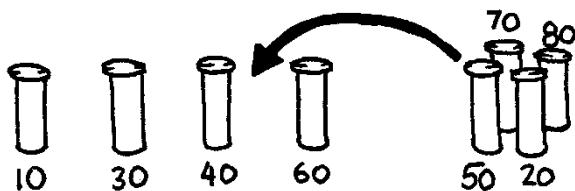
Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

## Kiti rikiavimo metodai

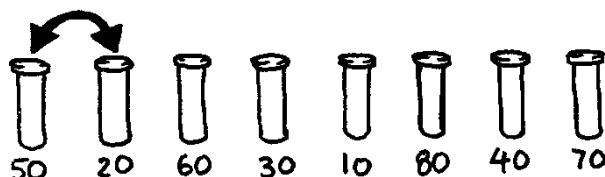
Sugalvota daug rikiavimo metodų. Pabandykite išrikuoti savo objektus įvairiais metodais.

Pagrindinis *jteriamojo rikiavimo algoritmo* principas – kiekvienas elementas imamas iš eilės ir jterpiamas į jam skirtą vietą jau surikiuotoje objektų grupėje. Kiekvieną kartą jterpus objektą, neišrikuuotoje grupėje objektų mažėja, o išrikuuotoje – daugėja.

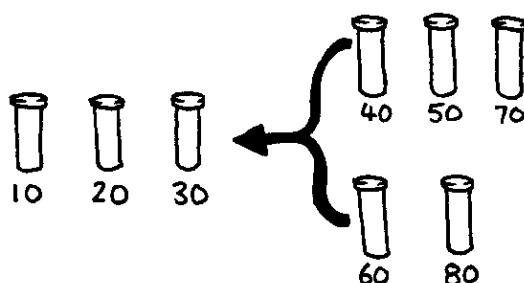


*Burbulinio rikiavimo algoritmo* principas – nuosekliai iš eilės patikrinti gretimų objektų poras, prireikus objektus sukeisti, t. y. perkelti mažesnį arčiau pradžios. Šitaip per pirmą žingsnį lengviausias objeketas perkeliamas į pirmą poziciją, vėliau tas pats principas taikomas likusiems objektams 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 objektais 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 objeketas perkeliamas į jungtinės objektų grupės pabaigą. Paveiksle 40 ir 60 gramų objektais yra kiekvienos grupės pradžioje, į jungtinę grupę dedamas mažesnio svorio objektais – 40 g ir t. t.



Kaip išrikuoti mažai objektų? Patartinas sąlajinio rikiavimo algoritmas! Visos grupės padalijamos po vieną objekta, todėl nereikia rūpintis, kada baigt išrikuoti.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Apie ką visa tai?

---

Išrikuotame sąraše daug lengviau surasti reikiama informaciją. Telefonų knygoje, knygų kartotekoje elementai rikiuojami pagal abécélę. Jei jie nebūtų išrikuoti pagal abécélę, būtų daug sunkiau susigaudyti informacijos gausybėje ir rasti reikiama pavardę telefonų knygoje ar autorų kartotekoje. Išrikuotame skaičių sąraše lengva pamatyti didžiausią, mažiausią ar vienodus elementus.

Dažnai rikiuojant pasitelkiami kompiuteriai, todėl informatikai ieško vis greitesnio rikiavimo algoritmo. Dauguma rikiavimo algoritmų, pavyzdžiui, *išrenkamojo*, *jterpiamojo ar burbulinio rikiavimo*, yra naudingi sprendžiant tam tikras problemas. Algoritmai, tokie kaip *greitojo* ar *sqlajinio rikiavimo*, yra daug efektyvesni ir pritaikomi didesniams objektams skaičiui. Pavyzdžiui, 100000 objektų *greitojo rikiavimo algoritmas* išrikuoja 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. 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 save patį, kol išsprendžia problemą.



## Sprendimai ir užuominos

---

1. Geriausias būdas surasti lengviausią objektą yra tikrinti visus objektus lyginant vieną su kitu. Kitaip tariant, objektais lyginami po du, iš kiekvienos poros ant svarstyklę lėkštelių paliekamas lengvesnis objektas tol, kol perrenkami ir palyginami visi objektais.
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 palyginti

$$7 + 6 + 5 + 4 + 3 + 2 + 1 = 28 \text{ kartus.}$$

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

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

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

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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 spendimo bendradarbiaujant

#### Amžius

- ✓ Nuo 7 metų

#### Priemonės

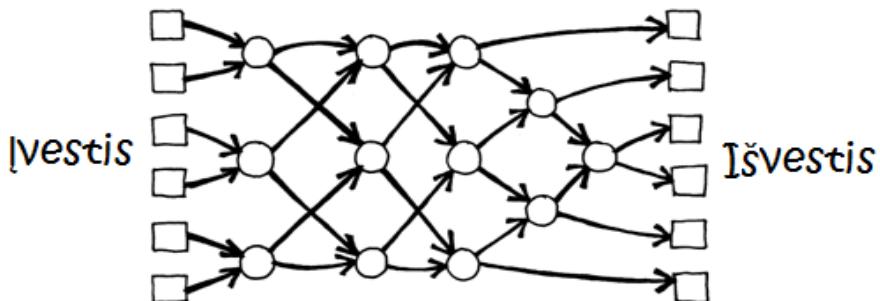
Lauko žaidimas.

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



# Tinklinis rikiavimas

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



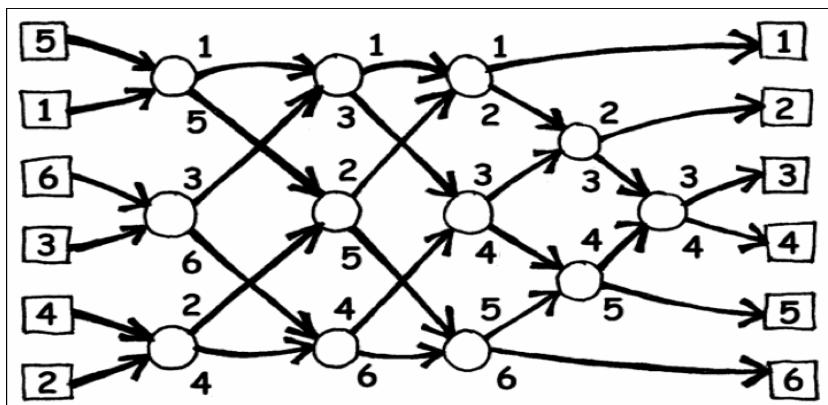
## Instrukcijos

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

1. Mokiniai susiskirsto į grupes po šešis. Vienu metu tik viena grupė naudojasi 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 neturėtų eiti iš eilės.
4. Mokinys juda nurodyta kryptimi. Atsistojęs ant apskritimo **palaukia kitos 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 – dešiniosios rodyklės kryptimi.
6. Mokinys teisingai baigia žaidimą, kai pasiekia kvadratą tinklo dešinėje (ten, kur pažymėta išvestis).

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

Rikiavimo pavyzdys parodytas toliau paveiksle.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Tinklinis rikiavimas

---

1

2

3

4

5

6

---

156

221

289

314

422

499

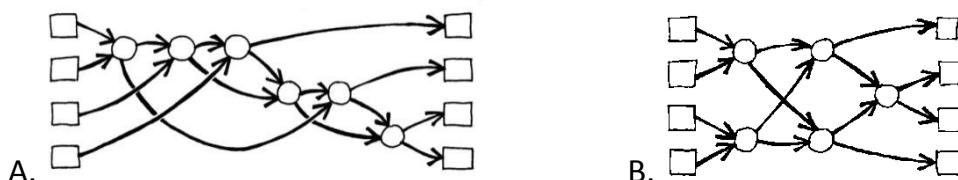
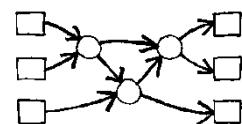


## Variantai

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

## Gudručiams

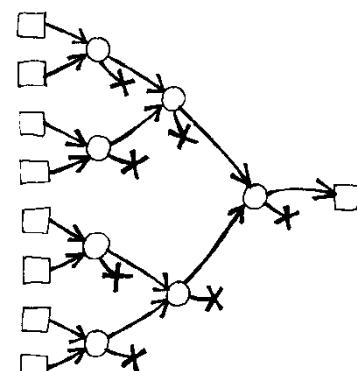
4. Kas atsitiktų, jei visi mokiniai, turintys kortelių su mažesniais (didesniais) skaičiais, pasuktų ne į dešinę (kairę), o į kairę (dešinę)? (Skaičiai būtų išrikiuoti atvirkščiai.)
5. Ar visada skaičiai išrikiuojami teisingai pasiekus išvestį, jei rikiavimo kryptys pakeiciamos priešingomis? (Ne visada. Mokiniai turėtų pateikti pavyzdį, kada išrikiuojama neteisingai.)
6. Mokiniam pasiūloma sukurti mažesnį ar didesnį tinklą. Pavyzdžiui, dešinėje, paveiksle, vaizduojamas tinklas, kuriuo išrikiuojami tik trys skaičiai. Mokiniai turėtų patys sukurti tokį ar panašų tinklą.
7. Toliau 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 – lygiagretasis.)



8. Mokiniam pasiūloma sukurti didesnį tinklą.

Naudojant tinklą galima surasti mažiausią arba didžiausią skaičių. Pavyzdžiui, toliau paveiksle išrenkamas mažiausias skaičius iš aštuonių. Atliekant šį algoritmą išvedamas tik mažiausias skaičius, o kiti lieka tinklo aklavietėse.

9. 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 pasamžius 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 žemų iš duobės negalima kasti, kol neiškastas metras žemų virš jo. Informatikai aktyviai darbuojasi ieškodami geriausio problemas padalijimo metodo, kad būtų galima pritaikyti lygiagretumo metodą.



## 9 veikla

---

# Purvinas miestas. Minimalūs jungiantieji medžiai

### Santrauka

Realiame gyvenime esama daug įvairių tinklų – tai telefonų tinklai, komunalinių paslaugų tinklai, kompiuterių tinklai, kelių tinklai ir kt. Kuriant paslaugų tinklus svarbu nustatyti, kur optimaliai tiesi 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 mokinui reikia:

- ✓ Darbo lapo „Purvinas miestas“



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# Purvinas miestas

---

## Ivadas

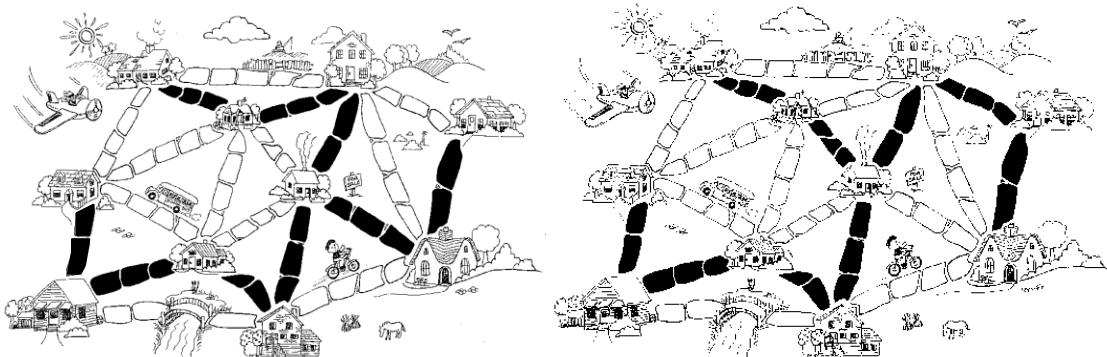
Ši veikla skiriama susipažinti, kaip kompiuteriai padeda surasti geriausią realios problemos – kaip geriausia nutiesti kelią – sprendimą. Mokiniams išdalijami darbo lapai, kuriuose paaiškinta *Purvino miesto* problema.

## Diskusija

Aptariama, kokias 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 nenutiesta. Kitas uždavinio sprendimo būdas, kai miesto plane yra sužymėtos visos gatvės, – šalinti po vieną nereikalingą gatvę ir tikrinti, ar j visus namus galima patekti likusiomis gatvėmis.

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



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

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

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



## Darbo lapas *Purvinas miestas*

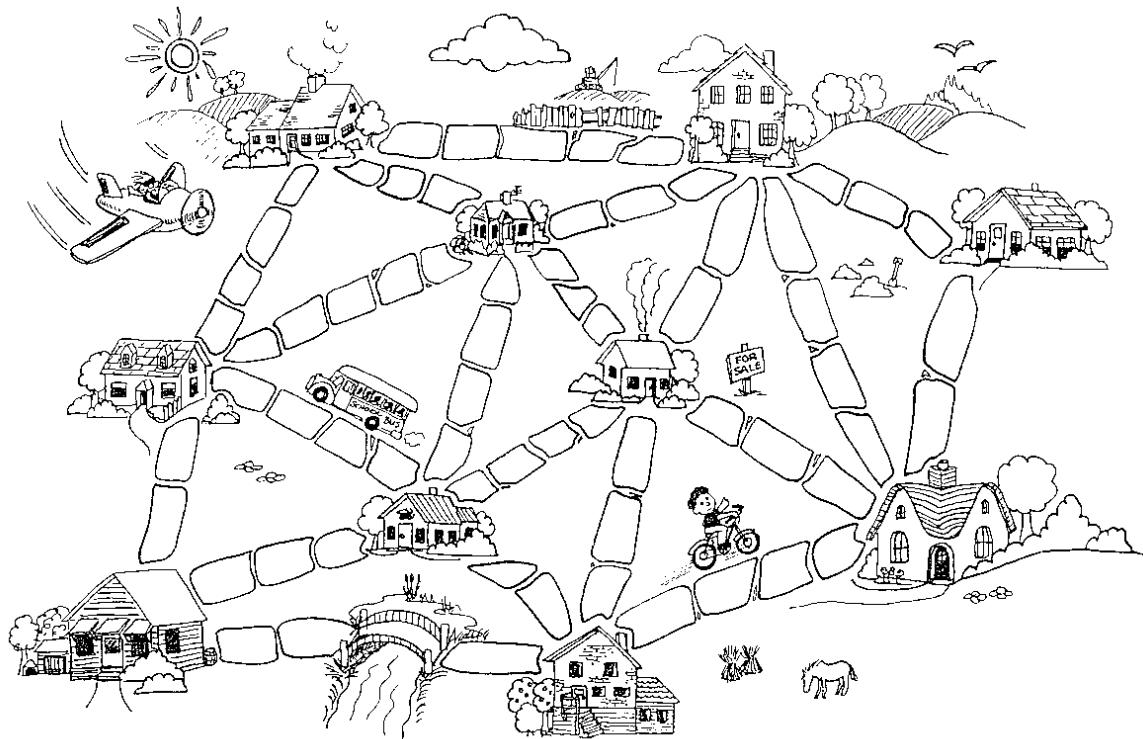
---

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

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

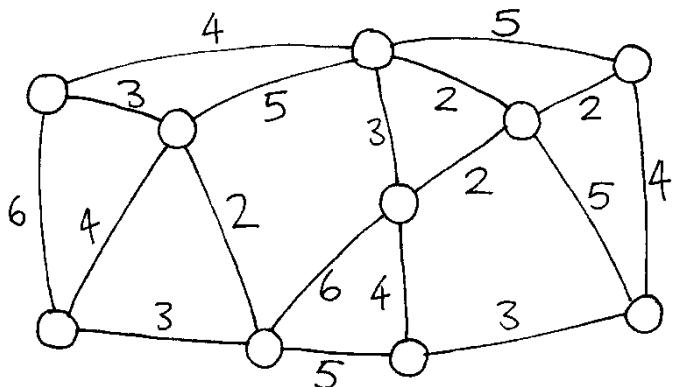
Pateiktas miesto planas. Akmenų skaičius tarp kiekvieno namo proporcingas gatvės grindimo kainai. Suraskite geriausią gatvių grindimo variantą, kuris sujungtų visus namus ir būtų pigiausias.

Pagalvokite, kokiui 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 ilgiai nurodyti skaičiais ties linijomis.



Informatikoje ir matematikoje uždaviniamas spręsti dažnai naudojami *grafai*. Pirmą kartą išgirdus apie grafo sąvoką galima pagalvoti, kad tai panašu į grafiką, vartojamą statistikoje duomenims vaizduoti. Nieko panašaus, grafai informatikoje yra kas kita.

*Grafuose* linijų ilgis neturi reikšmės – jokio mastelio nereikia laikytis.

Mokiniai galėtų sukurti savo miestą ir draugų paprašyti sujungti vaizduojamus namus gatvėmis. Kiek mažiausiai gatvių reikia 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 reikiama 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 algoritmų. Galima pradėti nuo miesto plano, kuriame nesužymėtos 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 išspresti sukurta ir tebekuriama gana efektyvių metodų.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Sprendimai ir užuominos

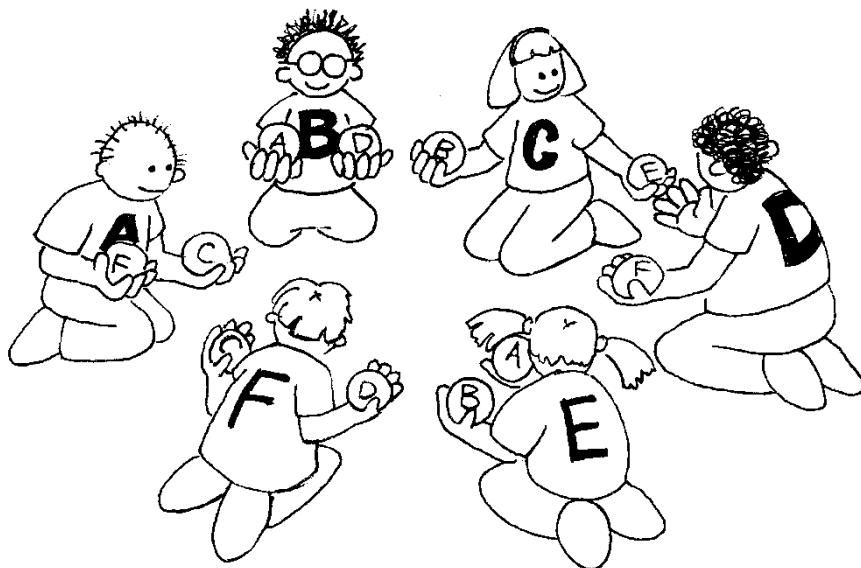
### **Variantai gudručiams**

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 mokinui reikia:

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# Apelsinų žaidimas

## Ivadas

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

1. Penki ar daugiau mokinų susėda ratu.
2. Kiekvienam mokinui 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 raide.
4. Apelsinai siunčiami ratu, kol kiekvienam mokinui lieka po vieną apelsiną su jo raide. Apelsinai siunčiami pagal dvi taisykles:
  - a) Rankoje vienu metu gali būti tik vienas apelsinas.
  - b) Apelsinas gali būti perduotas tik šalia esančiam mokinui, 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 neatidavę, grupė negali pasiekti žaidimo tikslų. Galbūt reikėtų pabrėžti, kad mokiniai tada nelaimi žaidimo, o žaidimas baigiamas, kai kiekvienas rankose laiko ne savo apelsinus.

## Diskusija

Kokias būdais mokiniai sprendžia problemą?

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

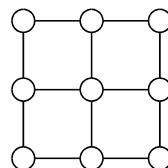
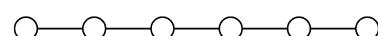
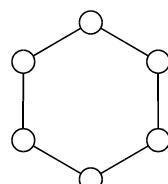
## Gudručiams

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

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, kaip parodyta paveiksluose.



## 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, bandydamis suprojektuoti tinklus, kad šios problemos sprendimas būtų paprastesnis.

Maršruto parinkimas, spūstis, grūstis ir aklavietė – tai jvairių tinklų sudarymo nesėkmų 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 galingą kompiuterį. Tokius žaidimus, kaip „Apelsinų“ žaidimas, su šiuo kompiuteriu galima žaisti be perstojo ir daug greičiau.



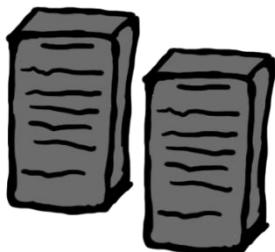
Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# 11 veikla

---

## Akmeninės plokštelės. Tinklo komunikacijų protokolas



### Santrauka

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

### Ryšiai su ugdymo programomis

- ✓ Matematika: loginės problemas
- ✓ Lietuvių kalba: komunikavimas, kito asmens klausymas

### Gebėjimai

- ✓ Problemos sprendimo bendradarbiaujant
- ✓ Loginio pagrindimo

### Amžius

- ✓ Nuo 9 metų

### Priemonės

Kiekvienam mokiniui reikia:

- ✓ Daug tuščių plokštelių (apie 30)

Kiekvienam pasiuntiniui reikia:

- ✓ Žinučių kortelių rinkinio

Mokytojui reikia:

- ✓ Laikmačio



# Akmeninės plokštelės

---

## Ivadas

Ši veikla skiriama paaškinti, kaip veikia skirtini komunikavimo metodai. Pateikiamas 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 plokštėlėje yra vietas 6 ženklams (vienas žinutės ženklas rašomas į vieną langelį), todėl viena žinutė netelpa į vieną plokštélę. Žinutei perduoti reikės apie 30 plokštelių kiekvienam mokinui.

**Patarimai.** Kortelės žymi tris veiksmus: atidėti, pristatyti ir nepristatyti. Šių veiksmų derinimas nusako perdavimo tinklu 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 nematyti vienas kito ir negalėtų kalbėtis. Jie gali būti skirtinguose kambariuose ar tame pačiame kambarysteje, bet nusisukę vienas nuo kito.
2. Vienam poros nariui duodama žinutę, kurią jis turi perduoti savo porininkui per pasiuntinį.
3. Veiksmų kortelės sumaišomos, pasirenkamas pasiuntinys. Juo gali būti mokytojas ar mokinys, jei mokinį skaičius nelyginis ir vienas mokinys lieka be poros. Jei klasė didelė, galima pasirinkti du pasiuntinius.
4. Mokinui, turinčiam žinutę, duodamos plokštelės. Pirmoje plokštélėje jis parašo dalį žinutės. Virš plokštelių užrašo mokinio, kuriam siunčia žinutę, vardą ir perduoda plokštélę pasiuntiniui.
5. Pasiuntinys, paėmęs plokštélę, traukia veiksmų kortelę, ją apverčia ir perskaito, ką turi padaryti su šia plokšteliu.
6. Kartojami 4 ir 5 žingsniai su kiekviena plokšteliu.

Po 5 ar daugiau chaoso ir nesėkmų minučių mokiniai turėtų suprasti, kad protokolui tik vardo neužtenka. Sustabdomas žaidimas ir pradedama diskusija.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

Su kokia problema mokiniai susidūrė pirmiausia? Ar tai plokštelių eiliškumas? Tikriausiai būtų protinga j vieną iš 6 ženklams skirtų vietų išrašyti siunčiamos žinutės plokšteliės eilės numerj? Tai reikštų, kad žinutės tekstui lieka mažiau vienos. Kaip tai lemia plokštelių skaičiu?

Po kurio laiko mokiniai turėtų pastebėti kitas problemas. Žaidimas sustabdomas ir padiskutuojama. Galima pastebėta problema – dingusios plokšteliės, nežinoma, ar plokšteliė buvo neišsiusta, ar reikia siusti iš naujo. Tikėtina, kad mokiniai pasiūlys sprendimą: gavęs plokštelię porininkas turi išsiusti patvirtinimą ir tik gavus patvirtinimą išsiunčiama kita žinutės dalis. Tai reiškia, kad porininkas taip pat turi turėti tuščių plokštelių. Prieš tēsdami žaidimą mokiniai turi susitarti, kam bus naudojamos 6 ženklų vietas jų plokšteliėse.

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



<b>Kam:</b>     	<b>Kam:</b>     
<b>Nuo ko:</b>     	<b>Nuo ko:</b>     
<b>Kam:</b>     	<b>Kam:</b>     
<b>Nuo ko:</b>     	<b>Nuo ko:</b>     
<b>Kam:</b>     	<b>Kam:</b>     
<b>Nuo ko:</b>     	<b>Nuo ko:</b>     
<b>Kam:</b>     	<b>Kam:</b>     
<b>Nuo ko:</b>     	<b>Nuo ko:</b>     



## 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 siusti vienas kitam žinutes per visą miestą. Pasiuntiniai valdytojus skirdavo pagal jų namų numerius.

Vienintelis būdas nusiųsti žinutę – užrašyti ją ant stačiakampės akmeninės 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 plokštelii, bet jos būdavo labai sunkios ir vienu metu pasiuntinys galėdavo nunešti tik vieną plokštelę.

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

Miesto valdytojai norėjo rasti patikimesnį bendravimo būdą. Jie sumanė sukurti taisykles, kurios padėtų sužinoti, ar išsiųsta žinutę 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.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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 duomenų, 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 plokšteliems dalj 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 tokį paketą.

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

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





Licensed under Creative Commons  
© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# 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 vykdys 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 tikrujų komandas jis supranta pažodžiu, todėl, pavyzdžiu, jei kompiuteris yra įmontuotas į judantį robotą, norint išvengti nesėkmės reikia imtis atsargumo ir saugumo priemonių. Jei roboto kompiuteriui liepama eiti pro duris, tai robotas ir eina kiaurai pro duris, nekreipdamas dėmesio, ar jos uždarytos ar atidarytos.

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

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

Atliekant antrają veiklą sužinoma, kaip susikalbēti su kompiuteriu. Programuotojas turi išmokti nurodyti kompiuteriui, ką šis turi daryti, aiškiomis komandomis, nes kompiuteris jas supranta pažodžiu. 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 tekstu sekas, sudarytas, pavyzdžiui, iš raidžių arba žodžių. Informatikos mokslininkai šiuo tikslu naudoja baigtinį automatą, kuris atlieka komandas ir patikrina, ar kompiuteris atpažsta žodžių ar simbolių eilutes. Pasitelkdami lobiu žemėlapius, susipažinsime su baigtinio automato darbu.

### Ryšiai su ugdymo programomis

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

### Gebėjimai

- ✓ Žemėlapio skaitymo
- ✓ Schemas aiškinimo
- ✓ Logikos
- ✓ Komandų (instrukcijų) vykdymo

### Amžius

- ✓ Nuo 9 metų

### Medžiaga

Reikia:

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

Kiekvienam mokinui reikia:

- ✓ Darbo lapo „Maršrutai iki lobio Hyjumos saloje“
- ✓ Rašymo priemonės

Gudručiams siūloma atlkti papildomų veiklų. Jiems reikia:

- ✓ Darbo lapo „Lobio slėpimas“
- ✓ Darbo lapo „Paslaptingas monetos žaidimas“



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# Lobių sala

---

## Ivadas

Tikslas – surasti lobiu salą Baltijos jūroje. Draugiški Baltijos jūros piratai plaukioja laivais tarp salų pastoviais maršrutais. Norėdami pralobti piratai siūlo keliones turistams. Iš kiekvienos salos išplaukia du jų laivai – A ir B. Turistai gali pasirinkti, kurį laivą plaukti. J kiekvieną salą galima nuvykti arba laivu A, arba laivu B, bet ne abiems. Saloje yra žmogus, kuris pasako, j kuriaj salą toliau plauks laivas. Tačiau piratai yra mažaraščiai, be to, neturi visų salų žemėlapio, todėl turistams duotame žemėlapyje reikia pažymėti maršrutą ir laivą, kuriuo galima keliauti toliau. Reikia surasti geriausią maršrutą į salą, kurioje paslėptas lobis.

