

S  
i  
d  
r  
i  
s  
i  
s  
i  
s



# Informatikos ir informatinio mąstymo uždavinių rinkinys

## Nr. 3



Šiame rinkinyje pateikiami XIII informatikos ir informatinio mąstymo konkurso „Bebras“ II etapo uždaviniai, jų atsakymai ir paaiškinimai, koks informatikos turinys ir konceptai atskleidžiami, kaip ir kuo uždavinys ypatingas ar įdomus informatikai. Visi uždaviniai (įskaitant grafiką ir kitą medžiagą) licencijuojami pagal Kūrybinių bendrijų licenciją – „Creative Commons Attribution-ShareAlike 4.0 International License“. Šis uždavinių rinkinys skirtas ugdyti 9–12 klasių mokinių informatinio mąstymo gebėjimus.

*Dėkojame Daumilui Ardickui, Gytautui Beresnevičiui, dr. Gintautui Grigui, dr. Tatjanai Jevsikovai, dr. Anitai Juškevičienei, Alvidai Lozdienei, dr. Gabrielei Stupurienei, Elenai Sutkutei, talkinusiems verčiant ir adaptuojant uždavinius. Taip pat dėkojame tarptautinei „Bebro“ konkurso bendruomenei ir uždavinių autoriams.*

Parengė Lina Vinikienė  
Konsultavo Valentina Dagienė  
Redagavo Viktoras Dagys  
Viršelį kūrė Vaidotas Kinčius



Užduočių rinkinys platinamas pagal kūrybinių bendrijų licenciją nekomerciniais tikslais  
(Creative Commons Attribution–NonCommercial–ShareAlike)

## Įvadas

„Bebro“ konkurso tikslas – sudominti bendrojo ugdymo mokyklų mokinius informatikos mokslo fundamentinėmis idėjomis, skatinti sumaniau naudotis informacinėmis technologijomis, ugdyti mokinių kūrybiškumą, informacinę kultūrą, algoritminį, loginį, kritinį ir informatinį mąstymą.

2015 m. tarptautinė „Bebro“ bendruomenė keitė konkurso pavadinimą – pavadinta informatikos ir informatinio mąstymo iššūkiu. Nors lietuviškai vadinamas konkursu, iš tiesų tai įvairių informatikos veiklų tinklas informatiniam mąstymui ugdyti.

Daugelį metų konkurse dalyvauja beveik visos Europos Sąjungos šalys, taip pat Australija, Azerbaidžanas, Baltarusija, Brazilija, Iranas, Islandija, Izraelis, Japonija, Jungtinės Amerikos Valstijos, Kanada, Kazachstanas, Malaizija, Naujoji Zelandija, Pakistanas, Pietų Afrikos Respublika, Pietų Korėja, Rusija, Singapūras, Šveicarija, Taivanas, Turkija.

Dauguma šalių dalyvių leidžia knygeles, kuriose aiškinami uždavinių sprendimai ir – svarbiausia – kaip kiekvienas uždavinys susijęs su informatika, kokie konceptai jame slypi, parodoma, kaip galima toliau plėtoti informatinio mąstymo pažinimą.

„Bebro“ uždavinius rengia konkurse dalyvaujančių šalių atstovai – entuziastai mokslininkai ir mokytojai. Kasmet pavasarį rengiami seminarai, kuriuose kelias dienas intensyviai aptariami ir atrenkami kiekvienos šalies pasiūlyti uždaviniai. Keturis kartus šie seminarai – dirbtuvės – vyko Lietuvoje, Pasvalio r., Balsių malūne (2005, 2006, 2007 ir 2009 m.), 2008 ir 2013 m. seminaras surengtas Lenkijoje, Torunėje, 2010 m. – Vokietijoje, Dagštule, tris kartus (2011, 2012 ir 2014 m.) uždavinių dirbtuvės vyko Druskininkuose. 2015 m. dirbtuvės vyko Sankt Peltene, Austrijoje. 2016 m. gegužės 23–27 d. „Bebro“ dirbtuvės vyko Bodrume, Turkijoje.



„Bebro“ konkurso tarptautinių dirbtuvių dalyviai Bodrume, Turkijoje (2016 m.)



Prof. Valentina Dagienė, Bodrumas, Turkija (2016 m.)