## Demonstracija

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

Lentoje nupiešiamas trijų salų žemėlapis, kaip parodyta toliau paveiksle.

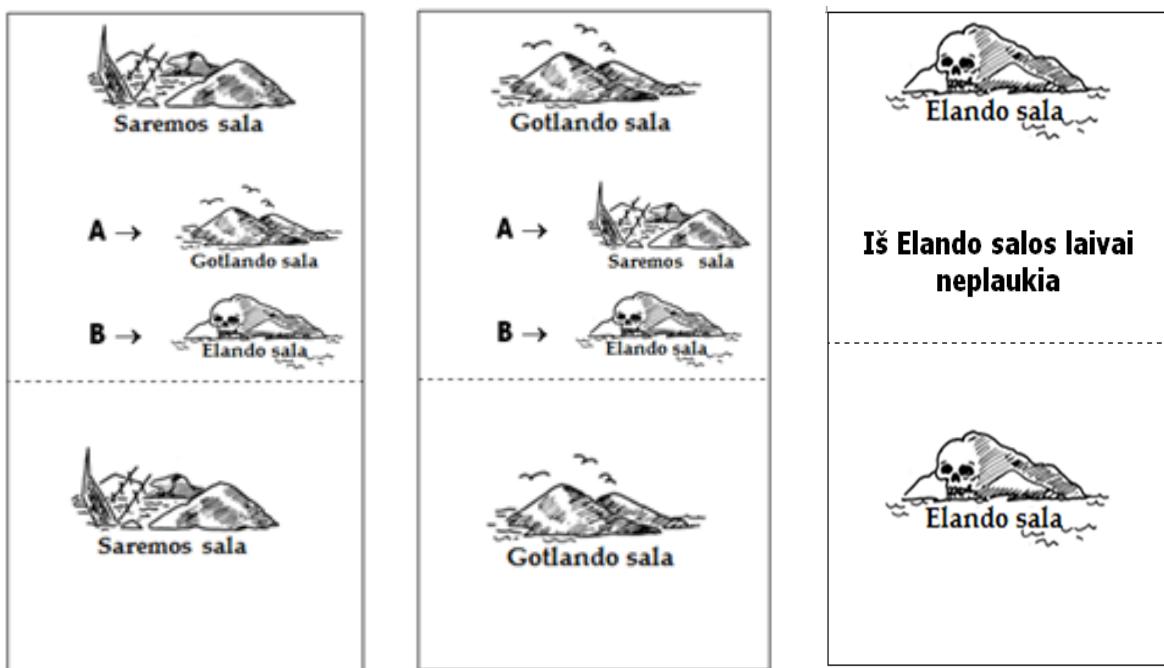


Atspausdinamos trys demonstravimo kortelės. Pakviečiami trys mokiniai, kurių kiekvienam duodama salos kortelę ir instrukcija. Jspėjama, kad veikloje, kuriaj mokiniai atliks vėliau, bus naudojamos kitų maršrutų kortelės.



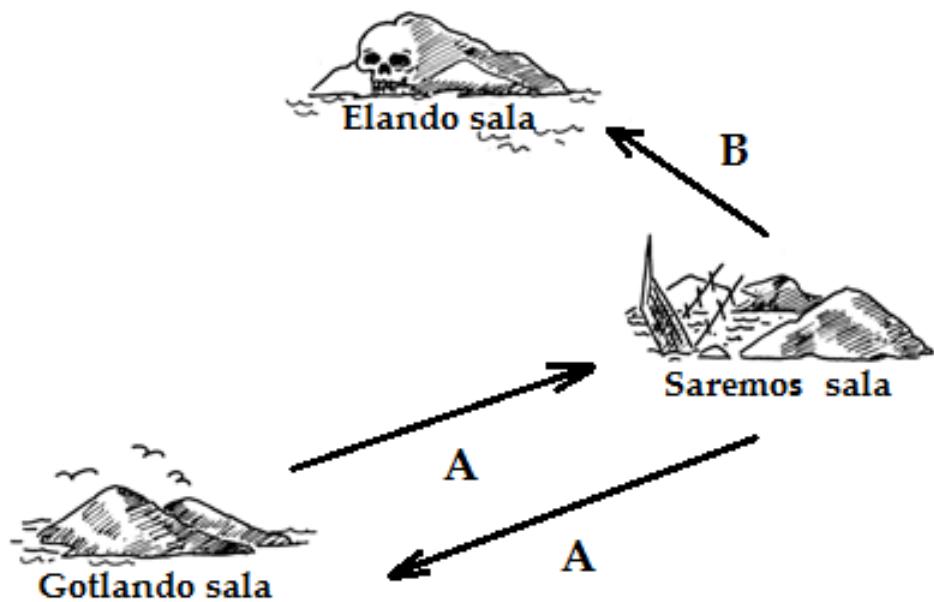
Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

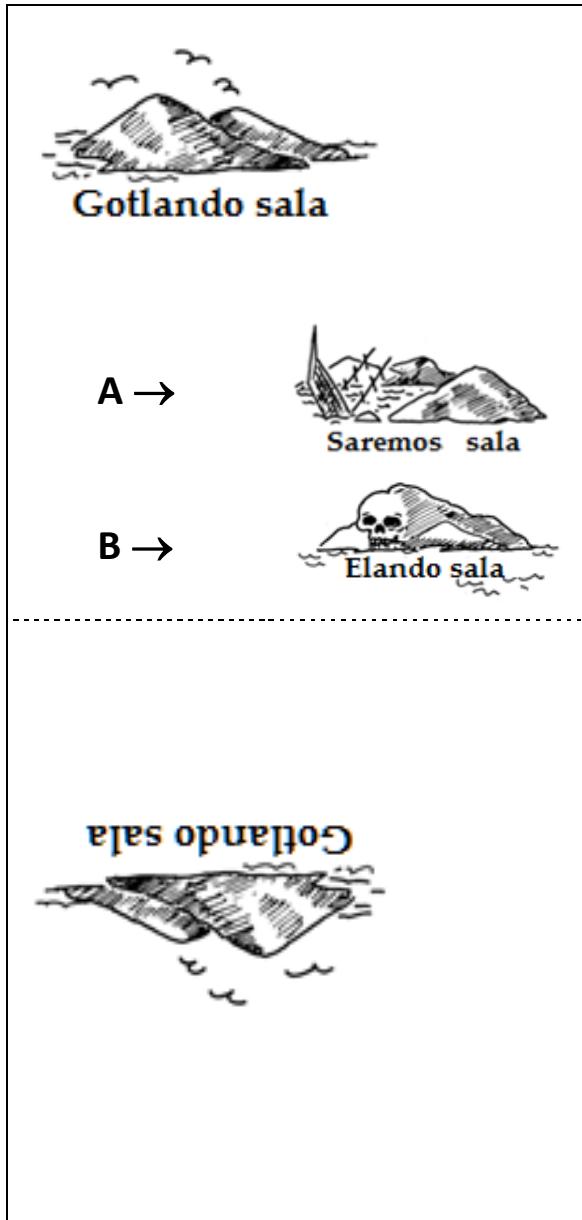
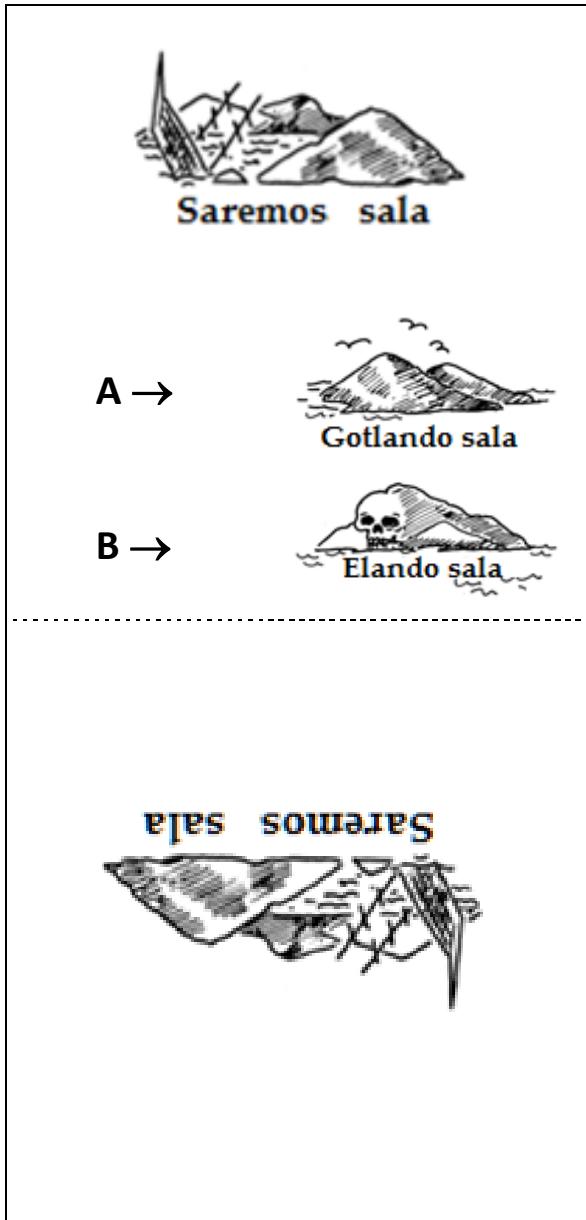


Kelionė pradedama Saremos saloje. Pasirenkamas laivas A. Mokiniai gali nukreipti laivą į Gotlando salą. Pažymimas maršrutas lentoje. Gotlando saloje vėl pasirenkamas laivas A. Grįztama į Saremos salą. Lentoje pažymima kelionė. Dabar pasirenkamas laivas B. Šis laivas plaukia į Elando salą, kur patenkama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

Galutinis maršrutas atrodo taip:

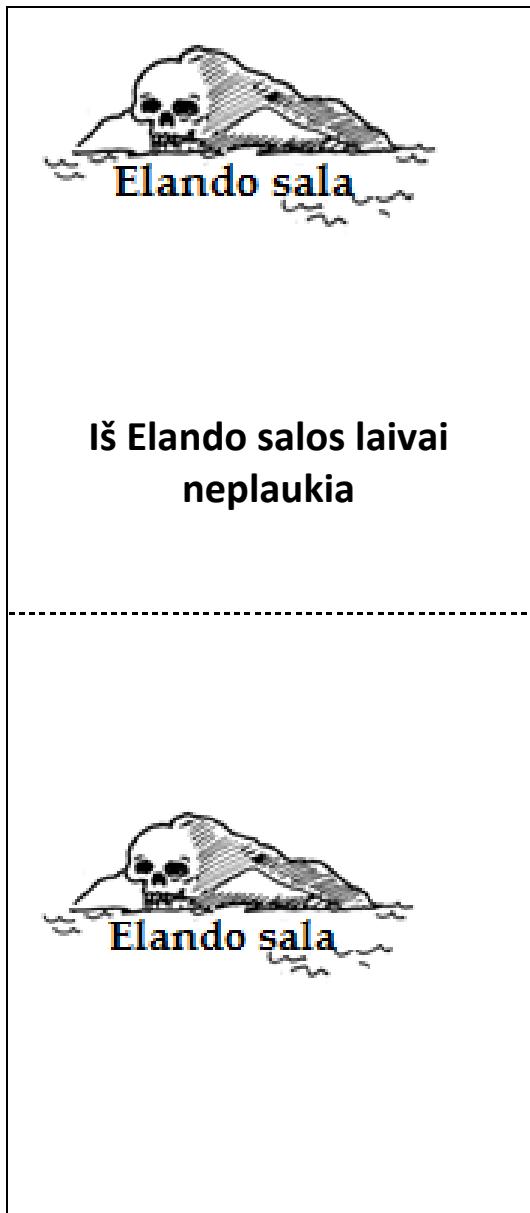


## Demonstravimo kortelės



## Demonstravimo kortelės

---



Iš Elando salos laivai  
neplaukia

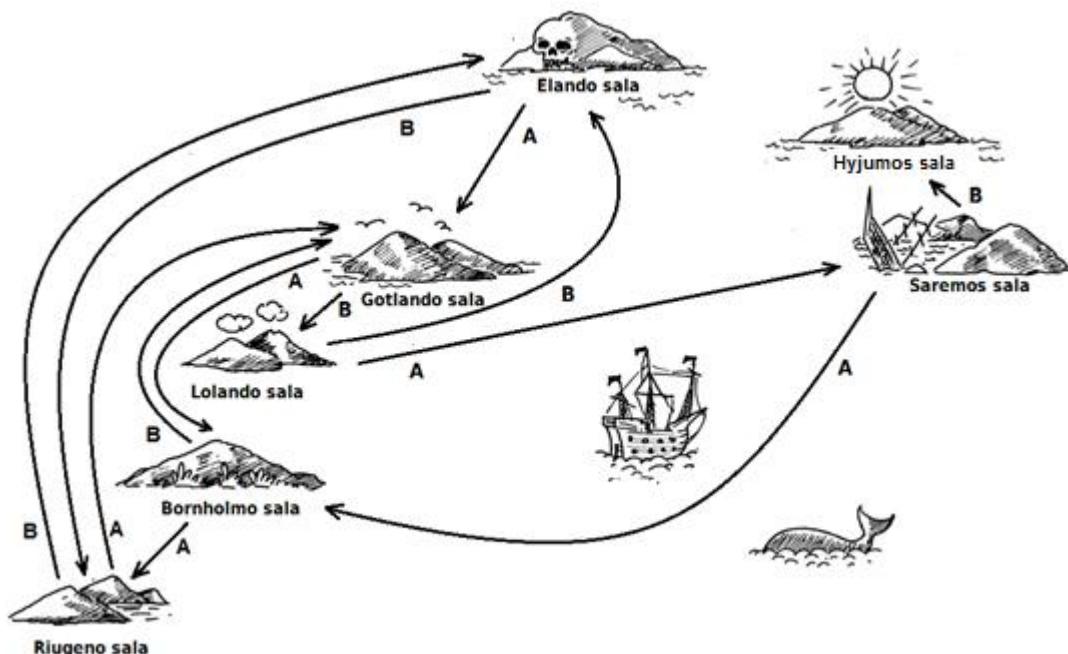


## Veikla

Parenkami 7 mokiniai, turėsiantys vaizduoti salas. Jie laiko savo salos kortelę, kurios kitoje pusėje užrašytos slaptos instrukcijos. Salas vaizduojantys mokiniai atsistoja kas sau aikštélėje ar kambaryje. Likusieji mokiniai vaizduos turistus. Kiekvienam iš jų duodamas salų žemėlapis, 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 išsiusti už durų.

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

Visi maršrutai parodyti paveiksle:



## Diskusija

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

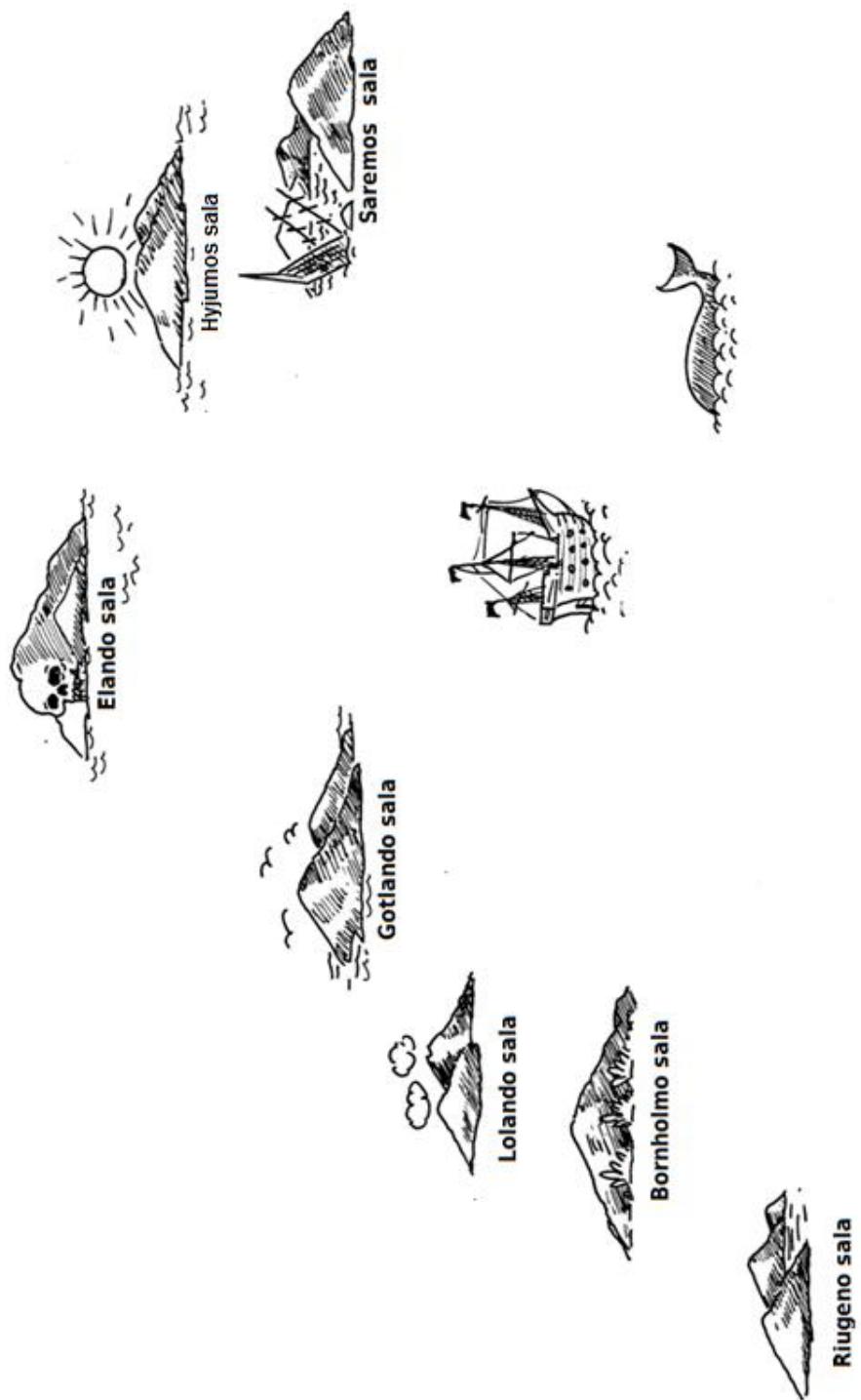


Licensed under Creative Commons

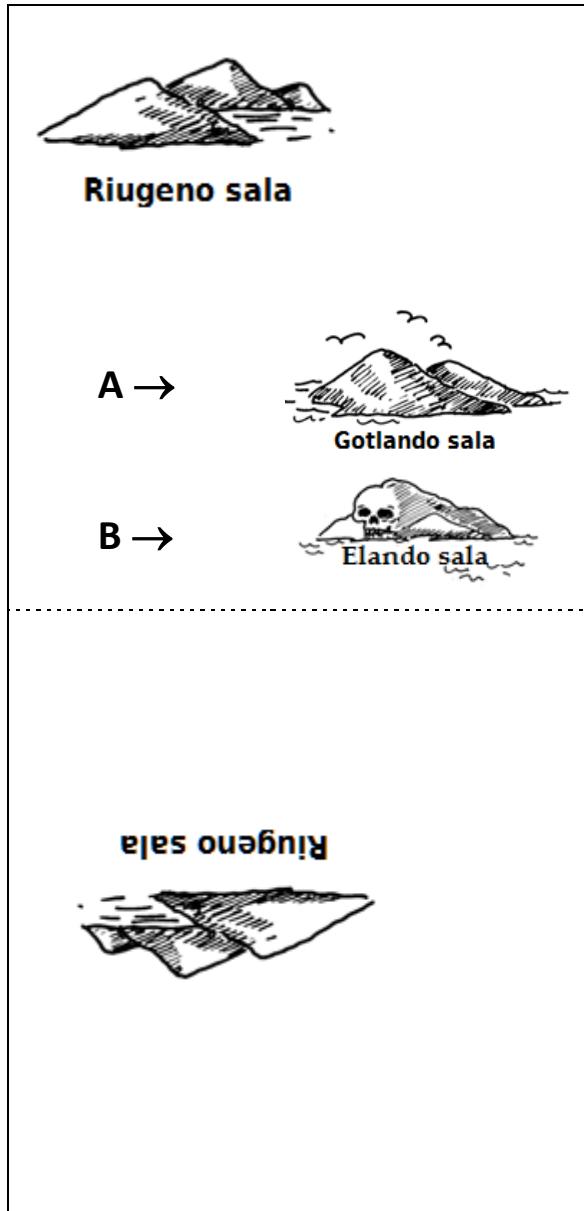
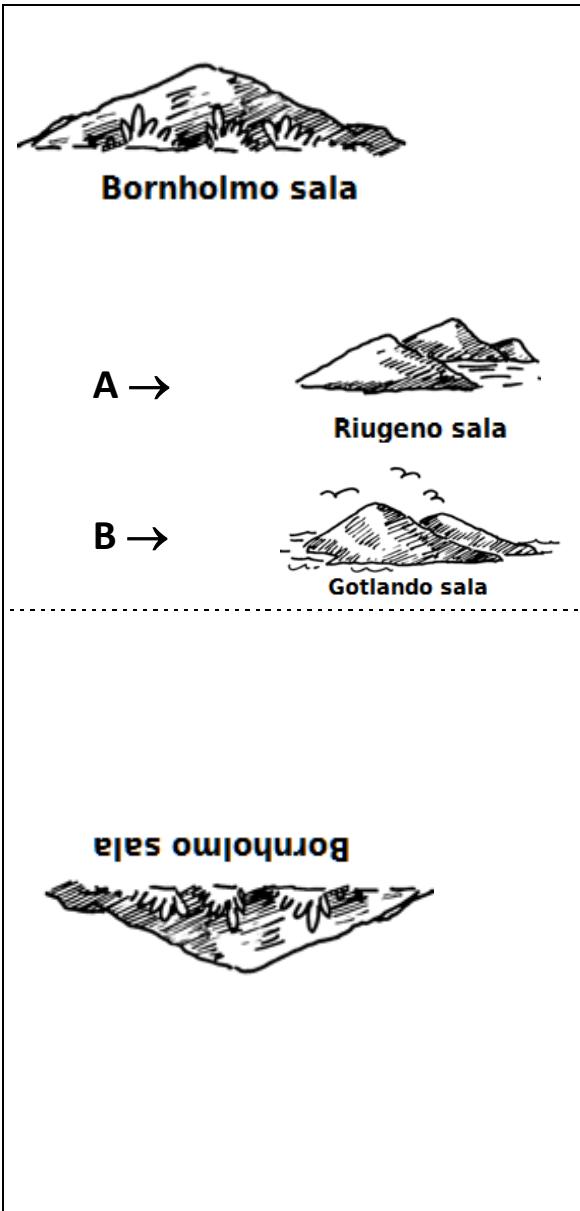
© 2015 Computer Science Unplugged (csunplugged.org)

## Darbo lapas Maršrutai iki lobio Hyjumos saloje

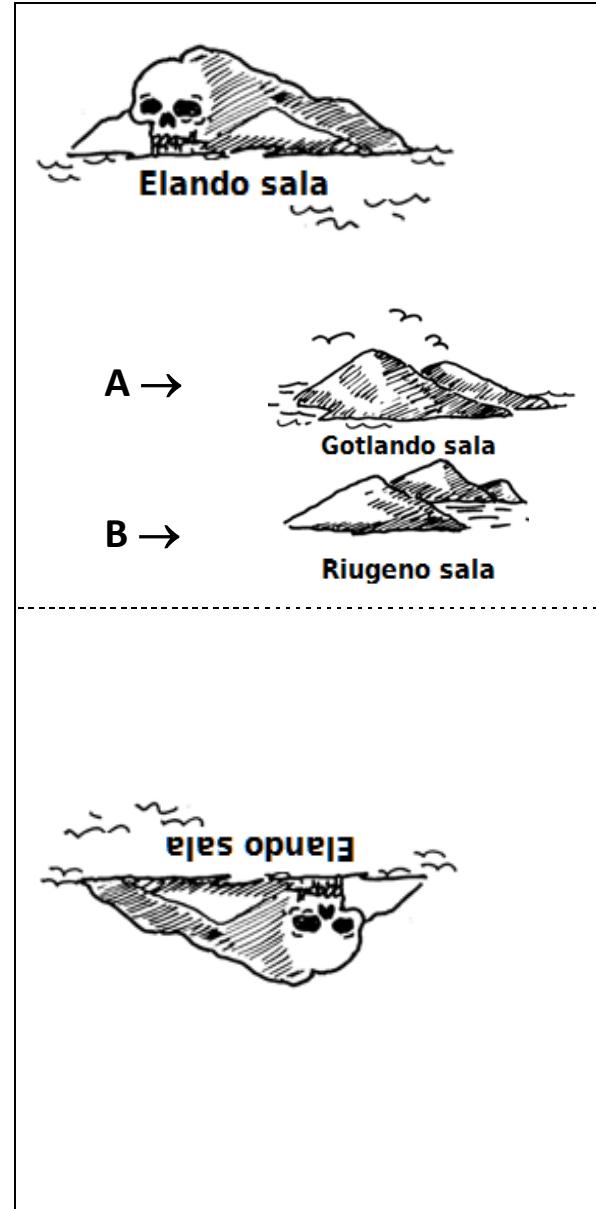
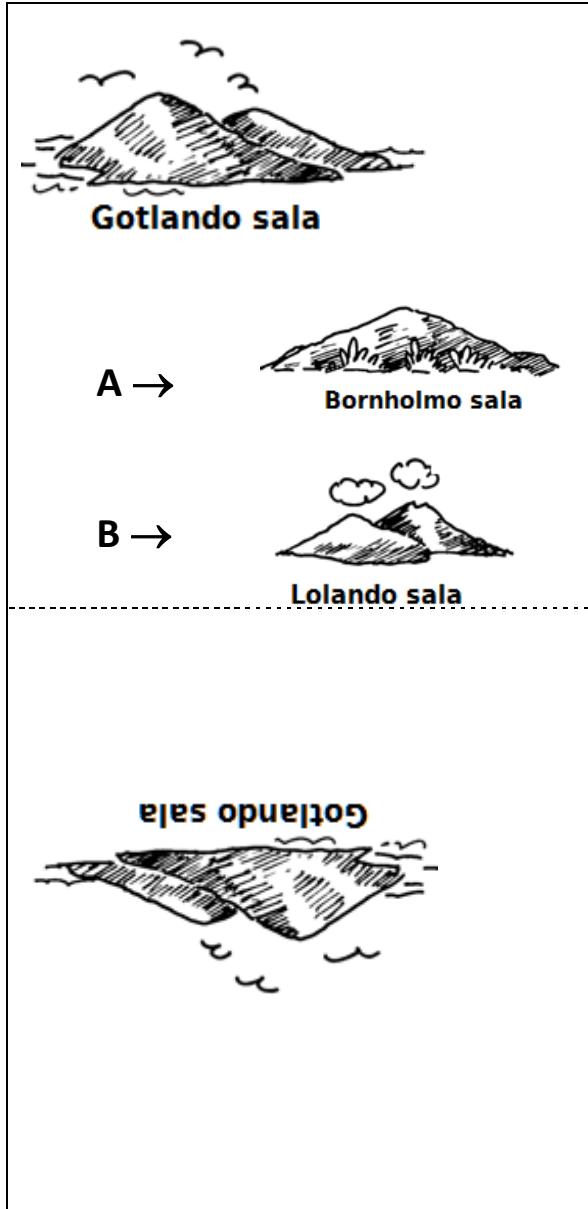
---



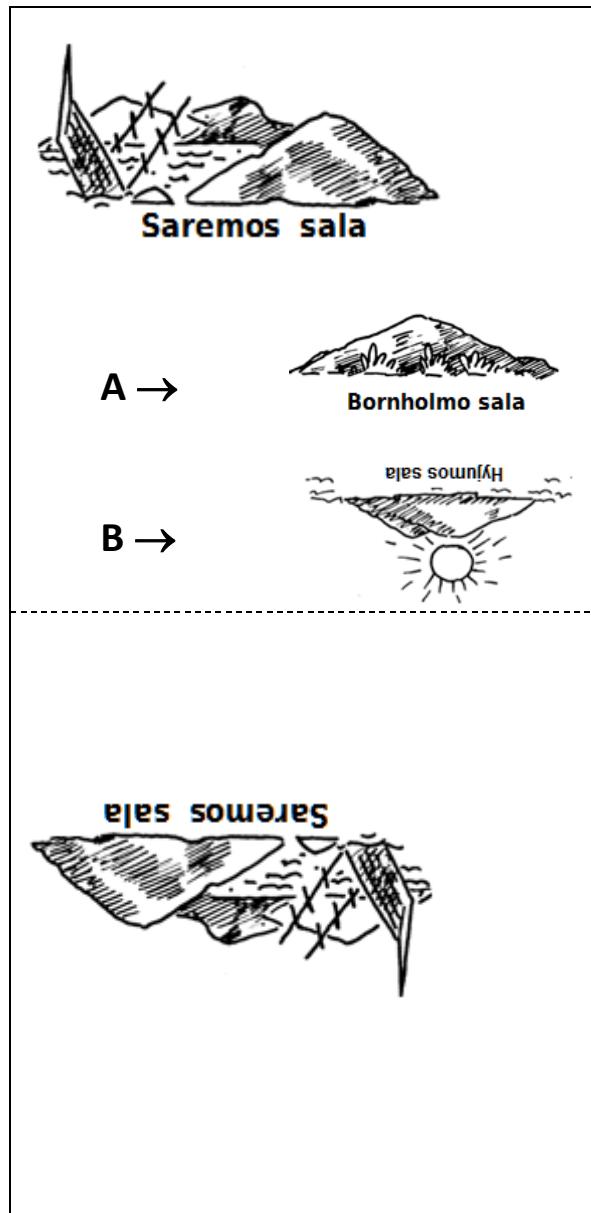
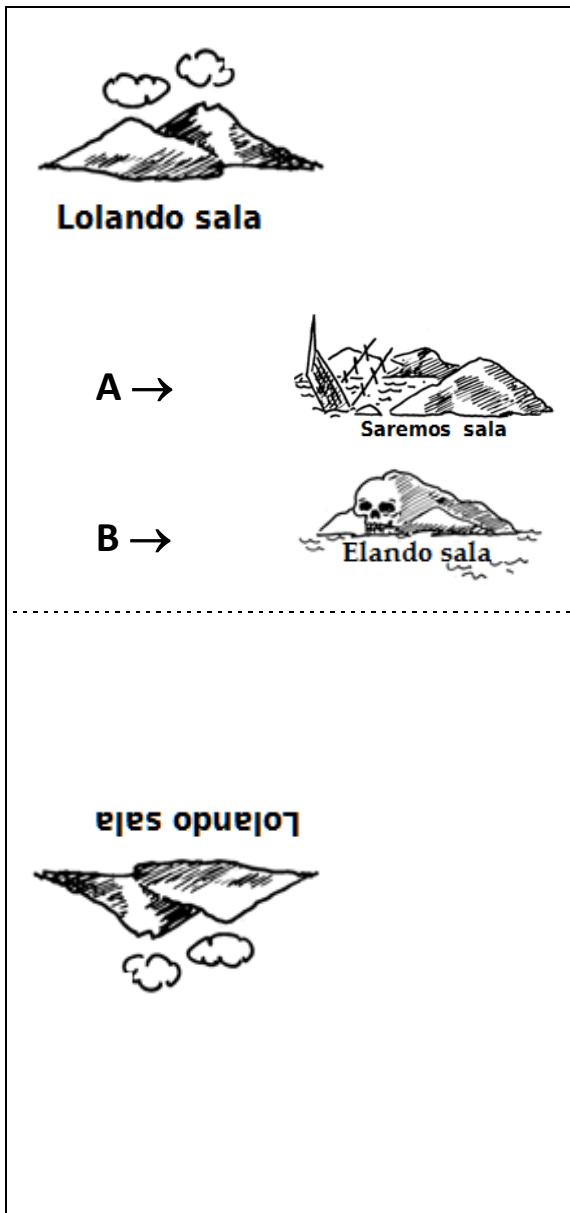
## Salu kortelēs su instrukcijomis (1/4)



## Salu kortelēs su instrukcijomis (2/4)

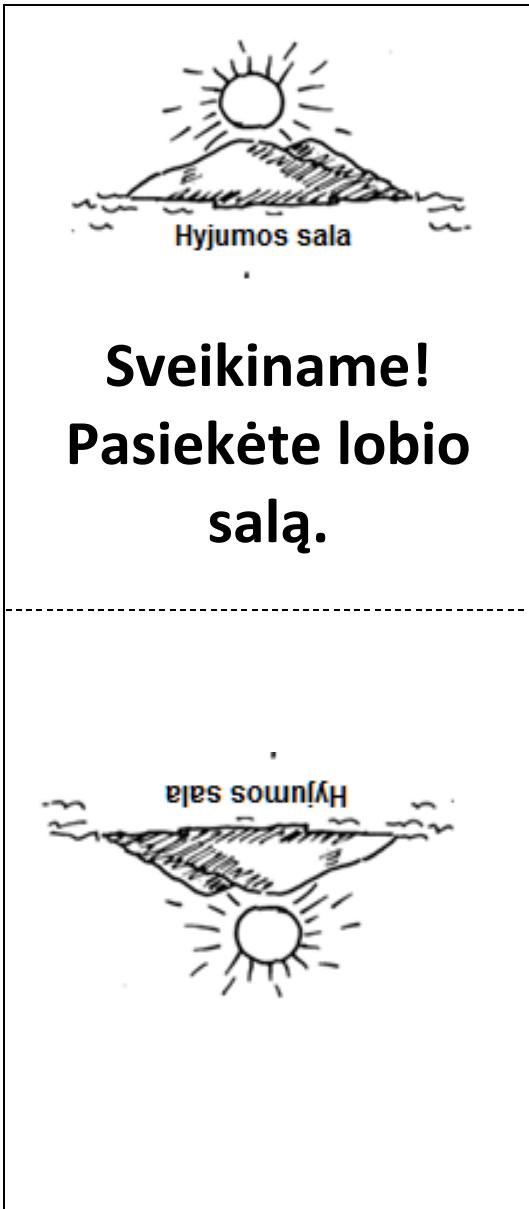


## Salū kortelės su instrukcijomis (3/4)



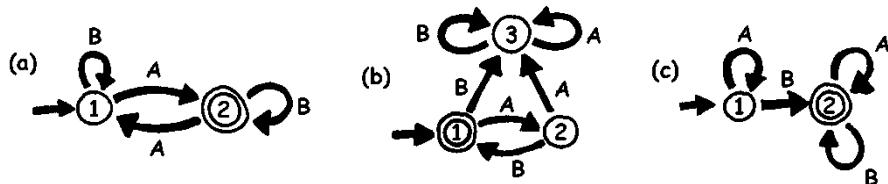
## Salū kortelės su instrukcijomis (4/4)

---



# Baigtinis automatas

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

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

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



Licensed under Creative Commons

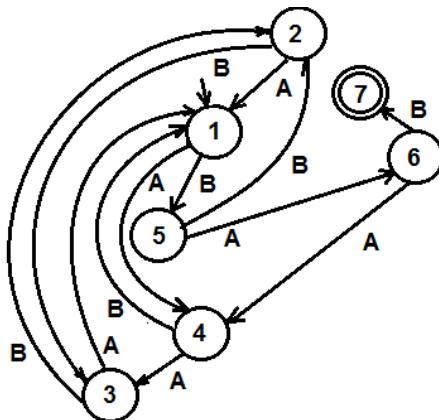
© 2015 Computer Science Unplugged (csunplugged.org)

## Darbo lapas *Lobio slėpimas*

---

Ar galėtumėte labai gerai paslėpti lobį? Kur geriausia paslėpti lobį, kad būtų sunku jį rasti? Laikas patiemis nubraižyti lobio žemėlapį!

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

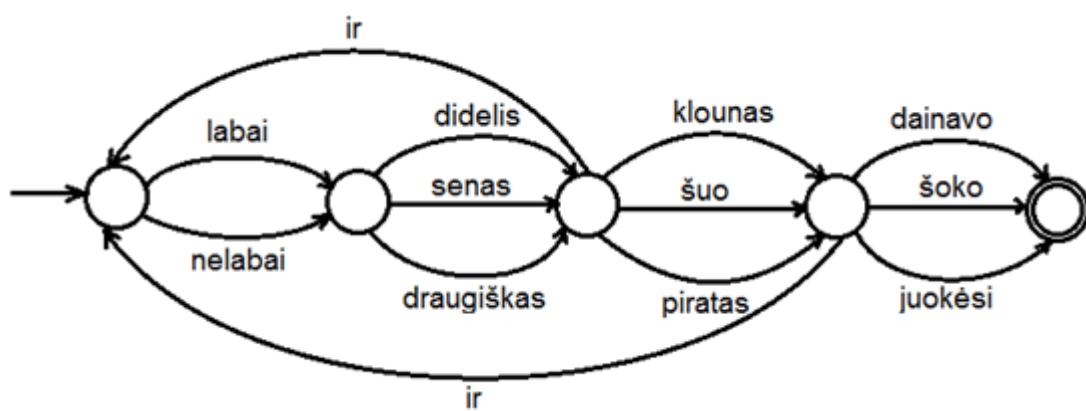


Pavaizduokite savo piratų plaukimo maršrutas schema, kaip tai daroma informatikoje. Nupieškite savo salų žemėlapius ir sudarykite salų korteles su savo instrukcijomis. Koks trumpiausias maršrutas, kuriuo pasiekiamas jūsų paslėptas lobis?

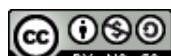
2. Išbandykite savo salų žemėlapj ir korteles su draugais. Duokite jiems A ir B sekas ir pažiūrėkite, ar jie suras paslėptą lobį.

Galite sukurti kitokį žaidimą, kurie iliustruoja baigtinio automato idėją.

3. Toliau paveiksle pavaizduota, kaip sudaryti sakinius baigtiniu automatu. Sakiniai sudaromi pasirenkant leidžiamus kelius schemaje ir užrašant kelyje pasitaikančius žodžius.



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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## **Darbo lapas *Paslaptinges monetos žaidimas***

---

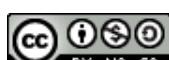
Draugai atsiuntė žaidimą iš interneto. Žaidime robotas meta monetą ir reikia spėti, kuria puse – herbu ar skaičiumi – ji atsivers. Iš pradžių žaidimas pasirodė labai lengvas. Tikimybė laimėti yra 0,5. Bent jau taip manė draugai. Po kurio laiko draugai pradėjo abejoti. Jie pastebėjo, kad moneta metama 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):

HHSHHSHHHSSHHHSSSSHHHHSHHHSSSHHHSSHHHHSSSSHHSSHHSSHH  
SSHHSHHHHHHHHSSHHSSSSHHHHSSSSSS

Ar galėtumėte nuspėti, pagal kokią taisyklę metamos monetos?

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

**Užuomina:** joje yra tik 4 „salos“!



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

## 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 atsiliepama įrašu: „Jeि norite ..., spauskite 1, jeि norite ..., spauskite 2, jeि 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. Jeि 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, kurios padeda sudaryti anglų kalbos sakinius. Jie gali ir generuoti sakinius, ir vykdyti tai, ką vartotojas įveda. 1960 m. 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 tēsti“. Žinoma, ji nieko nesupranta, tačiau yra gana įtaigi. Kai kurie ją išbandę 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 vykdyti.

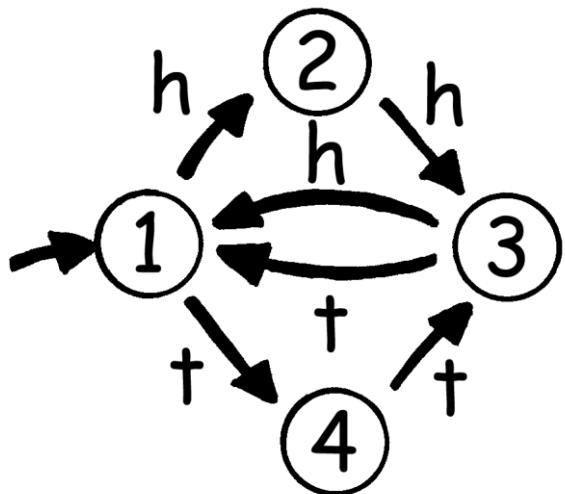


## Sprendimai ir užuominos

---

### Paslaptinges monetos žaidimas

Paslaptinges monetos žaidimas vyksta pagal tokią schemą:



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# 13 veikla

---

## Nurodymų eilė. Programavimo kalbos

### Santrauka

Kompiuteris gali atlikti veiksmus tik pagal algoritmą – programą, užrašytą taikant sutartas taisykles. Programa – tai algoritmas, užrašytas programavimo kalbos žyme–nimis taip, kad jį būtų galima atlikti kompiuteriu. Kompiuteris kiekvieną komandą supranta 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

Reikia:

- ✓ Kortelių su paveikslėliais

Kiekvienam mokinui reikia:

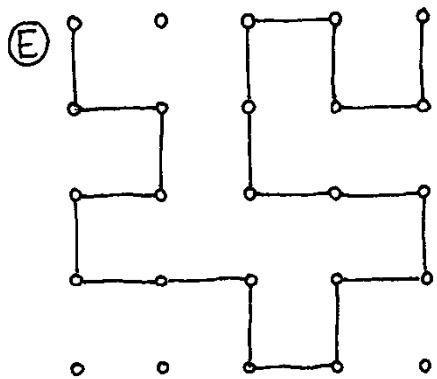
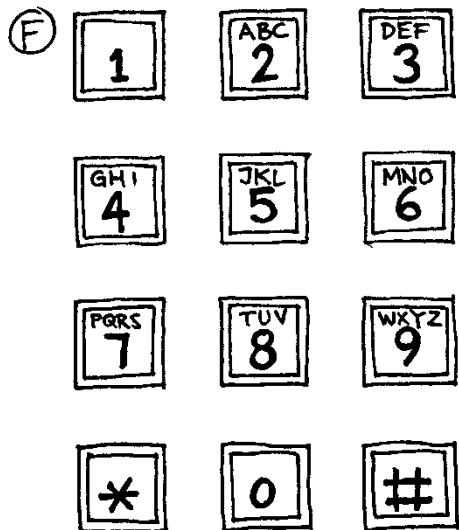
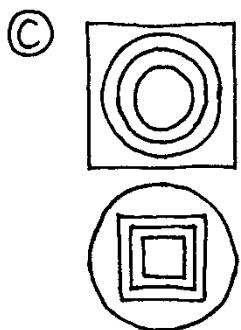
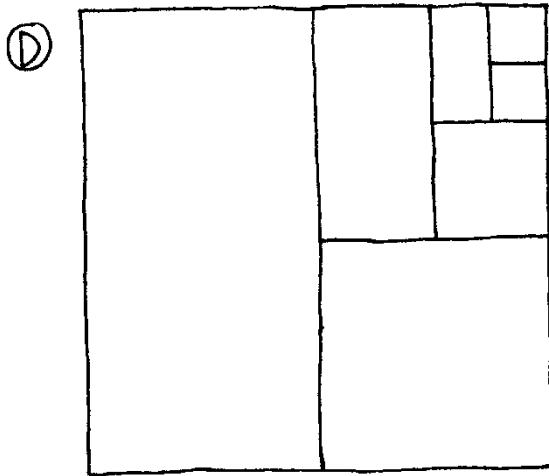
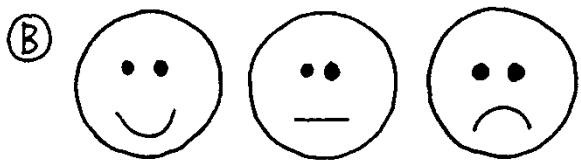
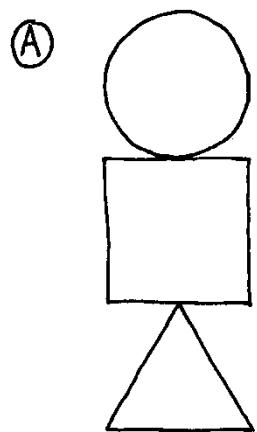
- ✓ Rašymo priemonės, popieriaus lapo ir liniuotės



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

125



# Nurodymų eilė

---

## Ivadas

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

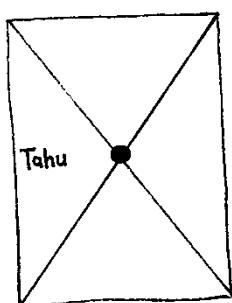
Kompiuteris tiksliai vykdo nurodytas komandas, net jei jos ir neturi jokios prasmės.

## Demonstracija

Mokiniai pabando vykdyti tikslius nurodymus:

1. Pažymėkite tašką lapo viduryje.
2. Pradėdami nuo kairiojo viršutinio lapo kampo su liniuote brėžkite tiesią liniją per tašką centre iki apatinio dešiniojo lapo kampo.
3. Pradėdami nuo apatinio kairiojo lapo kampo su liniuote brėžkite liniją per tašką centre iki viršutinio dešiniojo lapo kampo.
4. Parašykite savo vardą viduriniame trikampyje lapo kairėje.

Mokinių rezultatas turėtų atrodyti taip:



## **Veiklos**

1. Parenkamas mokinys, kuriam duodamas vienas iš aukščiau pateiktų paveikslielių. 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 piešti 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. Atkreipiama 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 ką nors nupiešia ir parašo to piešinio nurodymus, kuriuos gali išbandyti poromis ar su visa klase.

## **Variantai**

Mokiniai galėtų:

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

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

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

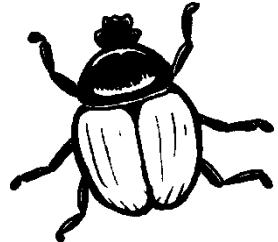
## Apie ką visa tai?

---

Kompiuteris gali atlikti tik tiksliai užrašytą algoritmą, vadinamą programą. Programai užrašyti vartojamos programavimo kalbos, sukurtos pagal tam tikras taisykles ir turinčios ribotą skaičių kompiuteriui suprantamą komandą. Vienos kalbos yra tinkamesnės vienokiam uždavinui spręsti, kitos – kitokiam.

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

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



Kuo sudėtingesnė programa, tuo didesnė jos kladų tikimybė.

Programos sudėtingumas tapo pagrindiniu iššūkiu, kai JAV buvo kuriamas Strateginės gynybos („Žvaigždžių karų“) programos kompiuterių valdymo sistema. Ši sistema turėjo sudaryti nejveikiamą gynybą nuo branduolinės atakos. Kai kurie informatikai tvirtino, kad sistema niekada neveiks, nes programinė įranga yra laba sudėtinga ir nepatikima. Pagrindinis jų argumentas buvo tai, kad programinę įrangą reikia testuoti ir nustatyti kiek įmanoma daugiau kladų. Esą šios sistemos ištetsuoti neįmanoma, nes norint išsitikinti, kad ji veikia, reikia patikrinti visas jos galimybes, iš kurių viena – paleisti JAV branduolines raketas.



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))



Licensed under Creative Commons  
© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# IV dalis

## Ko kompiuteris negali?

### *Sunkiai sprendžiami uždaviniai*



# Sunkiai sprendžiami uždaviniai

---

Ar yra tokius uždavinius, 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 ju aprašyti kompiuteriui suprantamomis komandomis. Tačiau yra ir 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 kompiuterio sprendžiamas uždavinys yra toks, kurį net ir sparčiausias kompiuteris sprendžia daug ilgiau, nei trunka žmogaus gyvenimas.

Šioje dalyje pateikiama uždavinių, neturinčių efektyvių sprendimų. Kompiuteris juos spręstų milijonus šimtmečių. Taip pat supažindinama su didžiausia informatikos mokslo paslaptimi: niekas nežino, ar yra efektyvesnis uždavinio sprendimo būdas! Tai reiškia, kad nėra jrodomo, kad efektyvus sprendimo būdas egzistuoja arba kad tokio sprendimo nėra. Informatikoje yra tūkstančiai neišspręstų visiškai skirtingu, 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 gretimoms 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 pilumo uždaviniais. NP pilumo sąvoka aprašoma skyreliuose „Apie ką visa tai?“ kiekvienos šios dalies veiklos aprašo pabaigoje. Veiklos turi būti atliekamos iš eilės. Jas atlikę mokiniai jau turės supratimą apie svarbiausius atviruosius informatikos klausimus.

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# 14 veikla

---

## Vargšas kartografas. Grafo spalvinimas

### Santrauka

Daugelyje optimizavimo uždavinijų tam tikri įvykiai negali vykti vienu metu ar tam tikri objektai negali būti gretimi. Pavyzdžiui, kiekvienas, kam teko daryti klasės ar susitikimo tvarkaraštį, yra susidūrės su visų dalyvių sąlygų suderinimo problema. Tokio pobūdžio uždaviniai gali būti sprendžiami taikant grafo spalvinimo metodą. Spalvinant grafą siekiama, kad gretimi jo elementai būtų skirtingu spalvų. Atliekant šią veiklą grafo spalvinimo metodu sudaromas žemėlapis.

### Ryšiai su programomis

- ✓ Matematika: skaičiai, dvejetainė skaičiavimo sistema
- ✓ Matematika: algebra, begalinė skaičių seka, skaičių sekos taisyklės, dvejeto 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 mokinui reikia:
- ✓ Vieno ar kelių darbo lapo kopijų
- ✓ Spalvotų smeigtukų ar spalvotų lipniujų lapelių
- ✓ Spalvotų pieštukų, flomasterių ar kreidelių



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

133

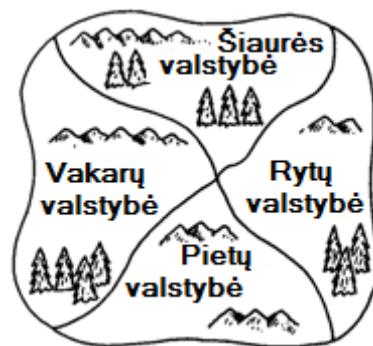
# Grafo spalvinimas

## Ivadas



Šiame skyriuje pasakojama apie kartografių, kuriam mokiniai turi padėti nuspaldinti žemėlapį taip, kad gretimos valstybės būtų skirtingų spalvų.

Pavyzdžiui, paveiksle dešinėje parodytas žemėlapis su keturiomis valstybėmis: Šiaurės, Vakarų, Rytų ir Pietų. Jei Šiaurės valstybė nuspaldinama raudonai, tai Vakarų negali būti spalvinama raudonai, nes nesimatys sienos su Šiaurės valstybe. Vakarų valstybę galima spalvinti, pavyzdžiui, žaliai. Tada ir Rytų valstybę 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 išsigyti daug spalvotų pieštukų, todėl prašoma naudoti kuo mažiau spalvų.

## Diskusija

Mokiniams užduotis paaiškinama lentoje.

Išdalijami pirmieji darbo lapai. Žemėlapiui darbo lape nuspaldinti 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 žemėlapį dviem spalvomis mokiniai gali suformuluoti taisykłę: jei viena valstybė spalvinama viena spalva, tai jos kaimynė turi būti spalvinama kita spalva. Ši taisykla kartojama, kol nuspaldinamas visas žemėlapis. Geriausia, kad šią taisykľę mokiniai sugalvotų patys.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

Pirmajį lapą baigusiems spalvinti mokiniams duodamas antrasis darbo lapas.

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

Vyresnių mokinį galima paprašyti paaiškinti, kokiu būdu spalvinant žemėlapį prieikia mažiausiai spalvų. Pavyzdžiui, šiam žemėlapiui reikia bent trijų spalvų, nes daugiausiai trys valstybės turi bendrų sienų.

Kai mokiniai nuspaldina visus žemėlapius, galima paprašyti, kad sukurtų ir savo žemėlapių, kuriems nuspaldinti reikėtų mažiausiai penkių spalvų, arba kad įrodytų, kad bet kuriam žemėlapiui nuspaldinti reikia mažiausiai keturių spalvų. Tokiai užduočiai atlikti mokiniams turėtų prieikti daugiau laiko. Mokiniai gali greitai nubraižyti žemėlapį, kuriam nuspaldinti reikia penkių spalvų, bet daugeliu atvejų galima parodyti, kad jį galima nuspaldinti ir keturiomis spalvomis.