Kiekviena šalis turi savo „Bebro“ konkurso organizavimo komitetą ir konkursą vykdo atskirai. Iki 2017 m. buvo skiriamos penkios konkurse dalyvaujančių mokinių amžiaus grupės, jungiant po dvi gretimas klases: 3–4, 5–6, 7–8, 9–10, 11–12 klases. Kiekviena šalis savo amžiaus grupės gali šiek tiek koreguoti. Lietuvoje 1–2 klasių mokiniai konkurse pradėjo dalyvauti nuo 2017 m.

2016 metų „Bebro“ uždavinius pasaulyje sprendė 1 609 619 mokinių (duomenis pateikė 34 valstybės). Lietuvoje tais metais dalyvavo per 33 tūkstančius mokinių. 2016 m. konkursas buvo vykdomas daugiau nei 50 pasaulio šalių (25 ES narėse).

Lietuvoje „Bebro“ konkursas dešimt metų buvo vykdomas vienu etapu: uždaviniai buvo sprendžiami mokyklose, prisijungus internetu prie „Bebro“ varžybų lauko ([lt.bebbras.lt](http://lt.bebbras.lt)). Nuo 2014 metų (XI konkurso) buvo pradėtas rengti ir antrasis etapas, į kurį kviečiami geriausi pirmojo etapo dalyviai, jo metu uždaviniai sprendžiami, atvykus į paskirtas aukštąsias mokyklas.



2016–2017 mokslo metais antrasis etapas vyko 2017 m. sausio 28 d. Į konkursą galėjo registruotis tik 9–12 klasių mokiniai. Iš viso užsiregistravo 548 mokiniai. Konkursas vyko septyniose Lietuvos aukštosiose mokyklose: Kauno technologijos, Klaipėdos, Šiaulių, Vilniaus universitetuose, Alytaus, Panevėžio, Utenos kolegijose. Į renginį atvyko ir uždavinius sprendė 385 mokiniai: 9–10 klasių – 198 ir 11–12 klasių – 187 mokiniai. Dalyviai turėjo išspręsti 15 uždavinių, tam buvo skirta 30 min. Pradėdami spręsti mokiniai turėjo po 45 taškus.

*Įdomu.* Lietuvos „Bebro“ logotipų kaita:



Iki 2015 metų



2015–2017 metais



Nuo 2017 metų

2016 m. profesorė Valentina Dagienė pasiūlė „Bebro žaidimo kortelių“ idėją. Sprendžiant kortelėse pateiktas užduotis mokiniai supažindinami su pagrindinėmis informatikos ir informatinio mąstymo sąvokomis, ugdomas jų kūrybiškumas, informacinė kultūra, algoritminė ir kompiuterinė mąstysena. Šias žaidimo korteles naudoja Lietuvos, Austrijos, Šveicarijos ir Švedijos mokiniai.