Licensed under Creative Commons

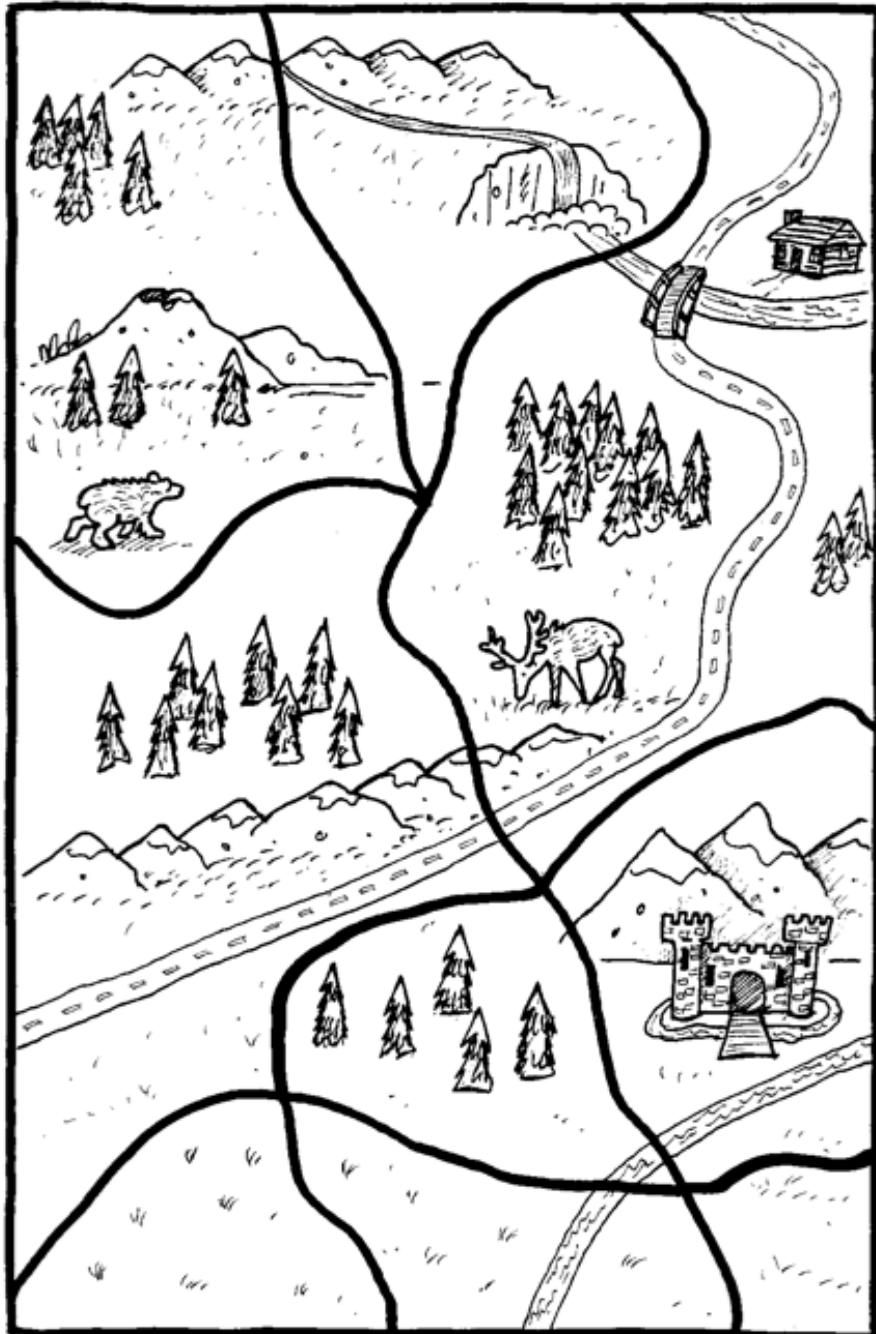
© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

135

## Darbo lapas Grafo spalvinimas (1)

---

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

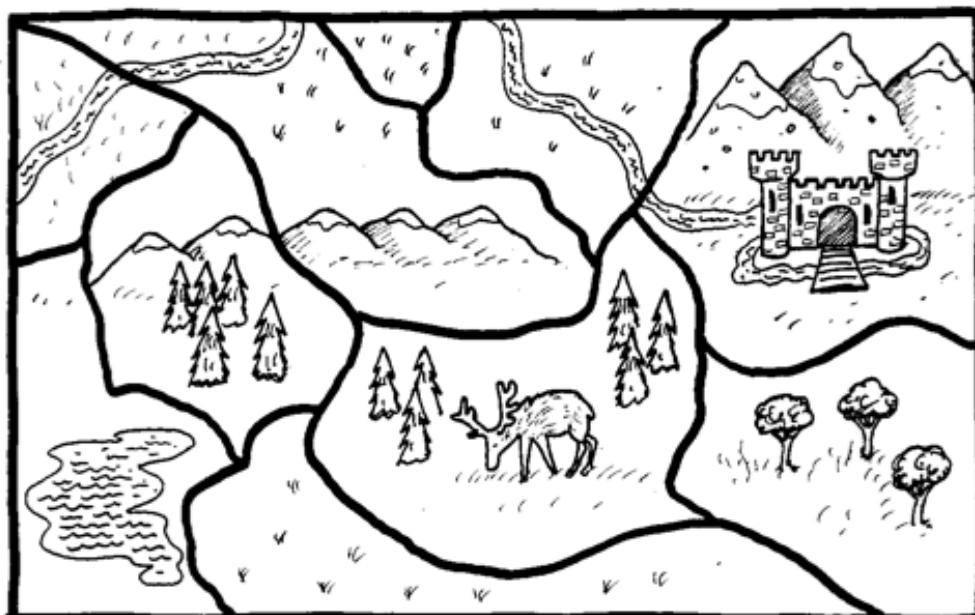
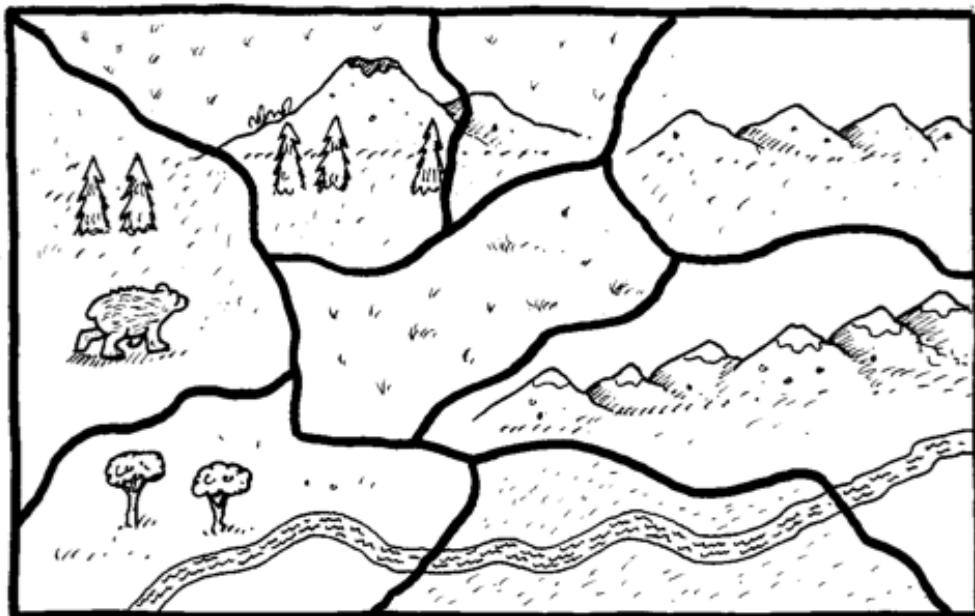


Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

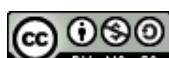
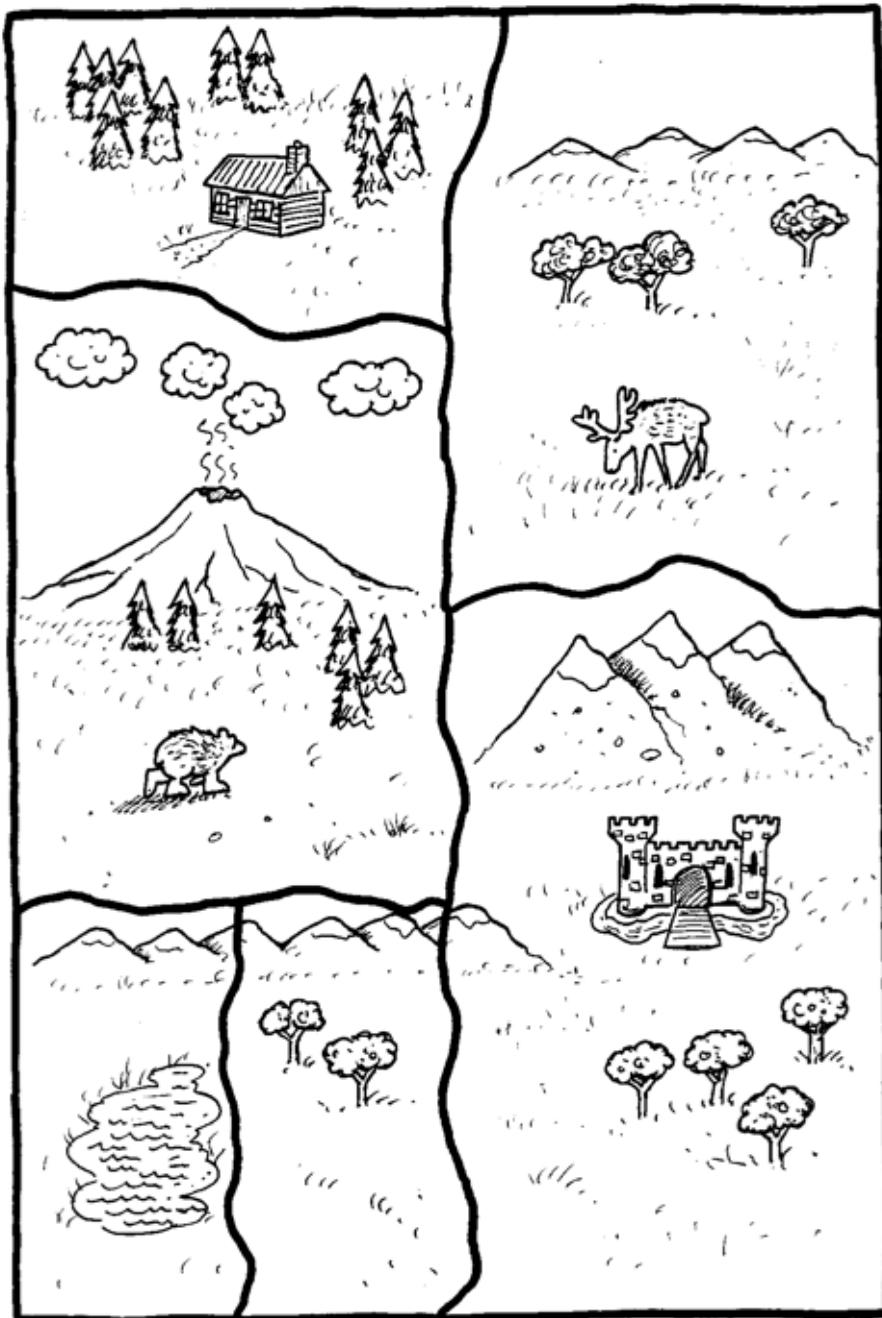
## Darbo lapas Grafo spalvinimas (2)

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.



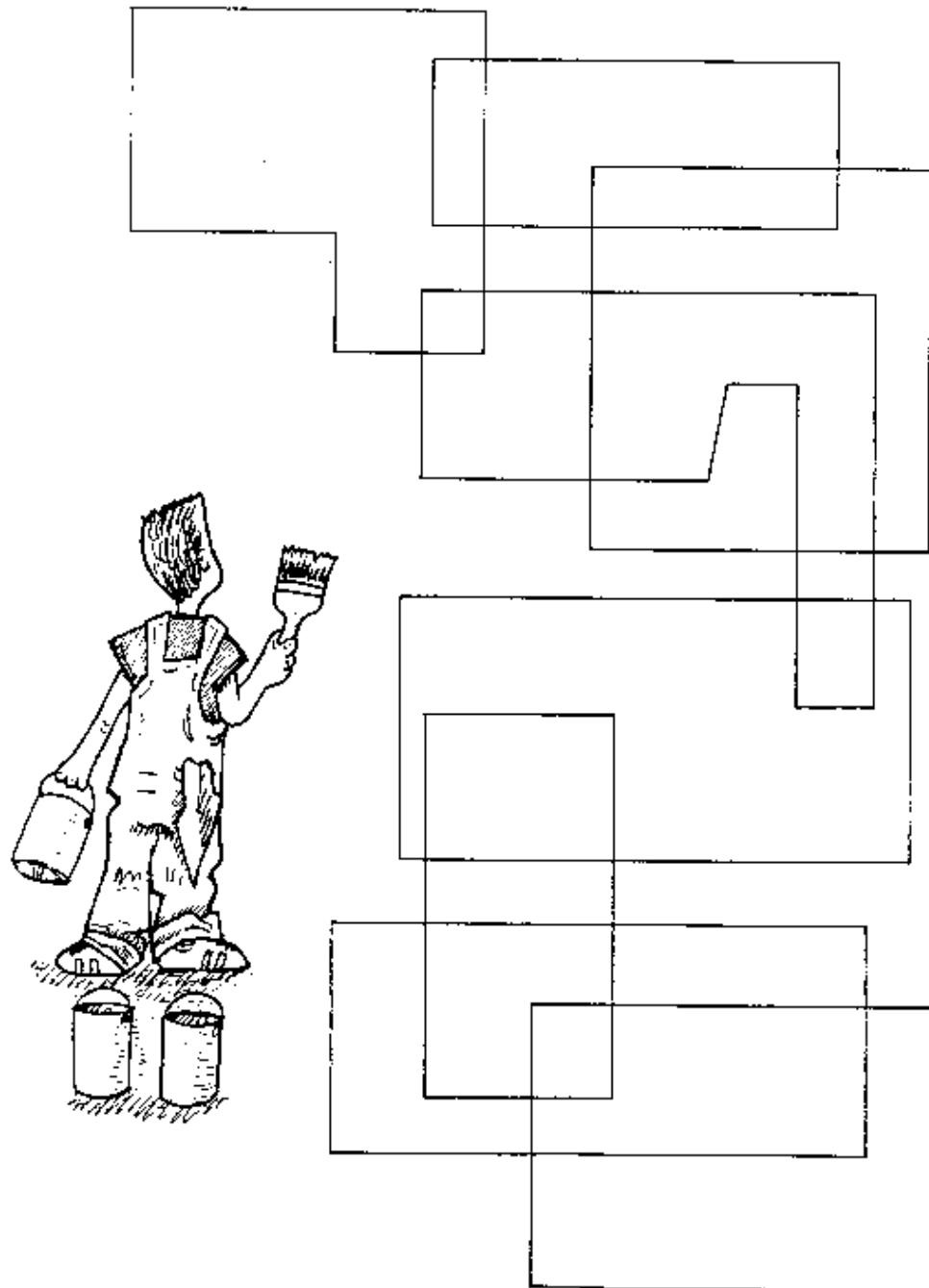
Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Darbo lapas *Grafo spalvinimas (4)*

---

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



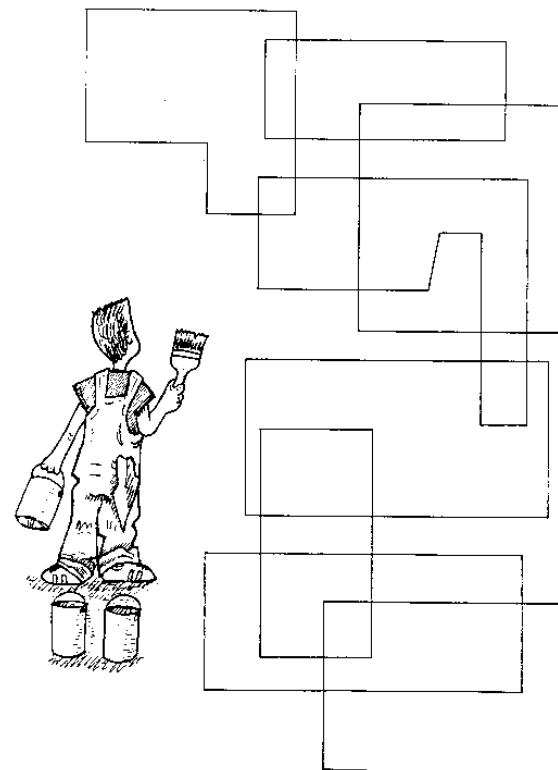
## Gudručiams

Paprasta sukurti žemėlapj, kuriam nuspalvinti reikia mažiausiai dviejų spalvų. Toks pavyzdys parodytas ketvirtame darbo lape, kur schema yra sudėliota iš susiklojančių uždarų geometrinių figūrų. Taip sudėliotą schemą ar žemėlapį 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 jdomu, kiek spalvų reikia žemėlapiui, kuris nupieštas ant kitokio paviršiaus, pavyzdžiu, toro. Šiuo atveju visada pakanka penkių spalvų. Mokiniai galėtų tai išbandyti.

Grafų teorijoje yra įvairių grafo spalvinimo uždavinijų interpretacijų. Pavyzdžiu, jei vienas žmogus spalvina žemėlapj, nubraižytą ant lapo, tai tikrai žino, kad protingai spalvinant užtenka keturių spalvų. O jei tas žmogus spalvintų valstybes paeiliui ne vienas, o su prieškai nusiteikusiu ar grafų spalvinimo teorijos nežinančiu partneriu? Vienas spalvina viską gerai apgalvodamas, o kitas neapdairiai spalvina valstybes paeiliui. Kiek mažiausiai skirtingu spalvų reikėtų žemėlapiui nuspalvinti? Didžiausias skaičius nėra žinomas. 1992 m. buvo įrodyta, kad 33 skirtingu spalvų visada pakaktų. 2008 m. buvo įrodyta, kad visada pakaktų 17 skirtingu spalvų. Ekspertai spėja, kad turėtų užtekti 10 skirtingu 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 skirtingu, 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 skirtingu spalvų reikia šiems žemėlapiams nuspalvinti? Atsakymas dar nežinomas.



## Apie ką visa tai?

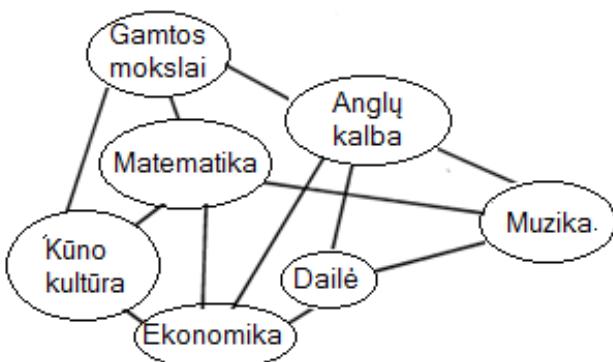
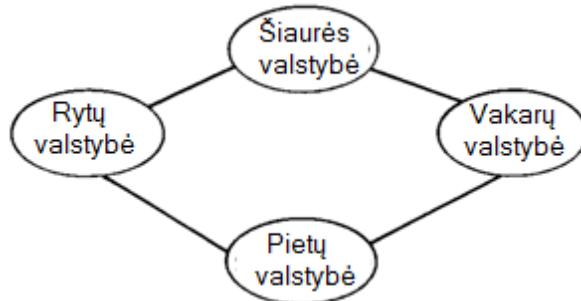
„Žemėlapio spalvinimo“ uždavinyje, spręstame atliekant šią veiklą, reikėjo surasti mažiausią skaičių skirtinį spalvą (dvi, trys, keturios spalvos), kuriomis būtų galima nuspalvinti vieną ar kitą žemėlapį. 1852 m. buvo suformuluotas teiginys, kad bet kokį žemėlapį galima nuspalvinti mažiausiai keturiomis skirtinomis 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 neatsakyti klausimų.

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

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

Paveiksle dešinėje viršūnės atitinka mokykloje mokomus dalykus. Linija, jungianti dvi viršunes, reiškia, kad mažiausiai vienas mokinys mokosi tų dviejų dalykų, todėl tvarkaraštyje abiejų dalykų pamokos negali būti įrašytos kaip vykstančios vienu metu. Optimalaus pamokų tvarkaraščio sudarymas yra grafo spalvinimo uždavinys, kai kiekviena spalva reiškia kitą pamokos laiką. Informatikos grafo 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 neoptimaliu, bet vis tiek labai geru rezultatu.

Kompiuterio sugaištas laikas grafo spalvinimo uždaviniui spręsti ilgėja eksponentiškai didėjant grafo dydžiui. Žemėlapio spalvinimo uždavinys gali būti sprendžiamas perrenkant visus jmanomus spalvinimo variantus. Žinoma, kad pakanka mažiausiai keturių skirtinį 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. Paprastu atveju, kai mažai valstybių, surasti optimalų sprendimą yra gana paprasta ir be kompiuterio, bet daugėjant valstybių uždavinys tampa nebeįsprendžiamas net ir kompiuteriu.

Realiame gyvenime dažnai tenka spręsti tokius uždavinius. Nors jie ir sunkiai sprendžiami, realiame gyvenime reikia gauti rezultatą. Informatikos metodai pateikia gana gerų, bet ne puikių sprendimų. Šie heuristinių 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 nejrodė, kad nėra efektyvaus būdo šiems uždaviniams spręsti, bet taip pat niekas nejrodė, kad jis yra. Informatikos mokslininkai lieka skeptiški, kad kada nors bus rastas efektyvus metodas. Kitų veiklų skyriuose daugiau pasakojama apie sunkiai sprendžiamus uždavinius.

## Daugiau informacijos

Davido Harelo knygoje „Algorithmics“ nagrinėjama keturių spalvų teorema, pateikiama jos istorija. Daugiau žemėlapio spalvinimo aspektų aptariama knygoje „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).

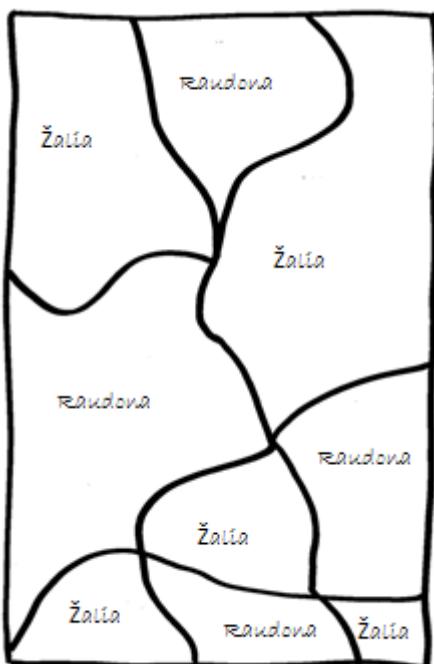
Mareko Kubale knygoje „Graph Colorings“ aprašoma grafo spalvinimo uždavinių istorija. Šia tema galima rasti daug tinklalapių. (American Mathematical Society, Providence, Rhode Island, 2004).



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Sprendimai ir užuominos



A.



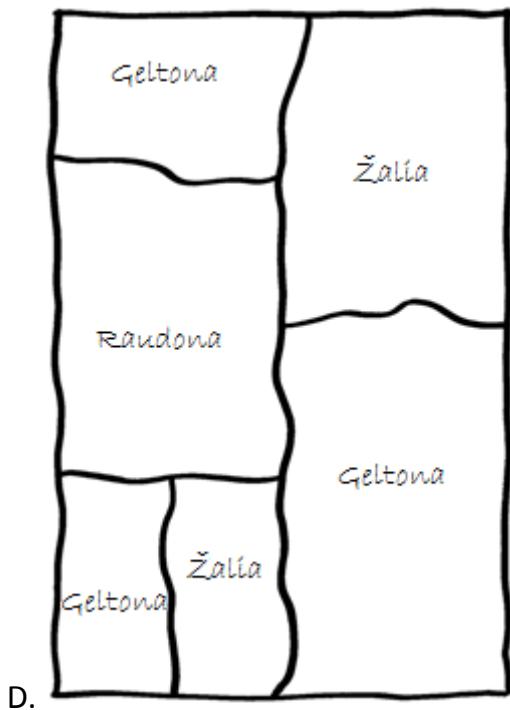
B.



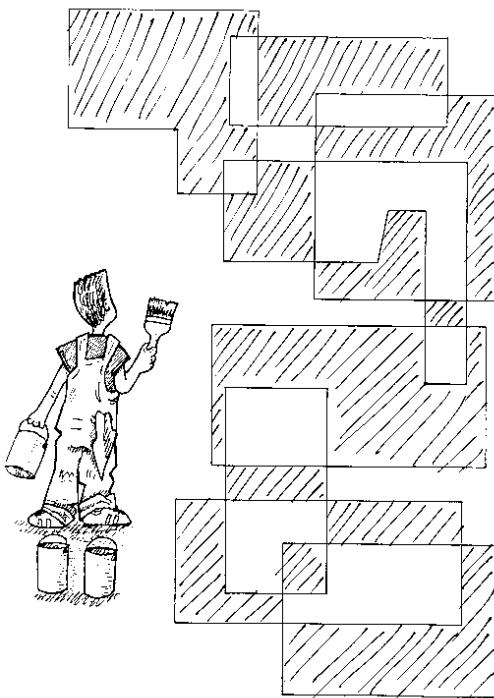
C.

Vienintelis pirmojo darbo lapo sprendimas parodytas A paveiksle (spalvos gali būti pasirenkamos).

Antrojo darbo lapo viršutiniams žemėlapiui reikia trijų spalvų. Galimas sprendimas parodytas B paveiksle. Apatiniams žemėlapiui reikia keturių spalvų. Galimas sprendimas parodytas C paveiksle.



E.



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

Ketvirtijo 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 viršunes taip, kad kiekviena viršūnė būtų nutolusi ne daugiau kaip per vieną žingsnį nuo kurios nors iš kitų pažymėtų viršūnių. Kiek mažiausiai tokų viršūnių reikia pažymėti?

### Ryšiai su ugdymo programomis

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

### Gebėjimai

- ✓ Žemėlapio skaitymo
- ✓ Ryšių nustatymo
- ✓ Dėlionės sprendimo
- ✓ Pasikartojančio tikslų siekimo

### Amžius

- ✓ Nuo 7 metų

### Priemonės

Kiekvienai mokinui grupei reikia:

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

Reikia:

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



# Dominuojančios sekos

## Ivadas

„Ledų furgono“ darbo lape grafu vaizduojamas Turistų miesto žemėlapis. Briaunos arba lankai tame 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 pažymeti 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 pažymetose sankryžose.
5. Mokiniai eksperimentuodami žymi skirtinges 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 įvykdomos, kai kiekvienoje sankryžoje pastatoma po furgoną, bet jdomesnis klausimas – 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 rasti! Jei mokiniams sunku rasti mažiausią furgonų skaičių, palaukus kiek laiko galima jiems tą skaičių pasakyti ir paprašyti furgonus pažymeti grafe. Tačiau ir tai gali būti gana sunku. Mokiniams nelengva surasti net ir sprendimą su aštuoniais ar devyniais furgonais.
7. Galima paaiškinti mokiniams, kaip bražomas toks žemėlapis (žr. šios veiklos sprendimus ir užuominas). Mokiniai gali pabandyti sukurti savo Turistų miestą parodytu metodu. Vieniems gali būti jdomu sukurti lengvai išsprendžiamą dėlionę, o kitiems ir tai gali būti labai sunku. Šis pavyzdys vadinamas *vienos krypties funkcija*: lengva pasiūlyti sunkiai sprendžiamą dėlionę, kai žinomas jos kūrimo nuo sprendinio metodas. Vienos krypties funkcijos yra kriptografijos pagrindas (17 ir 18 veiklos).



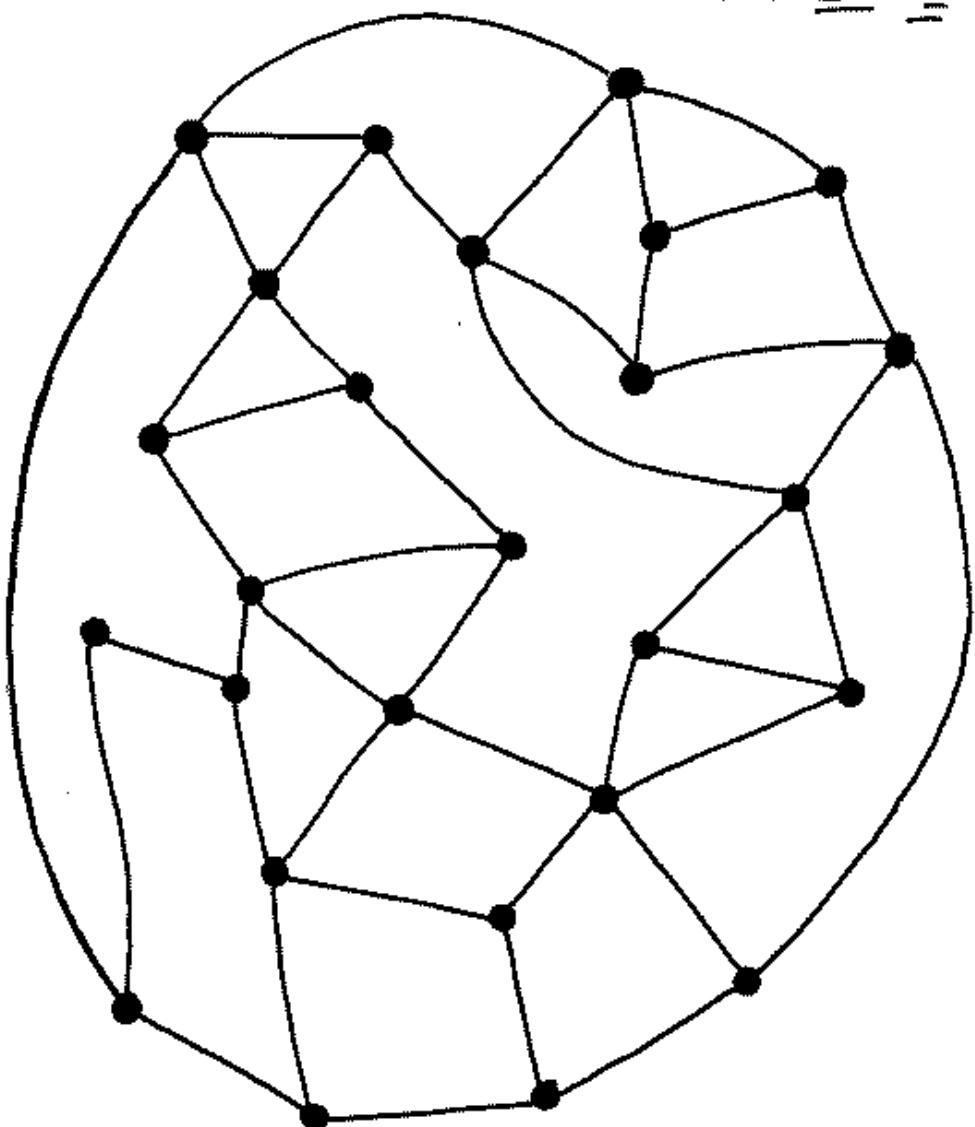
Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Darbo lapas *Ledu furgonai*

---

Pažymėkite, kuriose sankryžose turėti stovėti ledų furgonai, kad būtų patogu visiems Turistų miesto gyventojams.



## Gudručiams

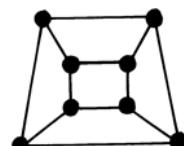
Realiamoje gyvenime „Ledų furgonų“ uždavinys gana dažnas, kai reikia nustatyti pašto punktų, gaisrinių ir kitų reikalingų objektų vietas, kad būtų galima greitai aptarnauti visus gyventojus. Tačiau realiamoje gyvenime negalima naudotis gudrybe, aprašyta šios veiklos skyriuje, ir lengvai išspręsti problemą.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

Galima pasitelkti visiškai nesudėtingą būdą: perrinkti visus jmanomus ledų furgono pastatymo variantus ir išrinkti geriausią. Kadangi Turistų mieste yra 26 gatvių sankryžos, tai yra ir 26 vienos 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 neberekia 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 jmanomas 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 tikrujų 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. atitinka visas sąlygas. Įdomu, kiek laiko galėtų sugaišti kompiuteris tikrindamas visas  $2^{26}$  galimybių. Jei patikrinti vieną galimybę, ar ji yra sprendinys, užtruktu 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. Mokiniamas pasakomas furgonų skaičius ir pasiūloma išsiaiškinti, ar jis yra mažiausias. Kai kuriais atvejais tai visai nesunku. Pavyzdžiu, grafe dešinėje vaizduojamas daug paprastesnis miestas. Jei gatvės įsivaizduojamos kaip kubo briau nos, 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.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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 viršinių 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 sprendžiamas 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 vadinais *minimalios dominuojančios sekos* (angl. *minimum dominating set*) uždavinių grupei, kurią sudaro tūkstančiai uždavinių, pavyzdžiui, žemėlapio spalvinimas, optimalaus maršruto paieška, tvarkaraščio sudarymas ir kt. Dar nėra žinoma, ar galima sukurti *polinominio laiko* algoritmus visiems šiemis 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 pilumo* uždaviniais. NP yra termino *nedeterministinis polinomas* santrumpa. Tai reiškia, kad uždavinys galėtų būti išspręstas per priimtiną 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 pilumo uždaviniai negali būti išspręsti per priimtiną 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 skirtini: žemėlapio spalvinimas skiriasi nuo ledų furgonų statymo ir t. t. Visi kartu šie uždaviniai gali būti išspręsti arba neišspėsti.

### Daugiau informacijos

Davidas Harelas knygoje „*Algorithmics*“ pateikia keletą NP pilumo uždavinių ir samprotauja, ar egzistuoja polinominio laiko algoritmas jiems spręsti. (*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).

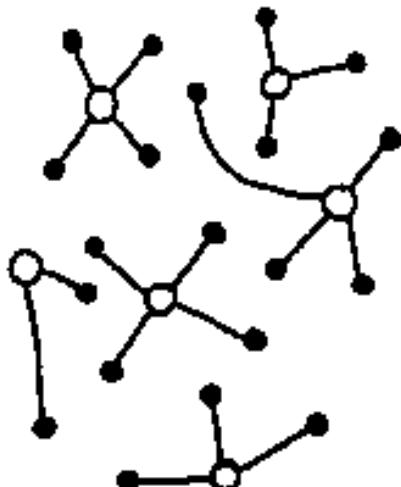
Michaelas R. Garey ir Davidas S. Johnsonas knygoje „*Computers and Intractability: A guide to the theory of NP-completeness*“ nagrinėja kelis šimtus NP pilumo uždavinių ir NP pilumo įrodymo metodų. (W. H. Freeman and co., New York, 1979).



## 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.



Laisvos viršūnės sujungiamos briaunomis arba lankais, kaip parodyta paveiksle.



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# 16 veikla

---

## Ledo keliai. Šteinerio medžiai

### Santrauka

Kartais tik šiek tiek pakeistas paprastas uždavinys tampa labai sunkiai sprendžiamas. Šiame skyriuje nagrinėjamas trumpiausio kelio paieškos uždavinys, panašiai kaip skyriuje „Purvinas miestas“. Tačiau šio skyriaus uždavinyje leidžiama pridėti naujų taškų, jei jie sutrumpina bendrą kelią. Uždavinys tampa daug sunkesnis ir nesusijęs su „Purvino miesto“ uždaviniu, bet algoritmavimo požiūriu atitinka „Žemėlapio spalvinimo“ ir „Ledų furgonų“ uždavinius.

### Ryšiai su ugdymo programomis

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

### Gebėjimai

- ✓ Vizualizavimo
- ✓ Geometrinio pagrindimo
- ✓ Algoritmavimo

### Amžius

- ✓ Nuo 7 metų

### Priemonės

Kiekvienai mokinį grupei reikia:

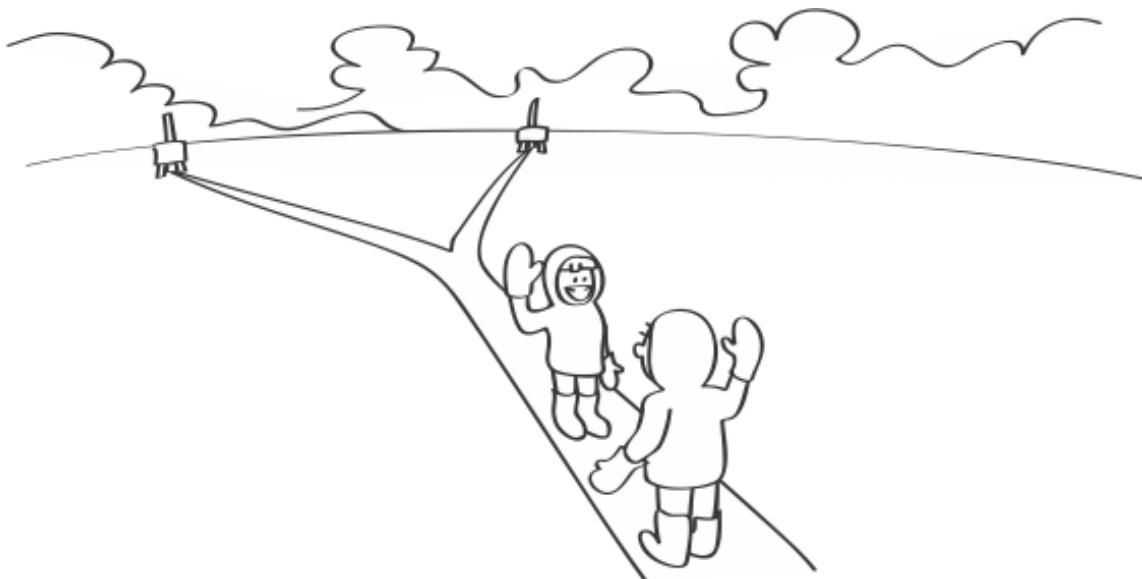
- ✓ Penkių ar šešių kuolelių (tinka palapinės kuoleliai)
- ✓ Kelių metrų virvės (mažiausiai 10 m)
- ✓ Ruletės
- ✓ Rašymo priemonės ir popieriaus lapo pastaboms



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

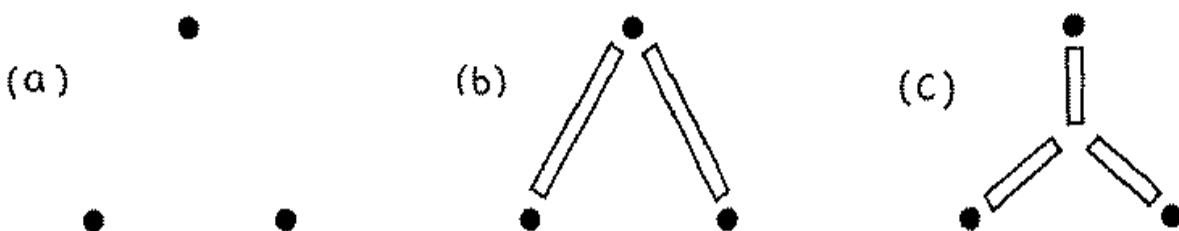
# Ledo kelias



## Ivadas

Šiaurinėje Kanados dalyje aplink didelį ežerą yra įrengti keletas gręžinių. Žiemą ežeras užšala, 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. Mokinių 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.

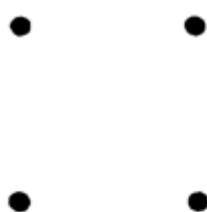


Pavyzdžiui, a paveiksle pavaizduoti trys gręžiniai. Šiuos taškus galima sujungti tiesiomis linijomis, kaip parodyta b paveiksle. Kitas jungimo būdas pavaizduotas c paveiksle. Matuojant nesunku išsiaiškinti, kad c paveiksle bendras kelių ilgis yra trumpiausias. Pridėtas papildomas taškas, sutrumpinantis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

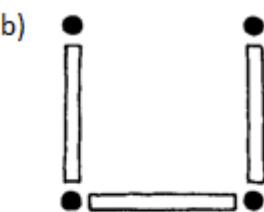
## Diskusija ir veiklos

- Užduotis aiškinama toliau paveiksle pateiktu pavyzdžiu.

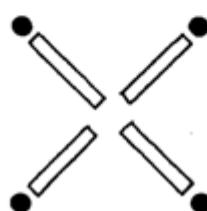
a)



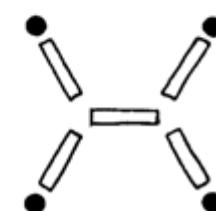
b)



c)



d)



- Užduotis išbandoma lauko aikštėje. Mokiniai aikštėje 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 būtų galima sujungti visus kuoliukus trumpesne virve?
- 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, panaudojama  $1 + \sqrt{3} = 2,73$  m virvės.
- Ar virvės reikėtų mažiau, jei rastumėte 3 Šteinerio taškus? (Ne. Du taškai yra geriausias variantas.)
- Diskutuojama su mokiniais, kodėl šis uždavinys yra sudėtingas. (Nežinoma, kaip nustatyti Šteinerio taškus, todėl tenka išbandyti daug variantų.)



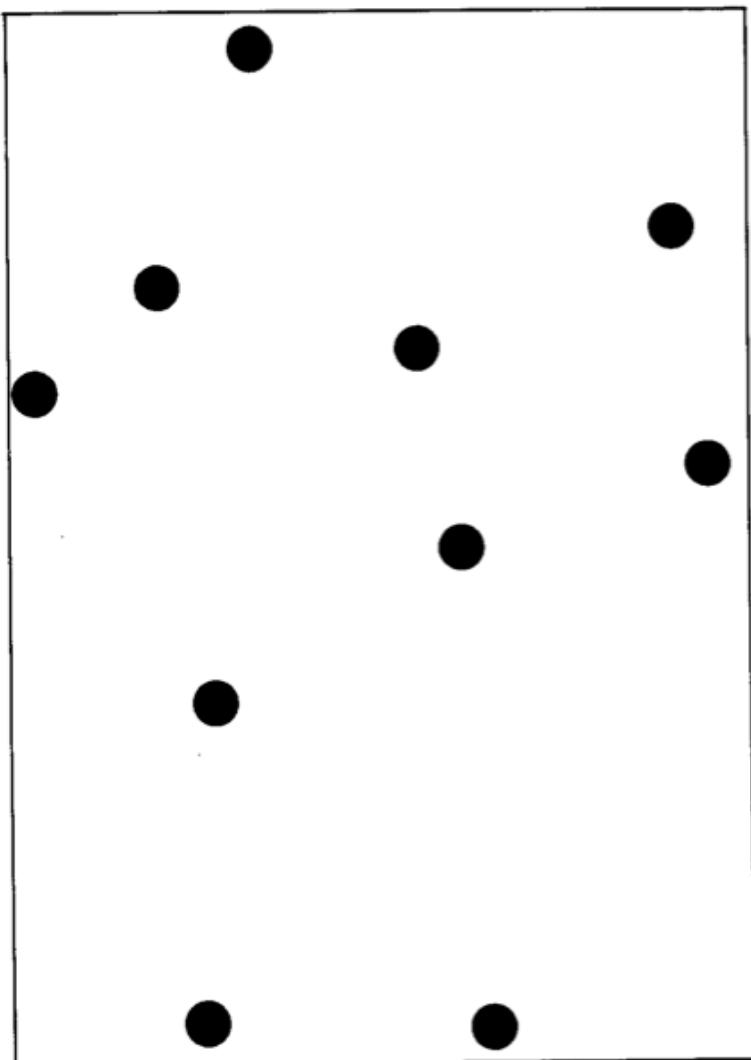
Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

153

## Darbo lapas *Šteinerio medis*

---

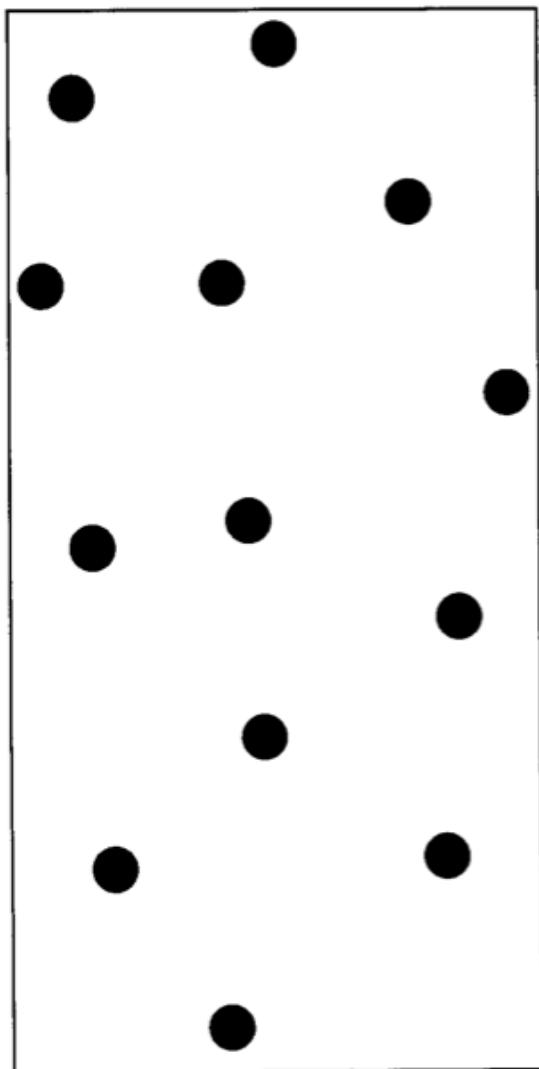


Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

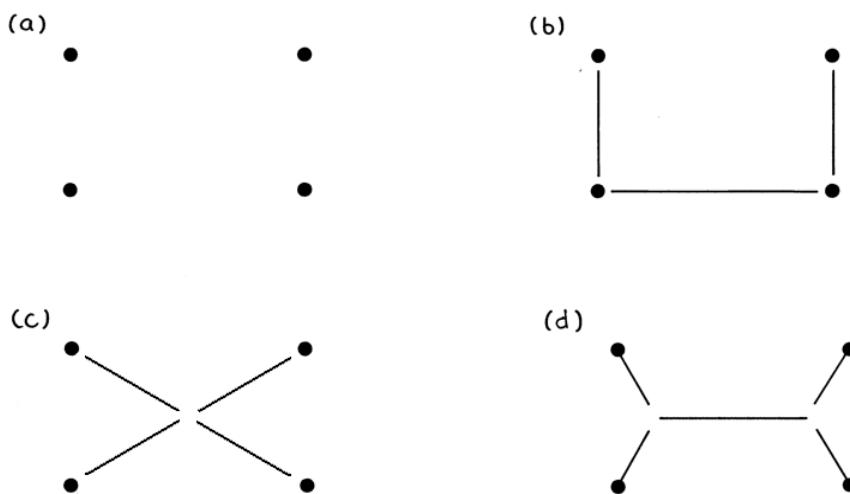
## Darbo lapas *Šteinerio medis*

---



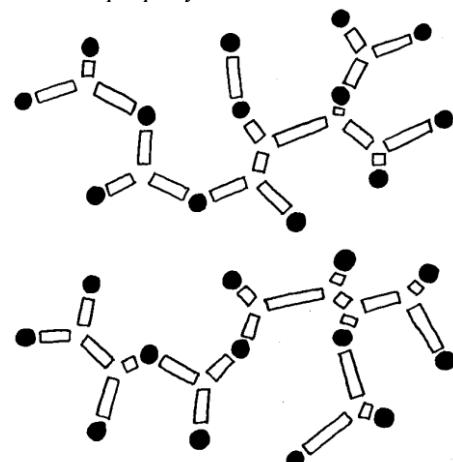
## Gudručiams

- Greitai pagrindinius uždavinius atlikę mokiniai gali bandyti rasti trumpiausią kelią stačiakampje  $1 \times 2\text{ m}$  (a paveikslas). Sujungus b variantu reikia  $4\text{ m}$  virvės. Suradus vieną Šteinerio tašką, reikia  $2\sqrt{5} = 4,47\text{ m}$  virvės (c paveikslas), kai turime du Šteinerio taškus, reikia  $2 + \sqrt{3} = 3,73\text{ m}$  virvės (d paveikslas). Mokiniai galėtų pagrįsti, kodėl pridėjus vieną papildomą tašką stačiakampje 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).

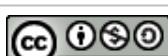
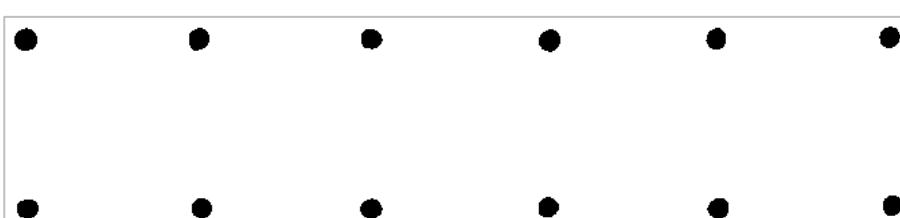


- Vyresnieji mokiniai gali nagrinėti uždavinį, kuriame yra daugiau taškų. Jie galėtų spręsti kitus darbo lapuose pateiktus uždavinius naudodami pieštukus ir trintukus arba naujas darbo lapų kopijas. Uždavinio sprendimą galima pavaizduoti ir aikštelėje, taškus sužymėti kuoliukais. Mokiniai galėtų diskutuoti tarpusavyje, kaip rasti geresnį sprendimą. (Pavyzdžių trumpiausi atstumai pavaizduoti paveiksluose dešinėje.) Pateikiami du panašūs Šteinerio medžiai rodo, kad uždavinys yra sunkiai išsprendžiamas. Čia galima pasirinkti gana daug Šteinerio taškų!

*Du galimi Šteinerio medžiai antrajam darbo lapo pavyzdžiui.*



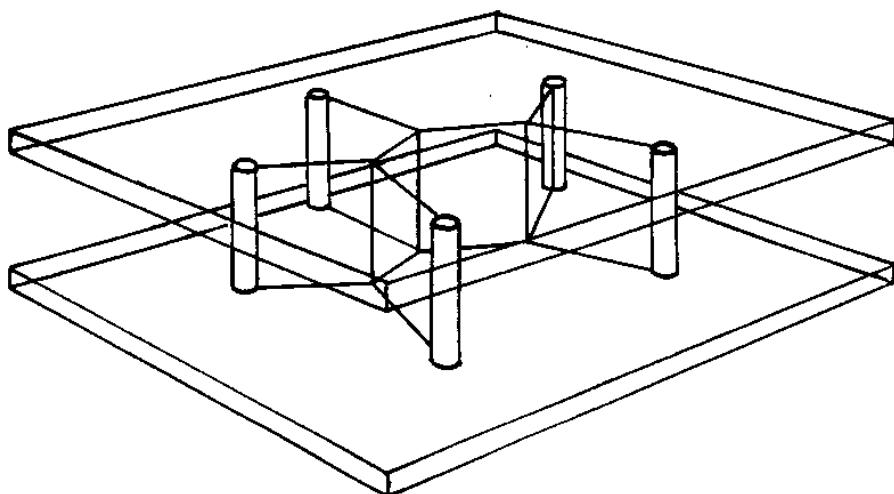
- Toliau pateikiamas pavyzdys virvinių kopėčių uždavinio, kurį galėtų spręsti mokiniai, baigę pagrindinius uždavinius:



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

- Dviejų laiptelių virvinių kopėčių uždavinys gali būti paverčiamas kvadrato uždaviniu. Trių laiptelių virvinių kopėčių uždavinio sprendimas yra visiškai kitoks. Jį galima pabandyti nubraižyti. Keturių laiptelių virvinių kopėčių uždavinį galima paversti dviejų sujungtų kvadratų uždaviniu. Penkių laiptelių virvinių kopėčių uždavinio 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, uždavinys sprendžiamas taip, tarsi tai būtų sujungti keli kvadratai, o jei nelyginis – sprendžiamas taip, tarsi būtų kartojamas trijų laiptelių virvinių kopėčių uždavinys. Tačiau šio sprendimo įrodymas nėra lengvas.
- Kita įdomi šių uždavinių interpretacija yra Šteinerio medžių muilo burbulai. Du permatomo plastiko lapai persmeigiami smeigėmis (ar virbalais), kaip parodyta toliau paveiksle:



Įmerkus visą konstrukciją į muilo tirpalą ir po to ištraukus, muilo burbulų plėvelę, 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 išdėstymentą šiame uždavinyje.



## Apie ką visa tai?

---

Šiame skyriuje nagrinėjami *minimalieji Šteinerio medžiai*. Jie vadinami *medžiais*, nes neturi ciklų, o tik šakas – kaip tikri medžiai; Šteinerio medžiai, nes jų originalū medžių būti įterpiama naujų Šteinerio taškų; *minimalka* medžiai, nes bendras tokio medžio briaunų ilgis yra mažiausias.

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

Minimalusis Šteinerio medis labai pravertė taupant lėšas telefono ryšio versle. Dar iki 1967 m. JAV vartotojų korporacija turėjo didelį privačių telefonų tinklą, kurio linijos buvo nuomojamos iš telefonų kompanijos. Mokesčius buvo skaičiuojamas ne pagal faktinį reikalingą laido ilgi, 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 medžį. Tačiau apie 1967 m. vartotojai pastebėjo, kad įterpus papildomą telefono linijos stotelę pakankamas atstumas iki vartotojo telefono sumažėja. Telefono kompanija buvo priversta sumažinti jkainius. 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 jungiančią taškų aibę.