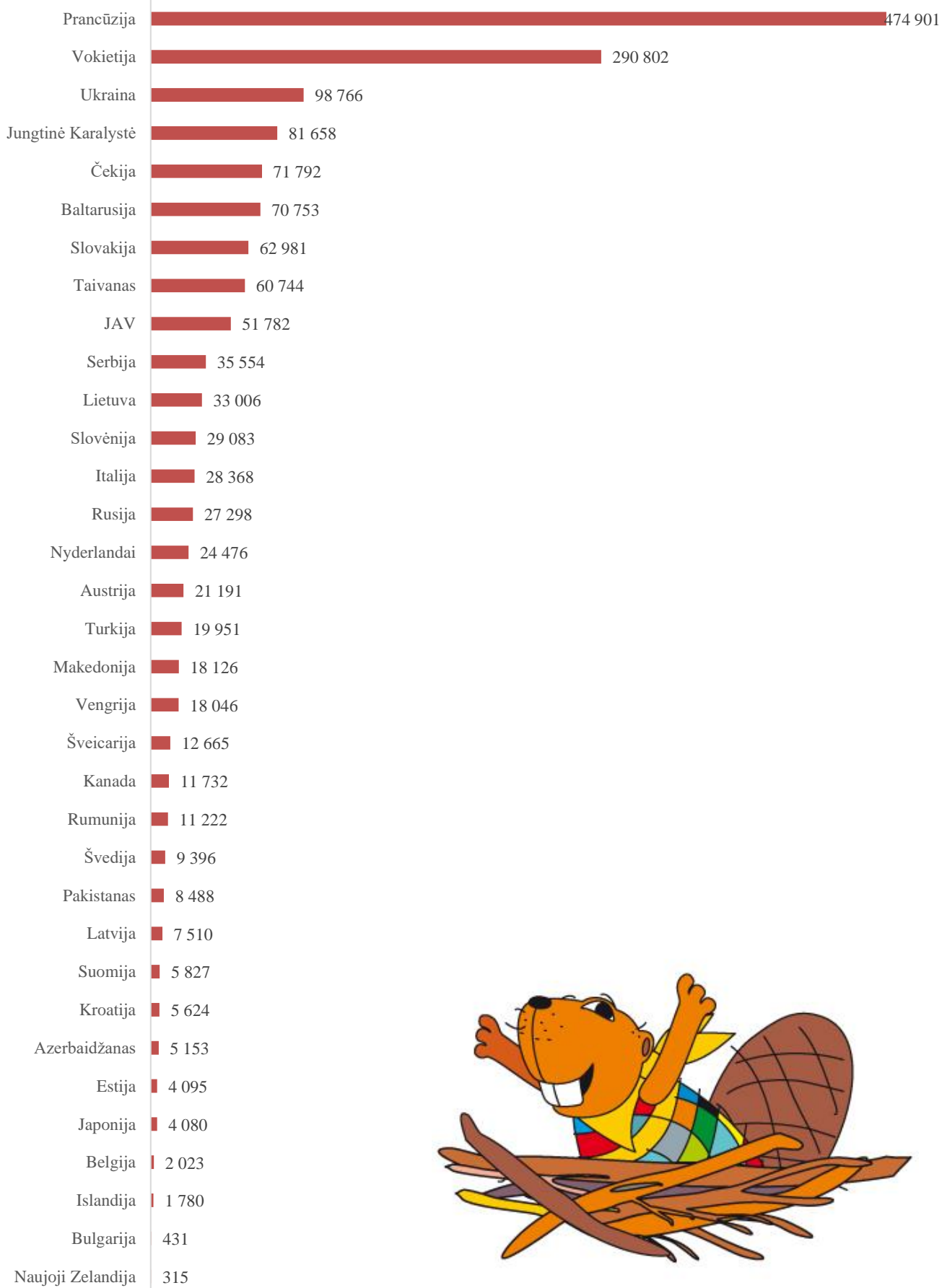
„Bebro“ uždavinių kortelės (2016 m.)



2016 m. konkurso sumanytoja prof. Valentina Dagienė pelnė Europos skaitmeninio pasaulio moters apdovanojimą už nuopelnus švietimui ir informacinių technologijų plėtrą. Taip pat 2016 m. Lietuvą garsino kartu su komanda pasiektas Gineso rekordas – modernių technologijų ir verslumo renginyje #SWITCH! surengta didžiausia programavimo pamoka vaikams.



*#SWITCH! didžiausia programavimo pamoka*



2016 m. „Bebro“ konkurso dalyvių skaičius pagal šalis (34 šalių duomenys)



Lentelėje pateikiamas XIV konkurso II etapo uždavinių skirstymas pagal amžiaus grupes.

| Nr. | Pavadinimas                   | Uždavinio identifikatorius | Jauniai (9–10 kl.) | Kolegos (11–12 kl.) |
|-----|-------------------------------|----------------------------|--------------------|---------------------|
| 1   | Skaičių sukimosi galvosūkis   | 2012-HU-01a                | 6                  |                     |
| 2   | Piramidė                      | 2012-SK-08                 | 6                  |                     |
| 3   | Skaičiavimas senu kompiuteriu | 2016-CA-06                 | 6                  |                     |
| 4   | Plaustų numeravimas           | 2016-LT-02                 | 6                  |                     |
| 5   | Autobusų stotelė              | 2016-UA-02-1               | 6                  |                     |
| 6   | Raidžių sukeitimas            | 2016-NL-05                 | 9                  | 9                   |
| 7   | Abėcėliniai vardai            | 2016-CZ-05                 | 9                  |                     |
| 8   | Maisto ruošimas               | 2016-HU-07                 | 9                  |                     |
| 9   | Apsirengimo tvarka            | 2016-SE-01                 | 9                  |                     |
| 10  | Tvorelė                       | 2016-SP-02                 | 9                  |                     |
| 11  | Spartusis kėlimas laipsniu    | 2016-CH-01                 | 12                 | 9                   |
| 12  | Augalo gyvenimas              | 2011-DE-01                 | 12                 |                     |
| 13  | Algoritmavimo menas           | 2016-AU-06                 | 12                 |                     |
| 14  | Šokinėjanti beždžionė         | 2016-CA-02                 | 12                 |                     |
| 15  | Plytelių dėjimas              | 2016-PL-04a                | 12                 | 9                   |
| 16  | Raidžių žaidimas              | 2011-CH-05                 |                    | 6                   |
| 17  | Rask vagį!                    | 2016-BE-02                 |                    | 6                   |
| 18  | Šuolių čempionatas            | 2016-IR-01                 |                    | 6                   |
| 19  | Penki kauliukai               | 2016-IS-01                 |                    | 6                   |
| 20  | Trasų tyrinėjimas             | 2016-LT-08                 |                    | 6                   |
| 21  | Salos ir tiltai               | 2012-LT-01                 |                    | 9                   |
| 22  | Funkcijų skaičiavimas         | 2016-CA-08a                |                    | 9                   |
| 23  | Viešbučio raktas              | 2011_NL_01                 |                    | 12                  |
| 24  | Kelionė diskais               | 2012-BG-03                 |                    | 12                  |
| 25  | Krypties keitimas             | 2012-CA-01                 |                    | 12                  |
| 26  | Savanaudės voverės            | 2016-RU-08a                |                    | 12                  |
| 27  | Žaidimas L figūromis          | 2016-TW-07b                |                    | 12                  |

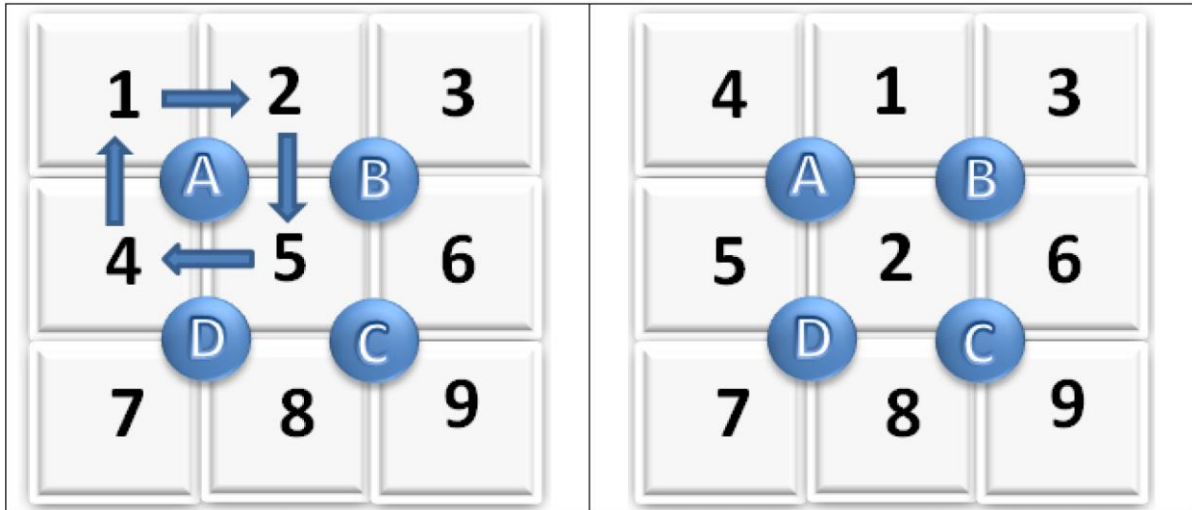
Kiekvieno uždavinio pradžioje nurodoma, kuriai amžiaus grupei jis skiriamas ir jo sudėtingumo lygis:

- lengvas – 6,
- vidutinis – 9,
- sunkus – 12.

Taip pat pateikiamas uždavinio atsakymas ir paaiškinimas, kaip uždavinys susijęs su informatika.

## 1. Skaičių sukimosi galvosūkis

Bebras Henris žaidžia žaidimą, kurio esmė tokia: kai paspaudžiamas vienas iš mygtukų (A, B, C ar D), aplink tą mygtuką esantys skaičiai pasukami pagal laikrodžio rodyklę (žr. paveikslą kairėje). A mygtuko paspaudimo rezultatas parodytas paveiksle dešinėje.



Henris paspaudė keturis mygtukus iš eilės: D, C, B, B.

Kur atsidurs skaičius 4 paspaudus minėtus mygtukus?

A.

|   |  |  |
|---|--|--|
|   |  |  |
| 4 |  |  |
|   |  |  |

B.

|  |   |  |
|--|---|--|
|  | 4 |  |
|  |   |  |
|  |   |  |

C.

|  |  |   |
|--|--|---|
|  |  |   |
|  |  | 4 |
|  |  |   |

D.

|  |   |  |
|--|---|--|
|  |   |  |
|  |   |  |
|  | 4 |  |

## Paaiškinimas

Teisingas atsakymas: B.