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, be to, nėra žinoma, ar egzistuoja polinominio laiko algoritmas šiam uždaviniui spręsti. Žinoma tik tai, kad, jei būtų surastas polinominio laiko algoritmas šiam uždaviniui spręsti, 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 liepia sugalvoti efektyvų algoritmą uždaviniui optimaliai išspręsti, o to padaryti negalima? Pavyzdžiu, kai oro linijos susiduria su faktu, kad skrydžių kaina gali būti sumažinta į skrydžių linijų tinklą įterpiant Šteinerio taškus. Būtų puiku įrodyti, kad geresnio sprendimo nėra, bet įrodyti neigiamus



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

rezultatus labai sunku, ypač informatikoje. Sumanus programuotojas gali gudrauti. Tačiau negalima tvirtai pasakyti viršininkui, kad jo paliepimas neįvykdomas ir uždavinys sunkiai sprendžiamas. Vienap 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 nejmanomas.“

„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 *heuristiniai* algoritmai, kurie neužtikrina geriausio rezultato, bet randa artimą rezultatą. Heuristiniai 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.



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

159



Licensed under Creative Commons  
© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# 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 tinklą, kai pasaulį apraizgė saitynas, kai internetu kasdien siunčiama ir gaunama svarbios informacijos, kriptografija tapo reikalinga visiems. Ji nebéra vien mokslas apie šifru 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 nederėtu 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 tikrujų taip nėra. Šiuolaikinės kompiuterių sistemos nesaugo slaptažodžių, nes, jei saugotų, bet kuris prieigą prie slaptažodžių turintis asmuo galėtų išsibrauti į visas sistemas: galėtų daryti bankų pavedimus, siuštį 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 tokų 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 kiti 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 sunkus uždavinys gali būti išsprendžiamas taikant stulbinančią 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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

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 mokiniamams netikėtumo atmosferą, aptariant reikiamaą atlikti veiklą išlaikyti mokinį susidomėjimą, kad jie matytų ne vien smulkmenas, pavienius sunkumus, bet sugebėtų aprépti 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.

### Tikslių mąstantiems

Kompiuteriai jsiveržia į kasdienį gyvenimą, o kriptografijos taikymas tampa kasdiene tendencija. Dauguma žmonių nežino, ką gali šiuolaikinės kriptografijos protokolai. Didelės organizacijos (vyriausybinės ir komercinės) naudoja jvairias 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 dalyvauti priimant šiuos sprendimus.

Ši medžiaga apie informaciją slepiančius protokolus, kriptografijos protokolus ir viešojo rako kriptografiją yra gana sudėtinga. Tačiau pačios idėjos gana paprastos, nors techninius jų aspektus be gilių konceptų analizės ir sunku suvokti. Praktinėse situacijose, pavyzdžiui, užsiimant elektronine prekyba, techninės detalės paslepiamos kompiuterio programose, todėl galima lengvai taikyti naujas šifravimo technologijas. Tačiau vis tiek svarbu suprasti idėjas, kuriomis šios šifravimo technologijos grindžiamos, kad įgytume daugiau 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 (pavyzdžiui, narkotikų prekeivai, 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 diskusijų: vieniems žmonėms atrodo, kad reikia plėsti kriptografijos sistemų galimybes, o kiti jaučiasi suvaržyti, kai vyriausybės turi galimybę stebeti 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 surkurta 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



Iustas būtų naudojamas visuose komunikacijų prietaisuose, tačiau tai sulaukė piliečių pasipriešinimo dėl grėsmės privatumui. Taigi visas kriptografijos sistemos yra techniškai jmanomos, bet nebūtinai politiškai priimtinios.

Kriptografija taikoma daug plačiau, ne tik pranešimams šifruoti. Ji pasitelkiama nustatant asmens tapatybę, pranešimo siuntėją. Kai negalima nustatyti tapatybės, neįmanoma elektroninė komercija. Taip pat neįmanomas ir elektroninis balsavimas: žmonės gali balsuoti elektroniskai nebijdami, 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 pasitiketi? Kaip galima nustatyti, ar gautas laiškas yra originalus, o galbūt kas nors ji paémė, 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 vietas 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.

Keletas knygų apie šifravimą ir dešifravimą (šifru nulaužimą):

Francis Harry Hinsley ir Alan Stripp(eds) „Codebreakers: the inside story of Bletchley Park“ (Oxford University Press, 1993) – ši knyga pasakoja, kaip pirmaisiais kompiuteriais per Antrajį pasaulinį karą buvo iššifruojama informacija.

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 17 veikla

---

# Dalijimasis paslaptinis. 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: mokinį 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 mokinį grupei reikia:

- ✓ Užrašų knygelės ar sąsiuvinio
- ✓ Rašiklio



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

# Dalijimasis paslaptimis



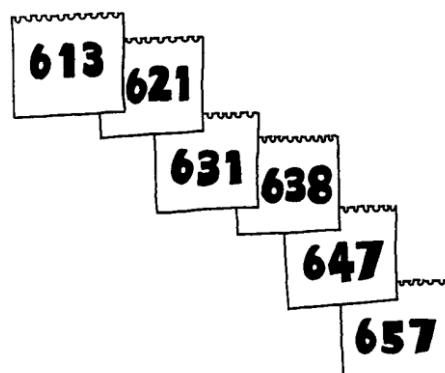
## Ivadas

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

Grupėse turi būti bent po 3 mokinius.

## Diskusija

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

6. Užrašų knygelė grąžinama pirmajam mokiniui. Mokinys atima savo sugalvotą skaičių iš užrašyto paskutinio skaičiaus. Pavyzdje 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 išplėštieji lapai su skaičiais.

### Gudručiams

Ši sistema gali būti taikoma slaptai 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 už balsavusių 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 jrašai ir pan. Šios informacijos privatumas labai svarbus, tačiau kai kuria informacija mums reikia dalytis su kitaais. Pavyzdžiui, parduotuvėje mokant už prekes banko kortele, pardavėjui reikia įsitikinti, kad pirkėjas turi lėšų savo banko sąskaitoje.

Dažnai pateikiama daugiau informacijos, negu iš tikrujų 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 paviešinti 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 perivedami 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

Davido 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 „Communications of the ACM“.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# 18 veikla

---

## Monetos metimas. Kriptografijos protokolai

### Santrauka

Šia veikla sprendžiamas paprastas, bet iš pirmo žvilgsnio atrodantis neįmanomas uždavinys. Du nepažintami 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 algebro taikymo
- ✓ Funkcijų taikymo
- ✓ Dėlionių sprendimo

### Amžius

- ✓ Nuo 9 metų

### Priemonės

Kiekvienai grupei reikia:

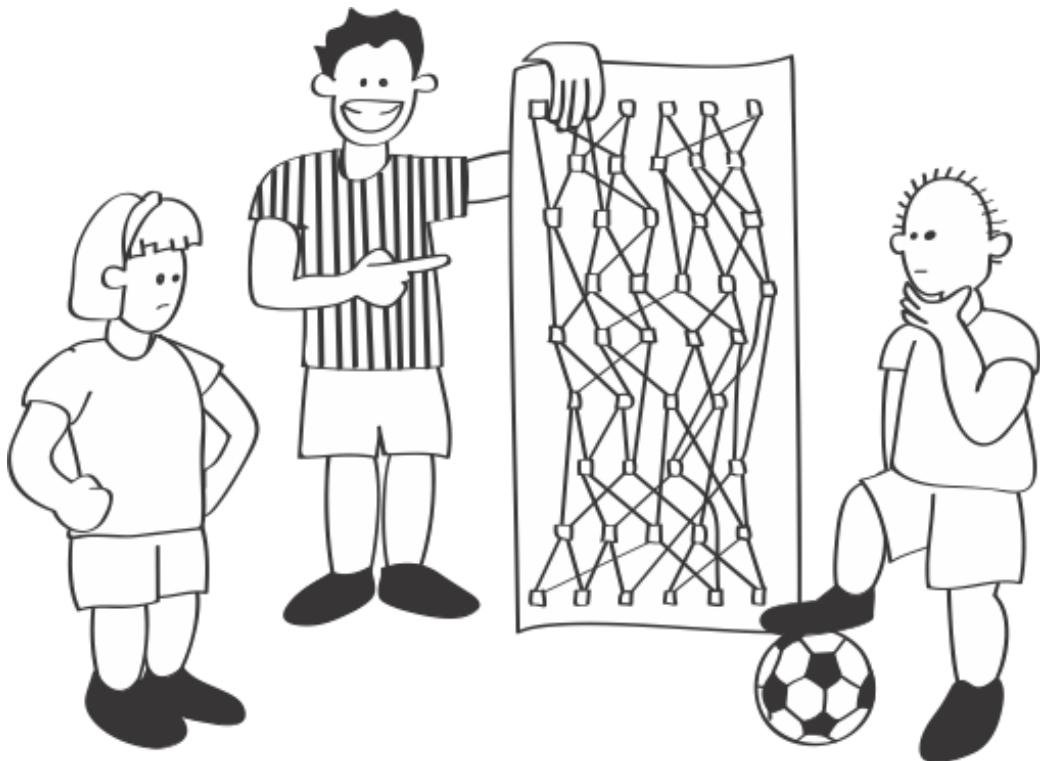
- ✓ Darbo lapo „Monetos metimas“
- ✓ Po pakelj dviejų skirtinį spalvų sagų ar figūrėlių



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

## Peru monetos metimas



### Ivadas

Šią veiklą sukūrė vienas Peru mokytojas, dirbdamas su mokiniais. Veiklos istorija 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 nuspresti – mesti monetą, bet abu miestus skiria didelis atstumas. Alisa, atstovaujanti Limai, ir Benitas, atstovaujanties 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 neapgaudinėja, 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: ji bus tokia sudėtinga, kad nė vienas negalės sučiauti. Galutinė schema yra abiem žinoma.

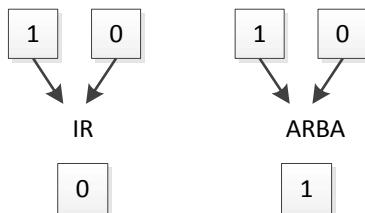


Licensed under Creative Commons

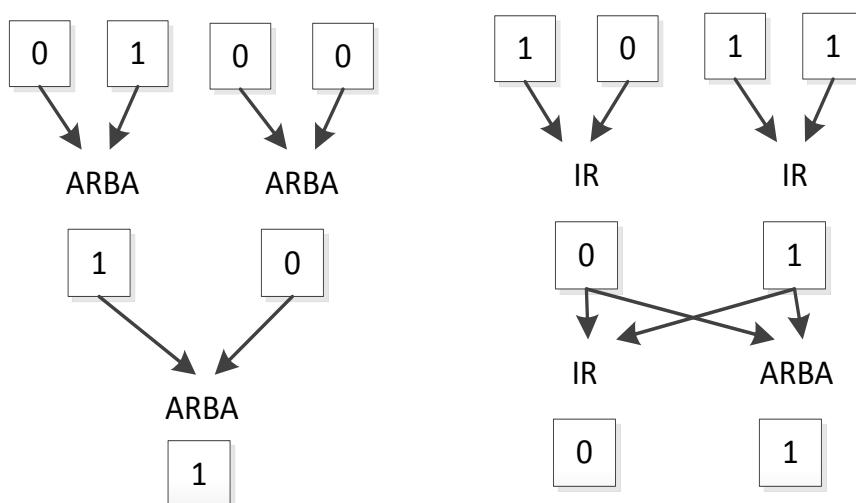
© 2015 Computer Science Unplugged (csunplugged.org)

## 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, jo išvedimas lygus 1. Loginio elemento ARBA išvedimas yra 0 („netiesa“), jei abu įvedimai yra 0, arba 1 („tiesa“) visais kitais atvejais. Pavyzdyme kairėje elemento ARBA išvedimas yra lygus 1, kai įvedimai yra 1 ir 0.

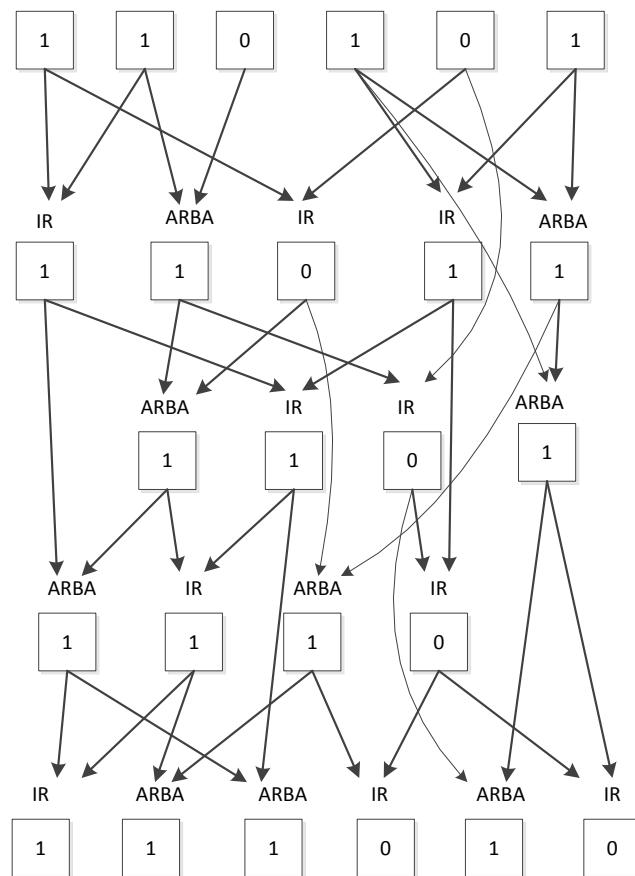


Vieno elemento išvedimas gali būti kito elemento (arba kelių elementų) įvedimas. Toliau esančiame pavyzdyme kairiosios schemas dviejų elementų ARBA išvedimas yra trečiojo elemento ARBA įvedimas. Tokios schemas rezultatas yra 1, jei bent vienas iš keturių pradinių įvedimų yra 1. Dešiniojoje schemaje aukštesniųjų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



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





Pagal šią schemą gali būti metama moneta telefonu toliau aprašytu būdu. Alisa pasirenka šešis pradinius jvedimus (nulius arba vienetus), kurių nesako Benitui. Sekdama schemas rodyklėmis atlieka numatytaus veiksmus ir gautus šešis išvedimus nusiunčia Benitui. Gavęs Alisos siūstus išvedimus Benitas turi atspėti, ar buvo jvestas lyginis, ar nelyginis skaičius vienetų, t. y. jis turi atspėti Alisos jvedimų lyginumą. Jei schema yra pakankamai sudėtinga, Benitas nesiaškina 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, vadinas, rungtynės vyksta Limoje, jei Benito spėjimas yra neteisingas. Kai Benitas pasako Alisai savo spėjimą, Alisa parodo savo jvedimus, kad Benitas galėtų įsitikinti, jog viskas buvo teisinga.

1. Mokiniai suskirstomi į grupeles. Kiekvienai grupelei duodamas darbo lapas, kuriame nubraižyta schema, keletas žaidimo figūrėlių ir paaškinamas uždavinys. Mokiniams bus jdomiau, 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 schemas jvedimų kvadrateliuose. Tada paaškinami elementų IR ir ARBA išvedimai, surašyti darbo lapo apačioje. Mokiniams galima pasiūlyti nuspalvinti schemų kvadratelius sutartinėmis spalvomis.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

3. Parodoma, kaip sekant schemos rodyklėmis ir atliekant nurodytus veiksmus dėliojamos figūrėlės ant kvadratelių ir šitaip apskaičiuojami išvedimų rezultatai. Viskas turi būti atlikta kruopščiai. Toliau lentelėje (mokiniams ji nerodoma) pateikiami visi jmanomi jvedimai ir jų galimi išvedimai, jei kam nors kiltų abejonių.

Jvedimas	000000	000001	000010	000011	000100	000101	000110	000111
Išvedimas	000000	010010	000000	010010	010010	010010	010010	010010
Jvedimas	001000	001001	001010	001011	001100	001101	001110	001111
Išvedimas	001010	011010	001010	011010	011010	011010	011010	011111
Jvedimas	010000	010001	010010	010011	010100	010101	010110	010111
Išvedimas	001000	011010	001010	011010	011010	011010	011010	011111
Jvedimas	011000	011001	011010	011011	011100	011101	011110	011111
Išvedimas	001010	011010	001010	011010	011010	011010	011010	011111
Jvedimas	100000	100001	100010	100011	100100	100101	100110	100111
Išvedimas	000000	010010	011000	011010	010010	010010	011010	011010
Jvedimas	101000	101001	101010	101011	101100	101101	101110	101111
Išvedimas	001010	011010	011010	011010	011010	011010	011010	011111
Jvedimas	110000	110001	110010	110011	110100	110101	110110	110111
Išvedimas	001000	011010	011010	011010	011010	111010	011010	111111
Jvedimas	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 pasidalinti pusiau – vieni atstovauja Alisai, o kiti – Benitui. Alisa pasirenka jvedimus ir apskaičiuoja išvedimus. Juos pasako Benitui, o šis bando atspėti jvedimo lyginumą (lyginis ar nelyginis vienetų skaičius buvo įvestas). Aišku, Benito atsakymas yra paprastas spėjimas. Alisa visiems pasako, kokie buvo jvedimai. Benitas laimi, jei atspėja lyginumą. Jis gali patikrinti pagal schemą, ar Alisa nepakeitė savo pasirinkimo.

Šiame etape monetos metimas baigiamas.

Benitas galėtų sukčiauti, jei rastų jvedimus, 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 jvedimai, tačiau labai sunku nuspėti jvedimus, kai žinomas rezultatas.



Alisa galėtų sukčiauti, jei rastų 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 jvedimų vienam rezultatui.

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

Pagal šią schemą Benitui sunku sukčiauti. Tačiau jei išvedimas būtų lygus 011000, tai jvedimas 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).

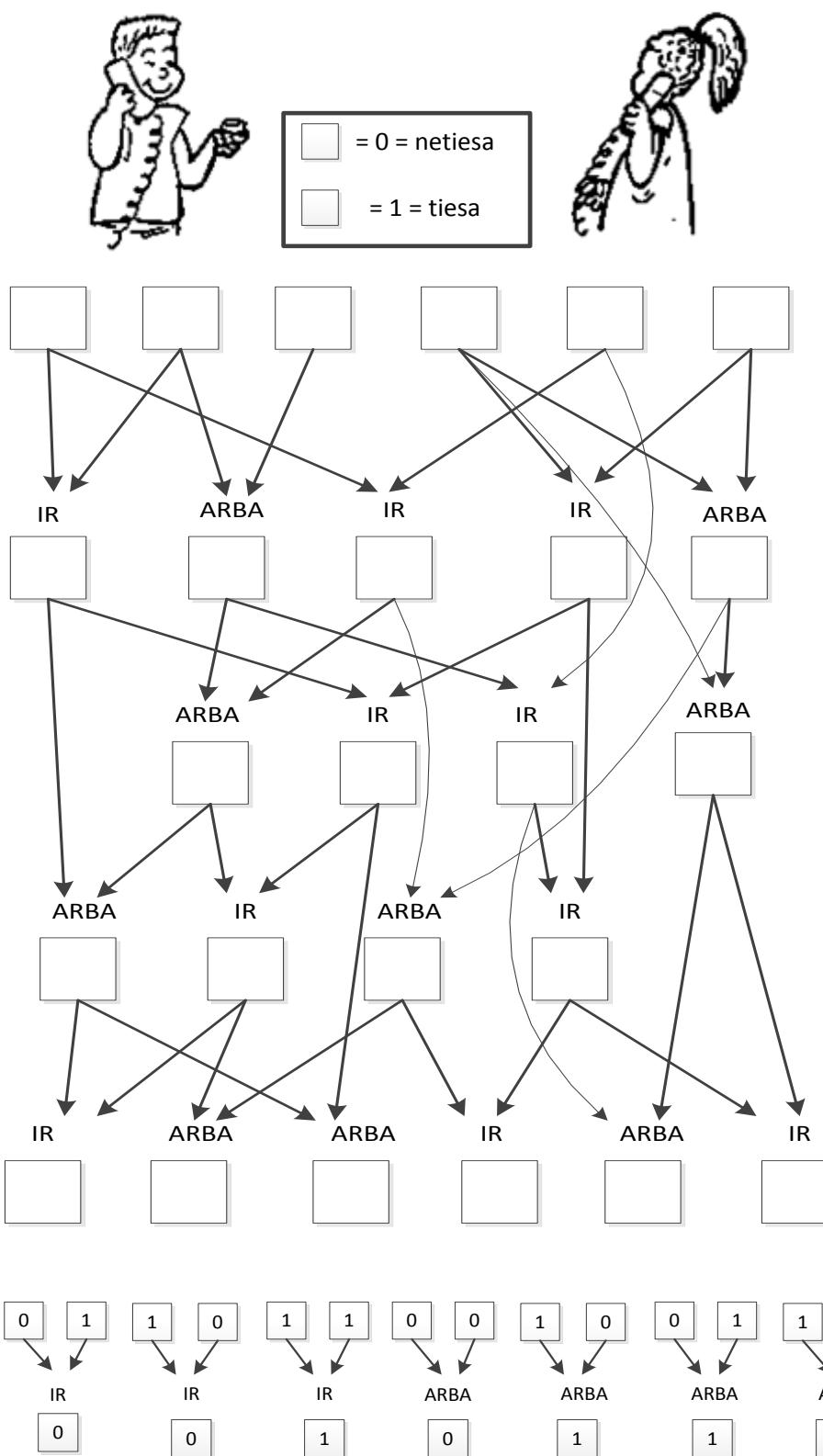
6. Mokiniai galėtų bandyti sukurti savo schemų šiam žaidimui. Galėtų sugalvoti schemą, kuri padėtų sukčiauti Alisai, ir schemą, kuri padėtų sukčiauti Benitui. Schemose nebūtinai turi būti šešios jvedimų pozicijos. Jose jvedimų ir išvedimų skaičius netgi gali skirtis.



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

## Darbo lapas Monetos metimas



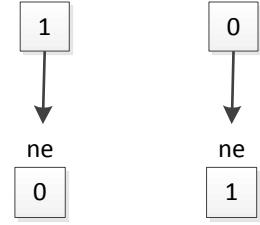
Pasirinkite įvedimus ir apskaičiuokite išvedimus pagal pateiktą schemą.



## Gudručiams

Pagrindinis uždavinys yra sukurti schemą, priimtiną ir Alisai, ir Benitui. Gana įdomu tokia veikla užsiimti telefonu, tačiau tai sudėtinga padaryti klaséje, todél galima alternatyva. Alisa ir Benitas atskirai sudaro savo schemas ir jas visiems parodo. Alisa atlieka abiejų schemų veiksmus su savo sugalvotais jvedimais 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 sukciauti. Jei viena iš schemų yra vienos krypties funkcija, tai abiejų schemų kombinacija taip pat yra vienos krypties funkcija.

Du tolesni uždaviniai nėra susiję su kriptografijos protokolais ar monetos metimu, jie skirti loginéms schemoms sudaryti atliekant IR ir ARBA operacijas. Šie uždaviniai 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.

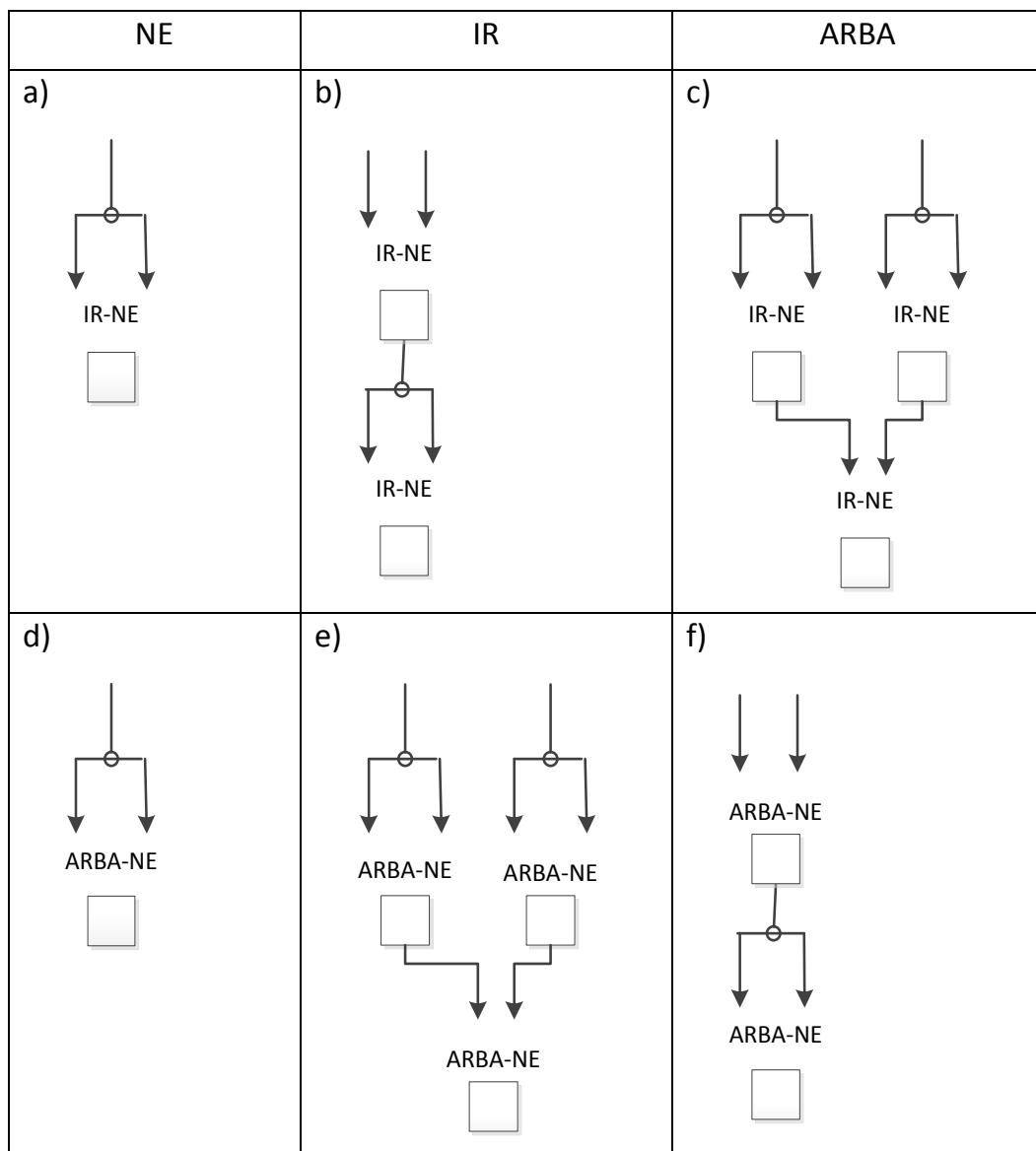
1. Mokiniai gali pastebeti, kad visų nulių jvedimas 000000 pateikia visų nulių išvedimą ir, priešingai, visų vienetų jvedimas 111111 pateikia visų vienetų išvedimą. (Aišku, ir kitų jvedimų variantai gali pateikti visų nulių ar visų vienetų išvedimus. Pavyzdžiui, pirmiau nagrinétoje schemaje jvedimas 000010 pateikia visų nulių išvedimą, o jvedimas 110111 pateikia visų vienetų išvedimą.) Toks rezultatas gaunamas todél, kad vartojami tik IR ir ARBA logikos elementai. Pridėjus dar vieną loginį elementą NE (vadinamajį neigimą), kurio išvedimas yra priešingas jvedimui (t. y. jei jvedamas 0, tai išvedimas yra 1, o jei jvedamas 1, tai išvedimas yra 0), galima sudaryti schema, kurios neatitinka visų nulių ir visų vienetų taisyklës.
2. Kiti du svarbūs loginiai elementai yra IR – NE ir ARBA – NE, kurių jvedimai atitinkamai pateikia priešingus negu elementai IR ir ARBA išvedimus. Tad *a* ARBA – NE *b* yra tas pats kaip NE (*a* IR *b*). Vartojant šiuos elementus 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 mokiniam 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. Toliau 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).