Tarpiniai rezultatai:

Paspaudus D:

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 7 | 4 | 6 |
| 8 | 5 | 9 |

Paspaudus C:

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 7 | 5 | 4 |
| 8 | 9 | 6 |

Paspaudus B:

|   |   |   |
|---|---|---|
| 1 | 5 | 2 |
| 7 | 4 | 3 |
| 8 | 9 | 6 |

Paspaudus B:

|   |   |   |
|---|---|---|
| 1 | 4 | 5 |
| 7 | 3 | 2 |
| 8 | 9 | 6 |

## Tai informatika!

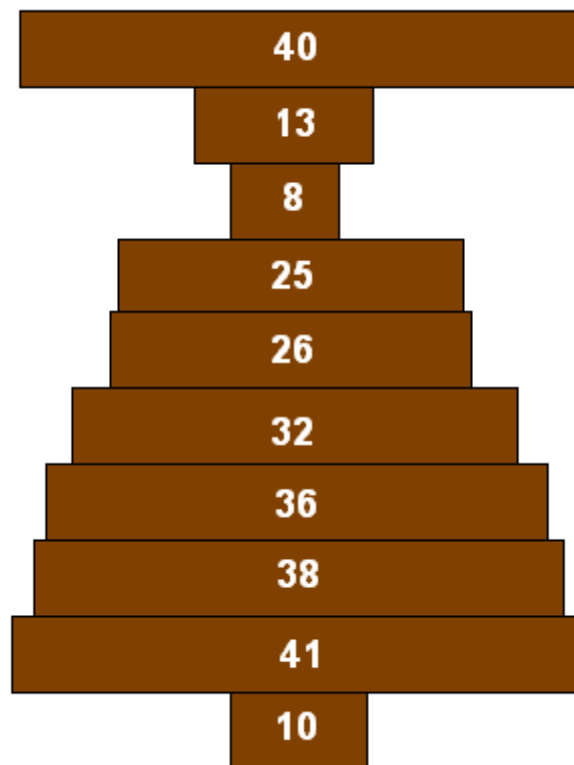
Mokėjimas perskaityti, suprasti ir pritaikyti taisykles – algoritmavimo pagrindas. Taisyklių užrašymas formaliais žymenimis – tai žingsnis į programavimą. Šiame uždavinyje svarbu suprasti taisyklę ir ją atidžiai pritaikyti.

Raktiniai žodžiai: algoritmo žingsniai, veiksmų atlikimas, kartojimas.

## 2. Piramidė

Faraonas Bebras Didysis statosi žemyn platėjančią piramidę iš skirtingo dydžio plokščių. Ant kiekvienos plokštės užrašytas kraštinės ilgis. Deja, bebrai statybininkai suklydo ir įdėjo keletą neteisingo dydžio plokščių. Darbų vadovas, taisydamas statybininkų klaidas, netinkamas plokštes keičia tinkamomis.

Keitimai aprašomi taip:  $40 \rightarrow 5$  reiškia, kad plokštė, kurios kraštinės ilgis 40, keičiama plokšte, kurios kraštinės ilgis 5.



Kuria keitimų seka gausime taisyklingą piramidę?

- A)  $40 \rightarrow 15, 13 \rightarrow 16, 10 \rightarrow 42$
- B)  $40 \rightarrow 5, 13 \rightarrow 7, 10 \rightarrow 40$
- C)  $40 \rightarrow 6, 13 \rightarrow 7, 8 \rightarrow 24$
- D)  $40 \rightarrow 12, 8 \rightarrow 15, 10 \rightarrow 42$



## **Paaiškinimas**

Teisingas atsakymas: D.

Taisyti piramidę galima daugeliu būdų, todėl racionaliausia teisingo atsakymo ieškoti tikrinant visas pateiktas keitimų sekas.

## **Tai informatika!**

Tai duomenų pertvarkymo uždavinys. Kartais pateikti duomenys neatitinka sąlygų, todėl juos reikia pertvarkyti. Jei duomenų daug, siekiama minimizuoti galimų pertvarkymų skaičių. Šiuo atveju uždutis paprasta – tiesiog pritaikyti pateiktas keitimų sekas ir nustatyti, kuri iš jų pateikia teisingą rezultatą.

Raktiniai žodžiai: duomenų pertvarkymas, algoritmo taikymas, komandų seka.

### 3. Skaičiavimas senu kompiuteriu

Bebrai mėgsta laikyti senus kompiuterius. Bebrė Roberta turi nusipirkusi kompiuterį, kuris priima tik 1 skaičių po kabelio: viskas, kas eina toliau, tiesiog nebepateikiama.

Pavyzdžiui, jei Robertos kompiuteriui užduotume suskaičiuoti  $7 : 5$ , tai gautume atsakymą 1,4 (jis yra teisingas), tačiau, jei norėtume apskaičiuoti  $7 : 4$ , kompiuteris pateiktų atsakymą 1,7 (nes  $7 : 4 = 1,75$ , kitaip tariant, 5 nepateikiamas). Taigi Robertos kompiuterio paklaida lygi 0,05.

Šitaip atliekamas kiekvienas veiksmas. Pavyzdžiui, Roberta skaičiuoja  $(7 : 4) : 2$ . Atlikus pirmą veiksmą, gaunama 1,7, toliau skaičiuojama  $1,7 : 2$  ir gaunama 0,8. Šiuo atveju paklaida lygi 0,075.

**Jei Robertai reikia apskaičiuoti  $((10 / 3) \times (10 / 3)) \times 9$ , kokia bus paklaida (skirtumas tarp tikro atsakymo ir pateikiamo Robertos kompiuterio)?**

- (A) 0,0
- (B) 1,3
- (C) 2,8
- (D) 3,3

## **Paaškinimas**

Teisingas atsakymas: (C) 2,8.

Robertos kompiuteriu apskaičiuojame:

$$(10 / 3) = 3,3.$$

Tada apskaičiuojame  $(10 / 3) \times (10 / 3) = 3,3 \times 3,3$  ir gauname 10,8.

$$\text{Tada } 10,8 \times 9 = 97,2.$$

Tikrasis atsakymas turėtų būti  $(100 / 9) \times 9 = 100$ .

$$100 - 97,2 = 2,8 \text{ – štai paklaida.}$$

## **Tai informatika!**

Kompiuteryje skaičiai yra baigtiniai, ribota kompiuterio atmintis gali kaupti tik baigtinį informacijos kiekį. Kai projektuojamas procesorius, numatomas maksimalus bitų skaičius, kuris naudojamas skaitmenims kaupti. Kai naudojame trupmenas, ne kiekvienas skaičius gali būti tiksliai įrašomas, todėl atsiranda apvalinimo paklaidų. Šios paklaidos kaupiasi, dėl to dideli skaičiavimai gali būti netikslūs.

Kompiuterininkai ieško algoritmų, kaip sumažinti skaičiavimų paklaidas ir gauti kuo tikslesnius atsakymus.

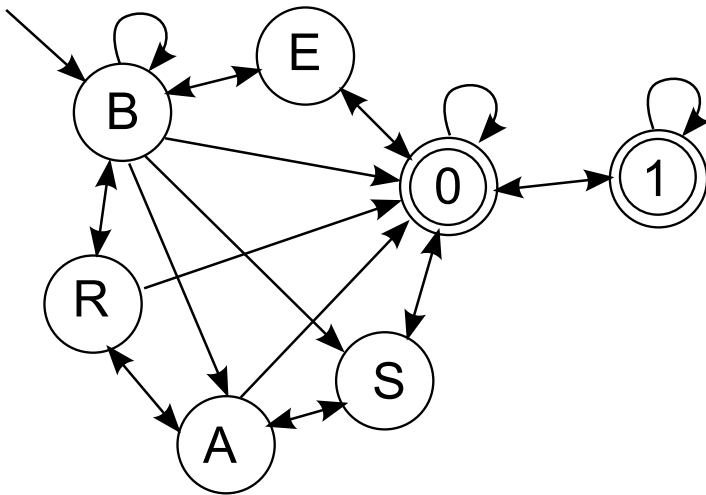
Raktiniai žodžiai: skaičiavimų tikslumas, skaičių apvalinimas, paklaida.

## 4. Plaustų numeravimas

Bebrai renčia plaustus. Plaustai registruojami – kiekvienam plaustui suteikiamas atskiras numeris ir šio numerio lentelė pritvirtinama prie to plausto.



Numeris sudaromas pagal taisyklės, pavaizduotas schema. Pradedama raide B ir baigiama skaitmeniu 0 arba 1.



Kurie du iš pateiktų numerių negali būti registruojami?

BB0001

BBB100

BBB011

BB0100

BR00A0

BSA001

BEOS01



## Paaiškinimas

Kiekvieną nurodytą numerį reikia kruopščiai patikrinti pagal pateiktą schemą. Taip nustatysime, kurie du numeriai sudaryti ne pagal taisykles.