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)



## Apie ką visa tai?

---

Pastaraisiais metais labai išaugo elektroninė prekyba, todėl būtina užtikrinti piniginių operacijų ir dokumentų pasirašymo saugumą. Criptografijos 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 rako criptografija“ dabar plačiai naudojama kaip pagrindinis informacijos keitimosi saugumo būdas. Pavyzdžiu, naršyklėje galima pamatyti SSL (saugiuju jungimų lygmens) arba TLS (transporto lygmens saugumo) protokolus, kurie grindžiami viešojo rako sistemomis, leidžiančiomis interneto naršyklei saugiai prisijungti prie saityno sistemų (pvz., banko), net jei kas nors slapta seka visus siunčiamus duomenis.

Criptografija 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. Criptografijos protokolai kuriami tam, kad būtų galima naudotis, atrodo, neįmanomomis galimybėmis: skaitmeniniai parašais, sertifikatais ir pan.

Realybėje Alisa ir Benitas patys nekurtų loginės schemas, o naudotusi kompiuterio programa, kuri atliktu šį darbą. Tikriausiai nė vienas iš jų nesidomėtų, kaip sukurta ši programa, 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 (pavyzdžiu, 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 jvardija, 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. Patikimesnis 32 dvejetainių skaitmenų įvedimas, nors ir tai neapsaugo nuo sukčiavimo – viskas priklauso nuo sudarytos loginės schemas. Galimi ir kiti metodai, pavyzdžiu, 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 attiekama kompiuteriu tam tikromis programomis.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)



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

Kriptografiją galima pritaikyti ir žaidžiant pokerį telefonu, kai nėra arbitro, dalijančio kortas ir registrojančio 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 tikrujų 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 elektro-niniai 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 rako 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 rako 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.

Paprasčiau kalbant, kas nors nusiperka spyną, ant jos užrašo savo vardą ir padeda ant stalo, kad kiti galėtų ja naudotis. Raktą, žinoma, turi tik spynos savininkas. Tarkime, kad kas nors nori jums išsiųsti pranešimą, taigi jideda 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 rako 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 rako kriptografija, slaptas kodas

### Gebėjimai

- ✓ Ryšių nustatymo

### Amžius

- ✓ Nuo 11 metų

### Priemonės

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

Kiekvienai mokinį grupei reikia:

- ✓ Dviejų darbo lapų „Vaikų šnipų žemėlapiai“



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)



# Vaikai šnipai

## Ivadas

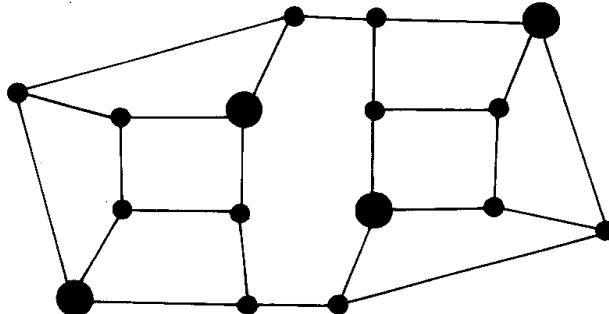
Ši veikla techniškai yra sudėtingiausia iš visų šioje knygoje aprašytų veiklų. Norint ją sėkmingai atlikti iki galo reikia kruopštaus darbo ir susikaupimo. Mokiniai turi būti susipažinę su vienos krypties funkcijomis (15 veikla „Turistų miestas“). Būtų lengviau atlikti šią veiklą, jei mokiniai atliktu 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 sakiny ar keletas sakinių, tačiau šiame uždavinyje Ema siūs tik vieną skaičių, kuris simbolizuojas vieną ženklu. Iš tikrųjų tai atrodo labai paprasta, turint galvoje, kad ji galėtų siųsti sakinį, sudarytą iš tokų ženklių, be to, tai bus padaryta dar ir kompiuteriu. Tačiau kartais net labai trumpi pranešimai yra labai svarbūs: vienas žinomiausių pranešimų istorijoje, siūstas Polo Reviro (Paul Revere), turėjo tik dvi galimas reikšmes. Emos pranešimas bus siunčiamas Bilo viešuoju raktu, todėl, net jei patektų į kitas rankas, jo nebūtų galima iššifruoti. Tik Bilas gali iššifruoti Emos pranešimą, nes tik jis turi raktą nuo „spynos“.

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

## Diskusija

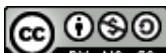
Darbo lape „Vaikų šnipų pranešimo užšifravimas“ vaizduojamas Bilo viešasis žemėlapis. Jis nėra slaptas, Bilas jį pateikia viešai (ant stalo ar tinklalapyje), kad kiekvienas galėtų jį pamatyti, arba duoda jį tam žmogui, kuris nori jam išsiusti pranešimą. Ema taip pat turi šio žemėlapio kopiją.



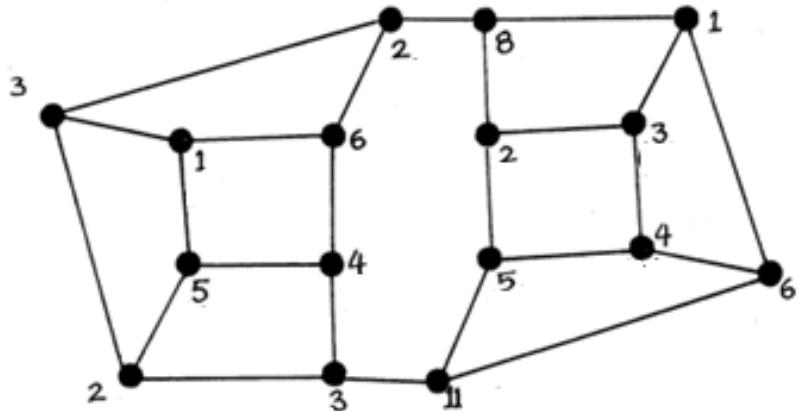
Paveiksle dešinėje yra Bilo privatusis žemėlapis. Jis panašus į viešąjį, tik kai kurie taškai paryškinti. Šis žemėlapis yra saugomas, kad niekas jo nematytu.

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 motyvuojatai, 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

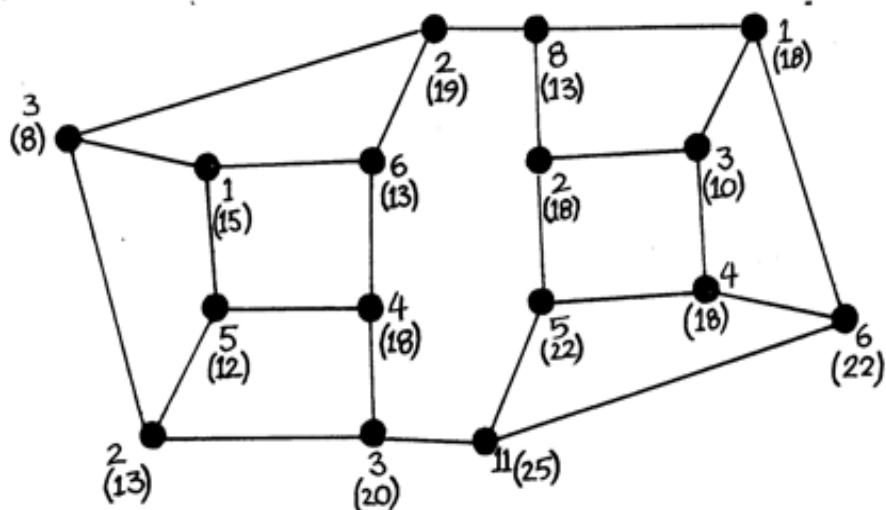


gatvių sankryžos žemėlapyje surašomi atsitiktiniai skaičiai, kurių suma lygi siunčiamam Emos skaičiui. Toliau pavyzdyme Ema sugalvojo siuštį 66, todėl prie gatvių sankryžų surašyti galimi skaičiai, kurių suma yra 66. Prireikus galimi ir neigiami skaičiai.



2. Toliau Ema turi nuspręsti, ką nusiųsti Bilui. Žemėlapio 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, toliau pateikto viešojo žemėlapio 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.



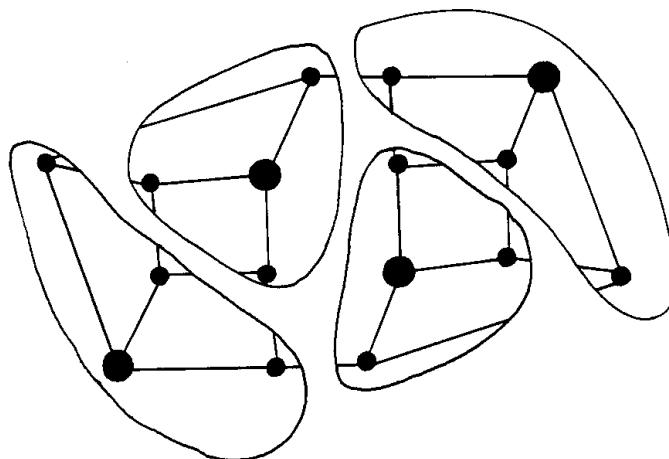
3. Ema nusiunčia Bilui žemėlapį, kuriame prie gatvių sankryžų surašyti tik skaičiai skliaustuose.

Prieš siunčiant Bilui galima nutrinti ne skliaustuose esančius skaičius arba perrašyti skaičius skliaustuose į žemėlapio kopiją. Tada galima paprašyti mokinį 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. Pavyzdyme Bilas sudeda skaičius 13, 13, 22 ir 18. Suma yra 66 – Emos siūstas pranešimas.

5. Kaip tai veikia? Viskas priklauso nuo žemėlapio. 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 toliau paveiksle. Tokiu būdu žemėlapis padalijamas į nesuskertančias 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: persiūstas visiškai slaptas pranešimas viešuoju raktu, be jokio išankstinio dalyvių susitarimo. Dabar galima skelbti savo viešajį raktą skelbimų lentoje (ar kur kitur) ir bet kas gali siūsti slaptas žinutes, bet niekas negali jų iššifruoti be privačiojo rakto. Realiame gyvenime visus skaičiavimus atlieka programinės įrangos paketas, paprastai integruotas į interneto naršyklę, todėl sunkiai dirba tik kompiuteris.

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

Viešojo žemėlapio sudarymas yra labai panašus į ledų furgonų sustatymą Turistų mieste (15 veikla). Žemėlapį nesunku sudaryti, jei pradedama nuo sričių braižymo privačiajame 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 žemėlapio su 50 ar 100 sankryžų. Regis,



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

niekas negalėtų jveikti tokio šifro, net protingiausi matematikai, kaip ir „Turistų miesto“ uždavinyje įrodyti, kad parinktas furgonų skaičius yra mažiausias.

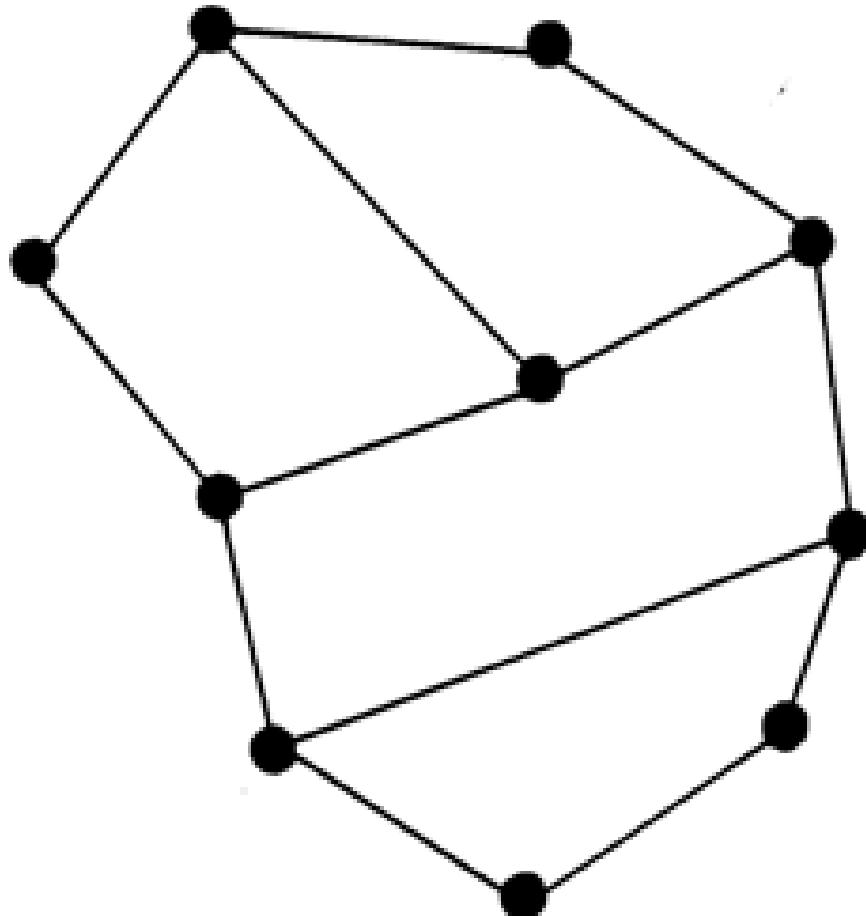
6. Su visa klase išnagrinėjus vieną pavyzdį mokiniai padalijami į grupes po 4. Kiekvienai grupės porai duodamas darbo lapas, kuriame pateiktas viešasis žemėlapis. Kiekviена pora susigalvoja siunčiamą „pranešimą“ (tai gali būti bet koks sveikasis skaičius), užšifruoja jį ir perduoda savo rezultatą kitai grupės porai. Šie bando iššifruoti, bet, aišku, jiems prireikia privačiuų žemėlapių. Ar pavyksta mokiniams iššifruoti pranešimus su privačiuoju raktu?
7. Kiekvienna pora gali pabandyti sukurti savo žemėlapį. Privatujį žemėlapį jie saugo, o viešajį duoda kitai grupės porai arba pakabina klasės lentoje. Žemėlapio kūrimo metodika panaši kaip ir ledų furgonų – sužymimi taškai ir tada sujungiami gatvėmis. Privačiajame žemėlapyje nubraižomos sritys, kuriose pažymimas vienas taškas ir iš jo išvedamos dvi ar trys gatvės, 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.



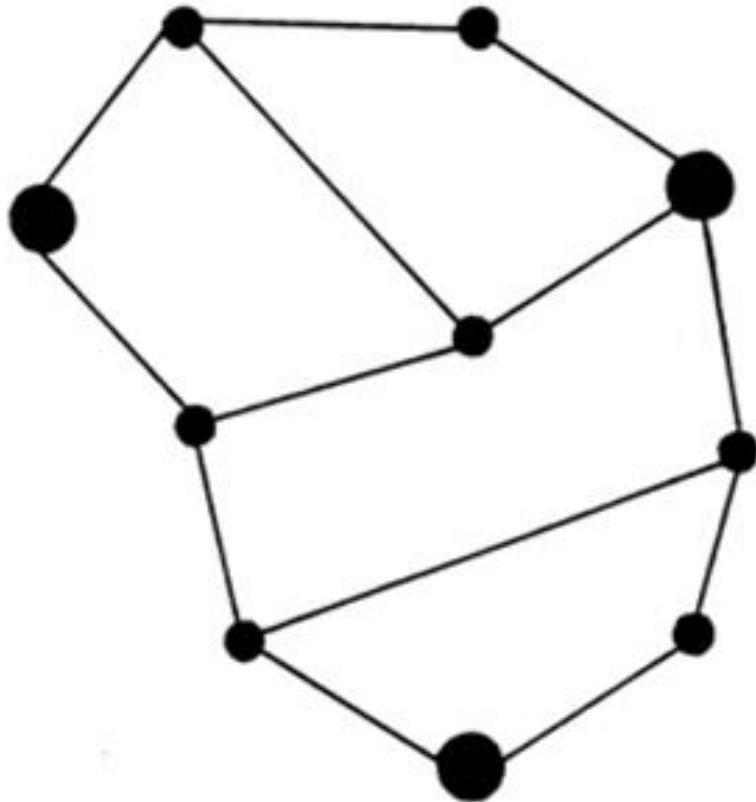
Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## **Darbo lapas Vaikų šnipų privatusis žemėlapis**

---

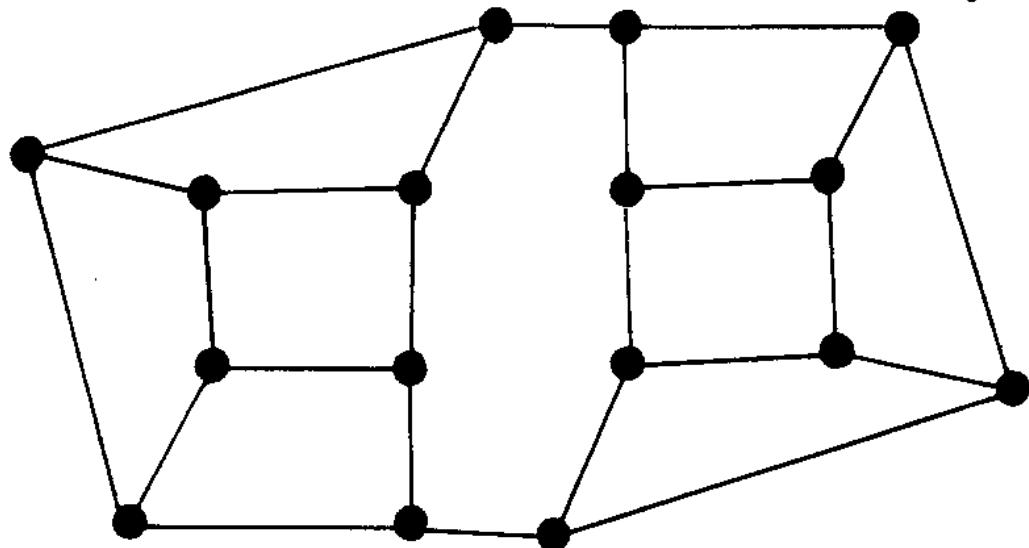
Naudodamiesi šiuo žemėlapiu iššifruokite pranešimus.



## Vaikų šnipų pranešimo užšifravimas

---

Šiuo žemėlapiu rodoma, kaip užšifruojamas pranešimas.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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 pasiėmės 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? Šis vaizduokime, Bilas siunčia elektroninį laišką: „Brangioji, aš nebeturiu pinigų ir negaliu grąžti. Prašau pversti 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 viešajį žemėlapį ir perskaičiuotus skaičius  $t_1, t_2, t_3, \dots$ , gali išreikšti visus perskaičiuotus skaičius kintamaisiais, vaizduojančiais pradinius skaičius. Gautą lygybių sistemą galima išspręsti lygčių sprendimo programa. Nustatomi pradiniai skaičiai, apskaičiuojama jų suma ir gaunamas siūstas skaičius (pranešimas). Taigi nereikia atkurti privačiojo žemėlapio. Ši lygčių sistema sprendžiama Gauso eliminavimo metodu, todėl skaičiavimo laikas yra proporcingsas lygčių skaičiaus kubui. Tačiau, kadangi dauguma šių lygčių 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 raktų kriptosistemomis, yra labai panašūs į tuos, kurie buvo atliekami šioje veikloje. Tik realios viešojo raktų kriptosistemos šifruoja jvairiais ir daug sudėtingesniais metodais, kuriais šifruoti rankiniu būdu net neįmanoma. Vienas iš saugiausių viešojo raktų metodų, naudojamų šiuolaikinėse šifravimo sistemose, yra grindžiamas sudėtingu dideliu skaičiu daugikliu skaičiavimu.

Kokie yra 100 ženklių 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

108.488.104.853.637.470.612.961.399.842.972.948.409.834.611.525.790.577.216.75  
3.

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

Dabar realioje viešojo raktų kriptosistemoje Bilas gali naudoti iš 100 skaitmenų sudarytą skaičių kaip viešajį raktą ir du jo daugiklius – kaip privatujį raktą. Sukurti tokius raktus nėra sunku: reikia tik būdo pirminiams skaičiams rasti. Taigi randami du gana dideli 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 išspėti privačiojo raktą, nebent galėtų prieiti prie superkompiutero ir turėtų keletą mėnesių laiko. O jeigu vietoj 100 skaitmenų viešasis raktas būtų sudarytas iš 200 skaitmenų, daugiklių radimas užtruktų metų metus. Tokio raktų nulaužimo sąnaudos yra daug didesnės, negu informacijos, kuri juo būtų iššifruota, vertė. Praktiškai 512 bitų ar didesni raktai, kurie yra ekvivalentūs apie 155 skaitmenų po kablelio ar daugiau, dažnai sudaromi saugiems ryšiams užtikrinti.

Vis dar nėra būdo, kaip užšifruoti pranešimus pirminiais skaičiais grindžiamu viešuoju raktu, kad jo nebūtų galima iššifruoti be privačiojo raktų. 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 grindžiamą sistemą? 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 nejrodyta). Vienaip ar kitaip, reikia būti budriems: jei



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

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

Nerimą kelia tai, kad, jei yra tik keli galimi pranešimai, piktavalis 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 pridedamų 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 uždaviniais. Niekam nepavyko sukurti sprendimo metodo, bet taip pat niekas nejrodė, 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 išsprendžiamas, būtų galima išspręsti juos visus. Kadangi tiek daug (nesėkmingų) pastangų jidėta ieškant efektyvaus NP sudėtingumo uždavinijų sprendimo algoritmo, tai skaidymo daugikliais uždaviniai puikiai tiktų kuriant saugias kriptosistemas. Šiandien kriptografija yra aktyvių 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.



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

191



# 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 privelia absurdų 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, vadina vartotojo sasaja. Tai svarbiausia dalis! Galima pamatyti, kad svarbiau yra tai, ką programa gali daryti, o vartotojo sasaja tik leidžia ta programa naudotis. Taip nėra, nes, jei negali su programa „susikalbėti“, tai negali ja ir naudotis. Vartojo sasają labai sunku suprojektuoti ir sukurti, todėl jos kūrimas reikalauja daug daugiau pastangų nei kitų programos dalių. Yra programų, turinčių aiškiai suprantamą ir patogią vartotojo sasają, bet ji tampa nepastebima, kai visas dėmesys skiriamas aplinkai suvokti. Labai daug ištisies gerų programų nenaudojamos vien dėl to, kad jų sasaja keista ir sunkiai suprantama. Visi kompiuterių programinės įrangos gamintojai stengiasi sukurti „protinę“ vartotojo sasają, padedančią naudotis visomis kompiuterio galimybėmis, kurios dažnai yra visiškai elementarios.

Kam reikalinga vartotojo sasaja? Kodėl negalima tiesiog pasakyti kompiuterui 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 sasajos kūrimo problemas, kompiuterio ribotumą ir sužinoti apie mitus, kuriais dažnai naudojasi dirbantieji kompiuterių pramonėje.

## Mokytojams

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

Vartotojo sasaja 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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

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

Šiame skyriuje aprašomos dvi veiklos. Pirmoji – apie kompiuterio programinę sąsają. Mokiniam 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 tiriamaja 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 tiria šiuos dalykus, o kompiuterio programų sąsajų projektavimo tyrimai tampa svarbia besiplėtojančia informatikos mokslo sritimi.

Dirbtinis intelektas visada sukelia labai daug diskusijų. Šioje knygoje pateikiama 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 – padrašinti 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 projektais rengiami uždarose 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 mokinius suvokti, kaip projektuojama žmogui lengvai suprantama vartotojo sąsaja. Kadangi visuomenėje yra paplitęs „skurdus“ dizainas, užuot jvardijus būtent netinkamo dizaino problemą, jprasta sau prisiimti „kaltę“ dėl negebėjimo tinkamai naudotis vienu ar kitu įrenginiu, aparatu ar programa („žmogiška klaida“, „nepakankamas mokymas“, „tai pernelyg sudėtinga man“). 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 reikia:

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# Šokolado gamykla

---

## Ivadas

Įdomų šokolado fabriką sukūrė panašių į elfus būtybių umpa lumpų<sup>4</sup> gentis. 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

Mokiniams 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 atveriant duris reikia stumti, ar traukti. Paprastai šios problemos sprendimo rezultatas – dviejų nešančiųjų susidūrimas ir šokolado ištaškymas.

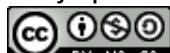
Mokiniai turi užpildyti darbo lapą „Kaip atverti duris?“ Kiekvienna grupė gali pažymėti daugiau kaip vieną variantą, ką daryti su kiekvienomis durimis. Nėra akivaizdu, kaip atidaryti kai kurias duris (jskaitant ir pirmąsias). Mokiniai turi parašyti, ką jie pabandytų pirmiausia. Kai jie užpildo savo lapus, 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 nusprenčia, kokios durys ir rankenos patogiausios fabrike.

Toliau pateiktoje lentelėje apibūdinamos kiekvienos durys. Tikros durys gali būti atveriamos į išorę arba į vidų. Mokiniai aptaria, kokios durys yra jų mokykloje. Ar yra tokiai, 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*), kai nusakomas objekto ypatybės ar funkcijos, t. y. kokios galimybės suteikiamos šiuo objektu ar kas leidžiama juo padaryti. Pavyzdžiui, daugumai aišku, kad kėdė skirta sėdēti, stalas – daiktams padėti, mygtuką reikia paspausti ir pan. Kompiuterio sąsajos įgalinimas yra mygtukai, tekstų laukai, meniu ir t. t., kurie suteikia vartotojui informacijos, kam yra reikalingi. Jei mygtukas būtų panašus į ką nors kita, nebūtų aišku, ką su juo daryti. Tai atrodo akivaizdu, bet netinkamų objektų vaizdavimo pavyzdžių programų sąsajose yra gausybė.

---

<sup>4</sup>Idėja paimta iš Roaldo Dahlio knygos „Čarlis ir šokolado fabrikas“.



Švaistinės durys	Paprastai stumiamos.	Durys su užrašu	Trumpai užrašyta 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 išdarkomas.