Numeris BBB100 neteisingas, nes skaičių dalis pradedama 1.

Numeris BR00A0 neteisingas, nes po 0 negali būti A (schemoje rodyklė rodo tik į vieną pusę).

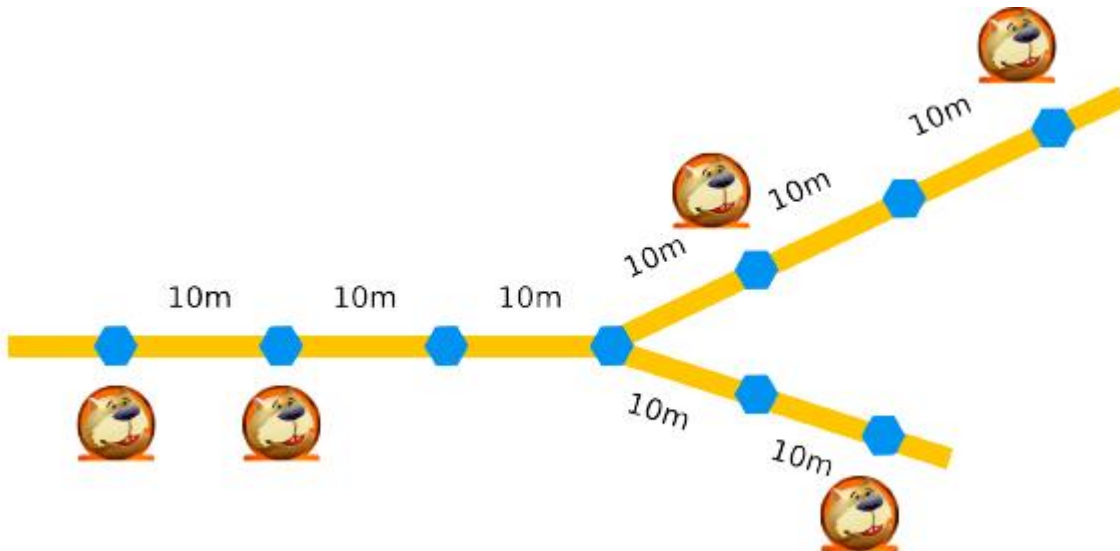
## Tai informatika!

Baigtinis automatas yra svarbi teorinio kompiuterių mokslo dalis. Kai tik reikia skaityti duomenis, ypač raidžių ir skaitmenų sekas, pasitelkiamas baigtinio automato modelis. Iš tikrųjų baigtinis automatas – tai matematinė abstrakcija, modeliuojanti sistemos būsenų kaitą, priklausomai nuo ankstesnės būsenos ir įėjimo signalų, kai būsenų ir galimų įėjimo signalų skaičius yra baigtinis (taip aiškintinas ir automato pavadinimas). Baigtiniu automatu patogiau aprašyti simbolių sekų sudarymą: naudojantis baigtiniu automatu, nesunku nustatyti, ar simbolių seka sudaryta teisingai, ar ne. Baigtiniai automatai paprastai vaizduojami grafais ar į juos panašiomis diagramomis.

Raktiniai žodžiai: baigtinis automatas, būsena, modeliavimas.

## 5. Autobusų stotelė

Penki bebrų nameliai sustatyti šitaip:



Bebrai nori įrengti autobusų stotelę vienoje iš devynių šešiakampiais pažymėtų vietų.

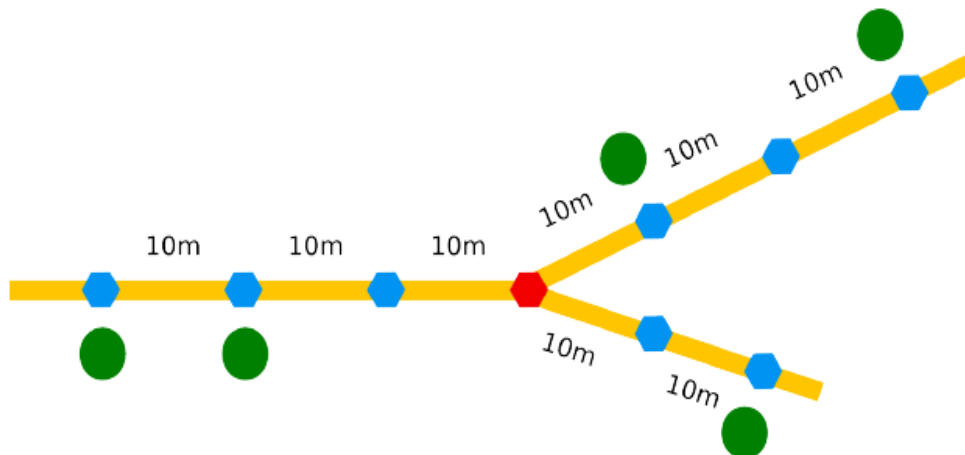
Paveiksle nurodyti atstumai tarp šešiakampių.