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 antraja iš kairės rankenėle; dešinioji viršutinė kaitvietė – trečia iš dešinės rankenėle, o apatinė dešinioji kaitvietė – paskutine, ketvirtąja, iš kairės rankenėle. Virdami šokoladą umpa lumpai visada klysta. Jie nuolat verda šokoladą nustatę netinkamą temperatūrą, apdegina savo rankoves siekdamai puodo su šokoladu.

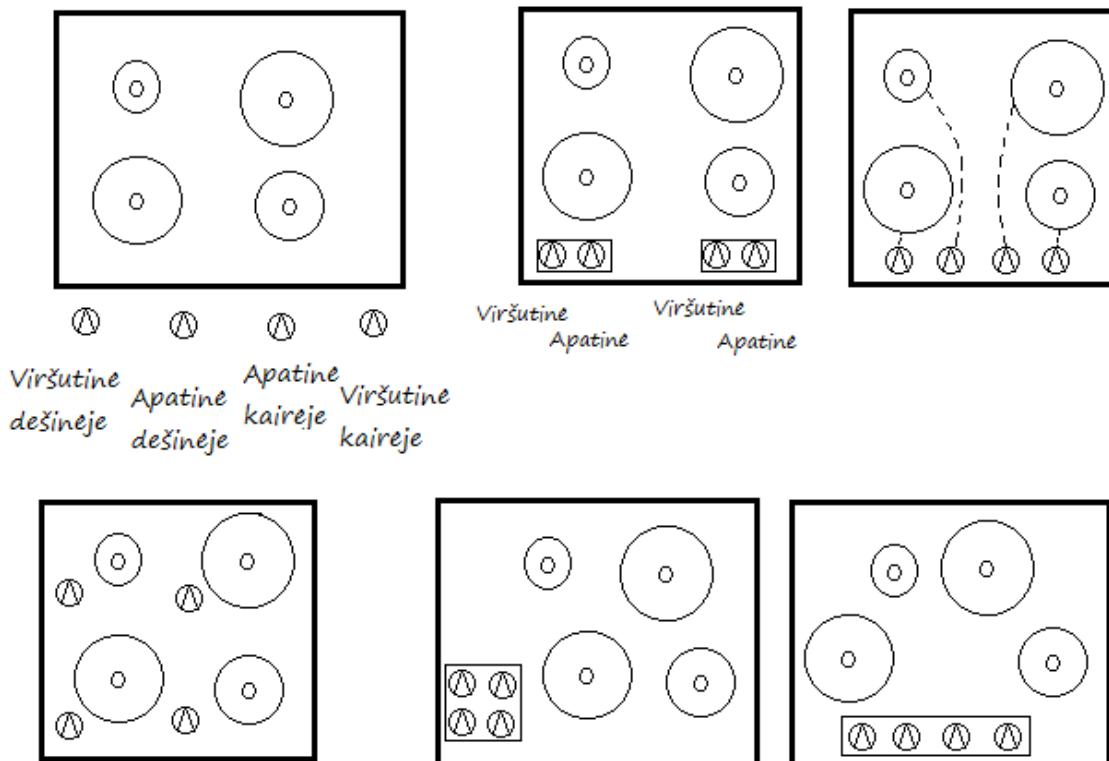
Mokiniai turėtų prisiminti, kaip išdėstyti viryklių valdymo rankenėlės ar jungikliai jų namuose, ir pasiūlyti, kaip geriausiai jas ar juos išdėstyti naujo fabriko viryklėse.

Ši užduoties dalis atliekama diskutuojant su visa klase. Toliau pateikti šeši paveikslai, kuriuose pavaizduota, kaip galima išdėstyti viryklės kaitviečių jungiklius. Visuose paveiksluose, išskyrus apatinį kairįjį, kuriame viryklės jungikliai yra prie kiekvienos kaitvietės, visi jungikliai yra priekyje. Viršutiniame kairiajame paveiksle kaitvietes įjungti viryklės jungikliais galima net 24 būdais, todėl prie jungiklių reikalingi



Licensed under Creative Commons

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 – taip tikrai geriau umpa lumpams. Apatiniuose paveiksluose 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 paaškinta (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 kambarje visada turi žinoti ir pasakyti kitiems, kada juostą sustabdyti, sulėtinti ar vėl paleisti. Anksčiau ši informacija buvo perduodama balsu: valdytojas iš valdymo kambario per garsiakalbj sakydavo, ką daryti. Tačiau fabrike visada labai triukšminga ir sunku išgirsti, kas sakoma.

Mokiniių 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!“, „Lėciau!“, „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 tuoju 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 tiktis ir kitoje. Mokiniai pritaiko gerai žinomą informaciją (šviesoforus) naujomis sąlygomis (sukeičia spalvų reikšmes). Šios veiklos situacijos gali atrodyti dirbtinės, tačiau pasaulyje pilna pavyzdžių, kai skirtinges visuomenės turi skirtinges taisykles. Pavyzdžiu, eismas Lietuvoje vyksta dešiniaja kelio puse, o Jungtinėje Karalystėje – kairiaja; 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.

Mokiniių grupės turėtų sugalvoti lengvesnį sprendimą, kaip umpa lumpams priminti, kur kiekvieno daikto vieta.

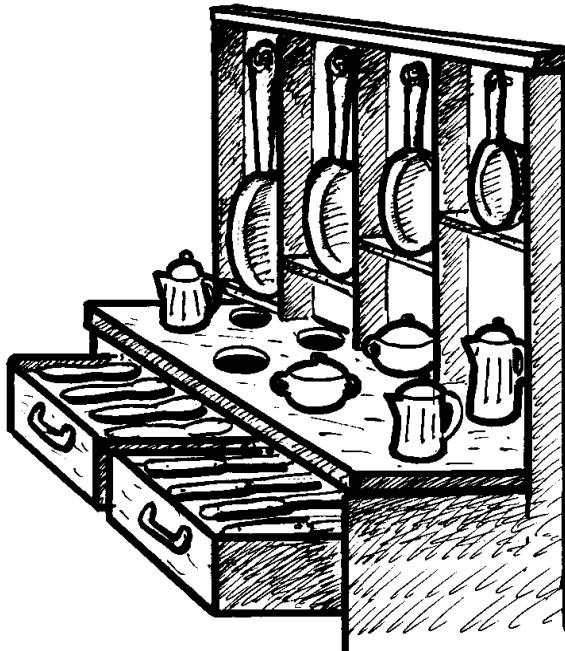
Paveiksle dešinėje pavaizduota, kaip turėtų būti sudėlioti daiktai (kartais jie taip ir būna sudėti).

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

5. Anksčiau minėtame valdymo kambarioje 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 susipažsta atlikdami darbo lapo „Piktogramos“ užduotis. Mokiniai turi nustatyti, ką reiškia pavaizduotos piktogramos. Pavyzdžiu, 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 mokiniių grupė gauna vieną arba dvi kortelles. Valdymo skyde esančios piktogramos reiškia 4–5 operacijas. Nieko neaiškindamos grupės parodo savo piktogramas kitoms grupėms ir paprašo paaiškinti, ką tos piktogramos galėtų reikšti. Padėkite 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

### Švaistines durys



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

### Durys su rankena



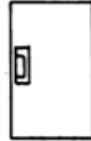
- Stumti  į kairę
- Traukti  į dešinę
- Stumti į šoną

### Varstomosios durys



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

### Stumdomosios durys



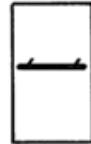
- Stumti  į kairę
- Traukti  į dešinę
- Stumti į šoną

### Durys su užrašu



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

### Baro durys



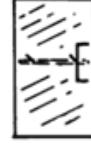
- Stumti  į kairę
- Traukti  į dešinę
- Stumti į šoną

### Durys su apvalia rankena



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

### Stiklinės durys



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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

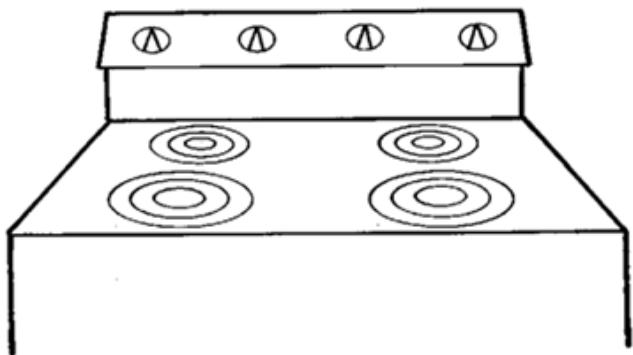
201

## Darbo lapas *Viryklė*

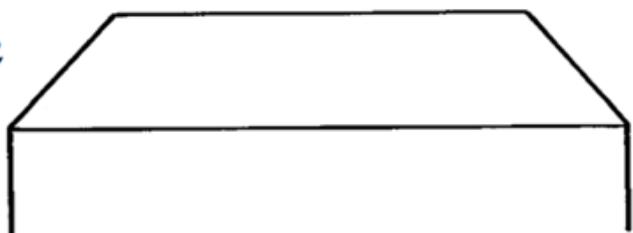
---

Išdėstykite kaitvietes ir jų jungiklius taip, kad umpa lumpams būtų patogu.

*Senoji  
viryklė*



*Naujoji  
viryklė*



Licensed under Creative Commons

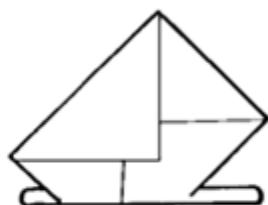
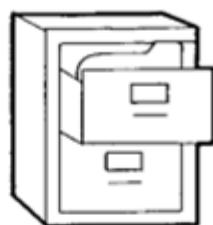
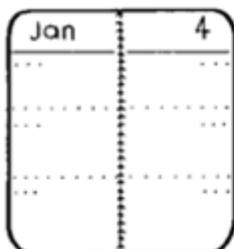
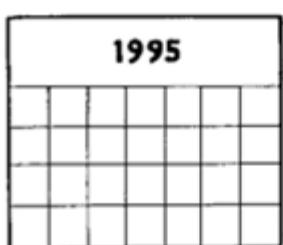
© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

## Darbo lapas *Piktogrammos*

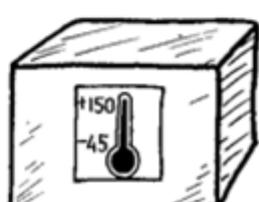
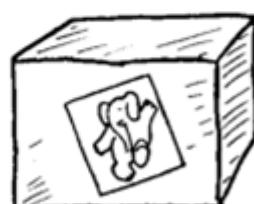
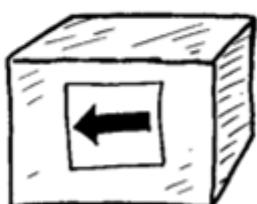
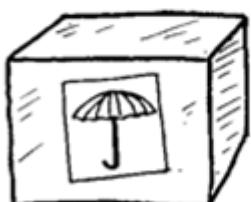
---

Kaip manote, ką vaizduoja šios piktogrammos?

*Biure...*



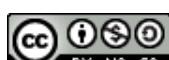
*Ant dežės...*



## Darbo lapas *Piktogramų kortelės*

Kiekvienai grupei duodama po vieną arba dvi kortèles. Grupė sugalvoja ir nupiešia piktogramą, atitinkančią kortelę.

Sudedamosios dalys	Priedai
<ul style="list-style-type: none"><li>• Kakava.</li><li>• Pienas.</li><li>• Cukrus.</li><li>• Papildomas cukrus.</li><li>• Sviestas.</li></ul>	<ul style="list-style-type: none"><li>• Riešutai.</li><li>• Karamelė.</li><li>• Imbieras.</li><li>• Razinos.</li><li>• Kokoso riešutai.</li></ul>
Gaminimas	Ragavimas
<ul style="list-style-type: none"><li>• Pradėti maišyti.</li><li>• Nustoti maišyti.</li><li>• Pradėti šildyti.</li><li>• Nustoti šildyti.</li><li>• Supilstyti į formas.</li><li>• Spausti antspaudą (daug įvairių).</li></ul>	<ul style="list-style-type: none"><li>• Paragauti ji.</li><li>• Nuostabus! – Pirmoji vieta.</li><li>• Geras. – Antroji vieta.</li><li>• Ech! – Perdaryti.</li><li>• Ech! Ech! – Išmesti.</li></ul>
Pjaustymas	Pakavimas
<ul style="list-style-type: none"><li>• Maža plytelė.</li><li>• Vidutinė plytelė.</li><li>• Didelė plytelė.</li><li>• Milžiniška plytelė.</li><li>• Nurodyti plytelės dydį...</li><li>• Šokolado drožlės.</li></ul>	<ul style="list-style-type: none"><li>• Įvynioti į foliją.</li><li>• Įvynioti į popieriu.</li><li>• Įdėti į maišelį.</li><li>• Įdėti į dėžutę.</li><li>• Įjungti konvejerį.</li><li>• Išjungti konvejerį.</li></ul>



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## Gudručiams

Ar mokiniai galėtų nustatyti laiką rankiniame laikrodyste arba mikrobangų krosnelėje? Susieti viryklės kaitvietes 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 rodyklų ar mygtukų skaičiaus (dažniausiai yra tik dvi rodyklės ir keli mygtukai), bet dėl šių prietaisų galimų atliliki funkcijų gausos. („Norint su tuo susidoroti reikia inžinieriaus mokslo laipsnio,” – pasakė žymus vartojo sasajos kūrimo psichologas Donas Normanas, pažvelgęs į savo naują rankinį laikrodį. Donas neturėjo aukštojo inžinerinio išsilavinimo, todėl nustatyti laiką savo naujame laikrodyste užtruko apie valandą. Kodėl jam prireikė valandos?)

Mokiniai galėtų pagalvoti ir įvardinti, kokius prietaisus ar įrenginius naudodami žmonės jaučiasi suglumę. Pavyzdžiu, 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 sasaja, 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 prie-monė, todėl jo vartojo sasaja turi būti suprantama ir lengvai naudojama.

Dėl kompiuterių vartotojo sasajos netikslumų įvyksta nemažai nelaimių (lėktuvų, traukinių, branduolinių elektrinių avarijų ir kt.), kai kada net prarandama žmonių gyvybių. Kai kurios dėl netinkamos vartotojo sasajos daromos klaidos piktina žmones (pavyzdžiui, per klaidą policijos išsiunciami 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 vartotojo sasajos klaidos. Žmonių galimybės apdoroti informaciją yra ribotos, j tai turėtų atsižvelgti vartotojo sasajos kūrėjai. Ne išeitis pateikti išsamią iš esmės blogos vartotojo sasajos instrukciją tikintis, kad žmogus ją studijuos ir atsimins. Be to, reikėtų atsižvelgti ir j žmogaus netobulumą (žmogus klysta), taip pat nereikėtų pamiršti ir tinkamo sasajos dizaino.

Vartotojo sasajos vertinimas yra svarbus jos dizaino kūrimo etapas. Šiame skyriuje yra keletas užduočių, kai mokiniai turi įvertinti pateiktas ir testuoti kitų mokinių sukurtas piktogramas. Kruopštesnis vertinimas gali būti atliktas kontroliuojamu psychologiniu eksperimentu.

### Daugiau informacijos

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

# 21 veikla

---

## Dialogas su kompiuteriu. Tiuringo testas

### Santrauka

Šios veiklos tikslas – diskutuoti, 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)



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

207

# Dialogas su kompiuteriu

---



## Diskusija

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

## Instrukcija

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 atsakinėjančiųjų 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 atsakinėtojui 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 taisyklos. 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 jvykdyti pasviruoju šriftu parašytas instrukcijas.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

Tadas ir Toma atsakymus užrašo ant lapelių.

1. Dar nepradėjus žaisti mokinį 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 jveikti 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. Mokiniams parodomи Tiuringo testo klausimai. Klausimų lapus galima padalyti mokiniams arba rodyti projektoriumi.

Mokiniams 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 atpažsta, kuris iš mokinų vaizduoja kompiuterį. Jei mokiniai greitai išsiaiškina „kompiuterį“, Tadas ir Toma mesdami monetą iš naujo parenka, kuriam mokinui – Žilvinui ar Kamilei – perduos klausimus. Auditorija ir vėl neturi žinoti, kuriam atsakinėtojui kuris tarpininkas perduoda klausimus.

Kamilės pateikiami atsakymai iš Tiuringo testo atsakymų lapo 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 tikrujų kompiuteris nesupranta, apie ką klausinėjama. Atsakymas „Aš nežinau“ yra neutralus ir kompiuteriu suteikia žmogiškumo, 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 Lindgren?
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  $2 \times 78$ ?
7. Kiek gautum ištraukės kvadratinę šaknį iš dviejų?
8. Sudék 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 mègsti valgyti?



Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

## **Darbo lapas *Tiuringo testo atsakymai***

---

Pasirinkite atsakymą į užduotą Tiuringo testo klausimą.

1. Koks yra Barto Simpsono mažosios sesers vardas?

**Nepamenu.**

2. Ką tu manai apie Astridą Lindgren?

**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  $2 \times 78$ ?

**166 (Sąmoningai neteisingas atsakymas!)**

7. Kiek gautum ištraukės kvadratinę šaknį iš dvejų?

**1,41421356237309504878**

8. Sudék 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.**

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.**



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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 – nuraštas iš testo.

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

Toliau 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 éjimas.

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

### II pokalbis

Klausimas: Pirmoje soneto eilutėje parašyta „Galéčiau tave lyginti su vasaros diena“. Gal „pavasario diena“ rimuoja 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 pokalbjų pagal labai paprastas taisykles. Skyriuje „Apie ką visa tai?“ aptariami kai kurie iš tų pokalbių. Mokiniai gali patys išbandyti „Elizą“.



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

## 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 absurdinė, 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. Pritariantys dirbtinio intelekto (DI) idėjai mokslininkai kritikuojami, 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 nuosakesnė nuomonė, kad DI nėra nei absurdinė, nei neišvengiamā idėja: kol dabartinės kompiuterių programos neparodo „intelekto“, DI klausimas yra eksperimentinis ir kol kas neatsakytas.

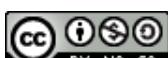
DI diskusijos priklauso nuo intelekto apibrėžimo. Siūloma ir diskutuojama daug apibrėžimų. Vieną iš apibrėžimų 1940 m. pasiūlė Alanas Tiuringas (Alan Turing), garsus Jungtinės Karalystės matematikas, karo laikotarpio kontržvalgybininkas, ilgų nuotolių bėgikas. Jis aprašė atvejj, kai kompiuteris galėtų parodyti savo intelektą. Jo scenarius buvo panašus į aprašytajį šiame skyriuje. Svarbiausia parinkti žmogaus ir kompiuterio (teletaipo – ankstyva 1940 m. technologija) tarpininką. Jei tarpininkas negalėdavo patikimai atskirti kompiuterio ir žmogaus, tai kompiuteris išlaikydavo Tiuringo intelekto testą. Šis kompiuterio naudojimo eksperimentas atmetė galimybę suteikti kompiuterui 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 parengiamajį 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, j kai kuriuos klausimus jis sunkiai atsakytu, pavyzdžiui, tikrai per ilgai skaičiuotų  $123456 \times 789012$  sandaugą ar ilgai užtruktų atlikti kitus aritmetinius veiksmus su dideliais skaičiais.

Tačiau kompiuterui gana paprasta vaizduoti natūralų pokalbj. 1960 m. sukurta gerai žinoma programa „Eliza“ imituoja psichoterapeuto pokalbj su pacientu. Pateikiame to pokalbio išstrauką.

- 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.



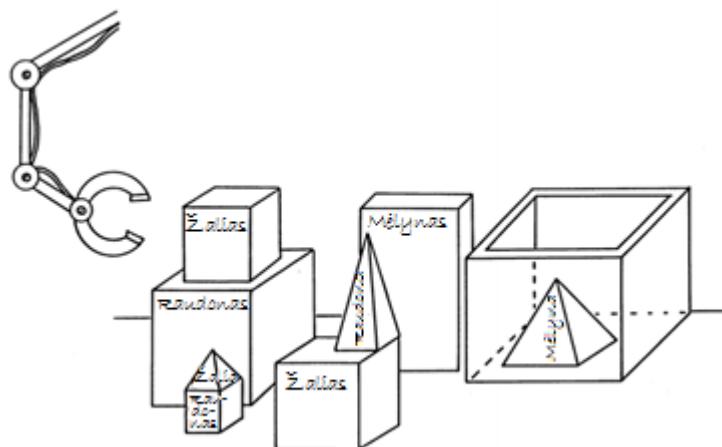
Kompiuteris: Gaila, kad jūs depresuojate.  
 Žmogus: Tai tiesa. Aš nelaiminga.  
 Kompiuteris: Ar manote, kad lankymasis čia jums padės būti laimingesnei?  
 Žmogus: Man reikia pagalbos.  
 Kompiuteris: Kas jums padėtų?  
 Žmogus: Gal galėčiau išmokti sutarti su savo mama.  
 Kompiuteris: Papasakokite 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 jrašytus atsakymus pagal pašnekovo pasakyta 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 (pvz., „Ar ką nors darėte, kad....“).

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

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

SHRDLU galėjo tęsti sklandų pokalbj. Toliau 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 dideli 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, kuria piramidę paimti.

*Kadangi yra trys pyramidės, programa nieko nedaro.*

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

*Programa turi išsiaiškinti, ką reiškia „jq“.*

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

*Informuoja, ar gerai suprato.*

Robotas: Gerai.

*Robotas ranka padeda raudoną kubą, paima aukštesnę mėlyną figūrą ir jdeda jq 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 minėtą piramide.*

Robotas: Dėžėje.

Žmogus: Padaryk bokštą...

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

*Programa iškart reaguoja, kai išgirsta nežinomą žodį.*

Žmogus: „Bokštą“ – 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 suprasta naujo žodžio „bokštas“ apibrėžtis.*

Žmogus: Pastatyk.

Robotas: Gerai.

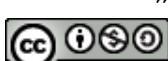
*Apibrėžtyje nurodytų instrukcijų pakanka, kad robotas pastatyti bokštą.*

Tačiau intelekto iliuzija gali būti lengvai sugriaunama.

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

Žmogus: Išmainyk dideli raudoną kubą į vandens pistoletą.

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



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

215

Žmogus: „Išmainyk“ – nemokamai apsikeisk daiktais.  
Robotas: Atsiprašau, nežinau, ką reiškia „nemokamai“.  
Žmogus: „Nemokamai“ – poelgis, padarytas noriai, be prievarbos.  
Robotas: Atsiprašau, nežinau, ką reiškia „poelgis“.  
Žmogus: „Poelgis“ – žmogaus tikslinai, 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 pripažsta, kad nesupranta žodžio, kurį pati vartoja.

Galime tik įsivaizduoti, kaip robotas galėtų testi pokalbj (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 piramide už tavo gleivętą varlę.

1. Robotas jvertina, 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 poliarinių lokų, 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 mésos. Jis rado mésos. Jis suédė Trumaną. Trumanas mirė.

Galima įsivaizduoti, kaip ją pagyvintų 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,



Licensed under Creative Commons

© 2015 Computer Science Unplugged (csunplugged.org)

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 įveiktu Tiuringo testą. Net jei jis būtų sukurtas, daugelis filosofų tai neigtų, sakydami, kad Tiuringo testas iš tikrujų 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“ (The MIT Press, Cambridge, MA, 1985) filosofas Johnas Haugelandas aprašo diskusijas apie dirbtinį intelektą, tai daugiausiai SHRDLU pokalbiai ir jų aptarimas. Šiame skyriuje pateikiti pavyzdžiai iš šios knygos.

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).

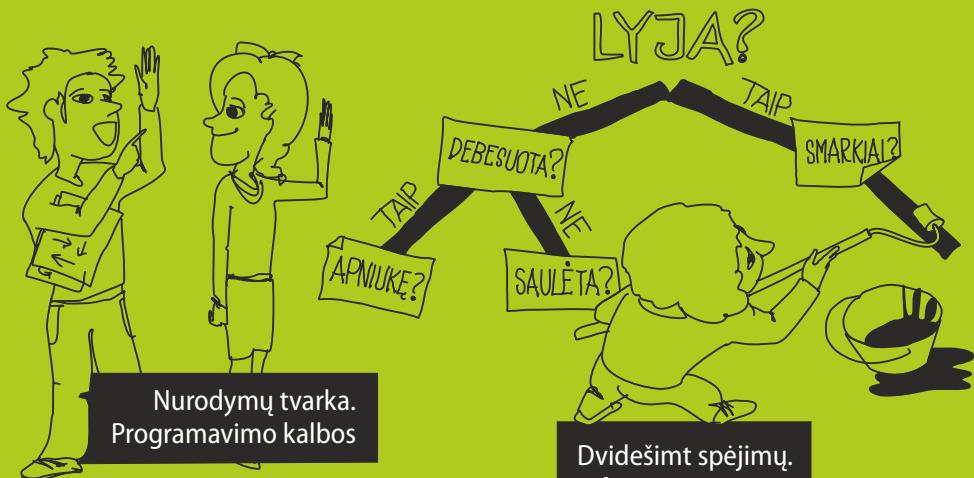


Licensed under Creative Commons

© 2015 Computer Science Unplugged ([csunplugged.org](http://csunplugged.org))

217

Akmeninės plokštelės.  
Tinklo komunikacijų protokolas



Nurodymų tvarka.  
Programavimo kalbos

Dvidešimt spėjimų.  
Informacijos teorija



Turistų miestas.  
Dominuojančios sekos



Dalijimasis paslaptimi.  
Informacijos slėpimo protokolai

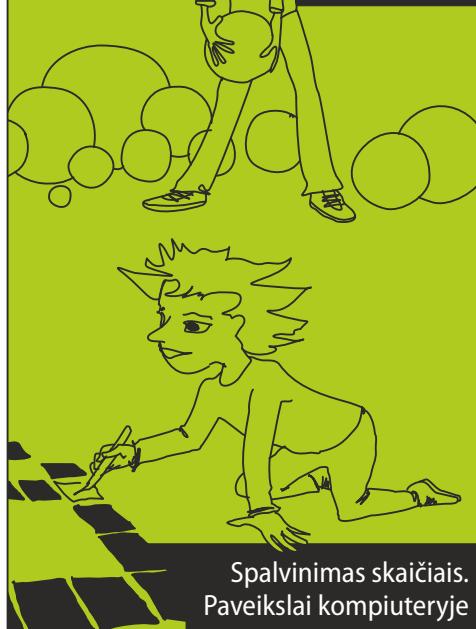


Vargšas kartografas.  
Grafo spalvinimas

Lengviausias ir sunkiausias.  
Rikiavimo algoritmai



Lobio ieškojimas.  
Baigtiniai automatai



Spalvinimas skaičiais.  
Paveikslai kompiuteryje



Dar greičiau!  
Tinklinis rikiavimas