Bebrai nori, kad atstumų suma nuo jų namų iki autobusų stotelės būtų kiek įmanoma mažesnė.

Nustatykite autobusų stotelės vietą.

## Paaiškinimas

Teisingas atsakymas:



Vienas iš sprendimų – išbandyti visas 9 galimas autobusų stotelės įrengimo vietas ir apskaičiuoti atstumus. Tam reikėtų atlikti daug skaičiavimų. Pagalvokime ir raskime gudresnį sprendimą.

Tarkime, autobusų stotelė bus kelių sankirtoje. Tada visų atstumų nuo namelių iki stotelės suma bus:

$$30 + 20 + 10 + 30 + 20 = 110$$

Jeigu stotelę perkelsime į kairę pusę  $x$  metrų, pirmieji du atstumai sutrumpės  $x$  metrų, o likusieji trys atstumai nuo namelių iki stotelės pailgės  $x$  metrų:

$$(30 - x) + (20 - x) + (x + 10) + (30 + x) + (x + 20) = 110 + x$$

Toks pats rezultatas gaunamas perkėlus stotelę per  $x$  metrų į dešinę aukštyn einančiu keliu:

$$(30 + x) + (20 + x) + (x + 10) + (30 - x) + (-x + 20) = 110 + x$$

Jeigu stotelę perkelsime į dešinę pusę žemyn einančiu keliu per  $x$  metrų, atstumų suma bus:

$$(30 + x) + (x + 20) + (10 + x) + (x + 30) + (20 - x) = 110 + 3x$$

### Tai informatika!

Tai klasikinis grafų teorijos medžio centro paieškos uždavinys.

Sprendžiame optimizavimo uždavinį. Jis panašus į realaus pasaulio transporto tinklų uždavinį, kai miesto planuotojams reikia parengti optimalų metro, autobusų, traukinių ar kito transporto sistemų planą.

Informatikoje sukurta daugybė algoritmų optimaliam sprendimui ieškoti. Šiam uždaviniui (kai viršūnių nėra daug) taikomas visiško perrinkimo algoritmas. Jei tinklai didesni (viršūnių daug), toks algoritmas jau netinka. Jei vietovės išrikiuojamos į vieną eilę, galima taikyti medianos skaičiavimo algoritmą. Optimalios viršūnės ieškoma dinaminio programavimo metodu.

Raktiniai žodžiai: optimizavimas, medžiai (grafų teorija).

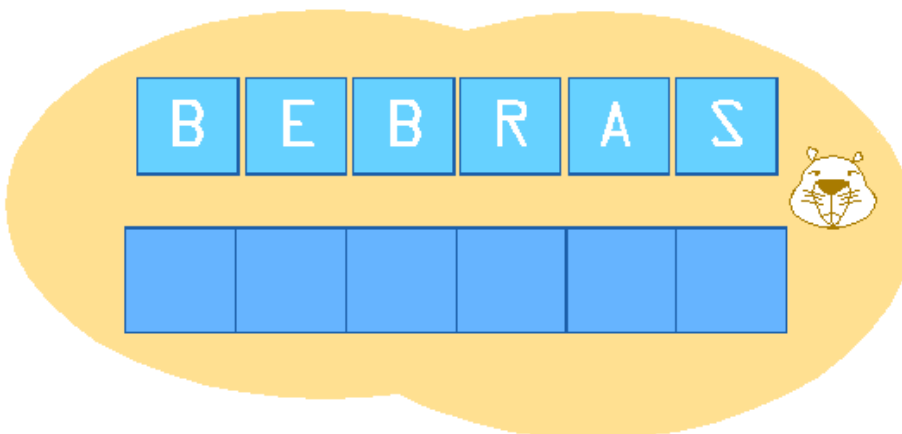
## 6. Raidžių sukeitimas

Funkcija **VisiškasSukeitimas(n, s)** turi du pradinius duomenis: skaičių ir žodį. Rezultatas – žodis, kurio raidės sukeistos vietomis.

Lentelėje parodyta, kaip veikia funkcija **VisiškasSukeitimas(3, programa)**. Šiuo atveju funkcija atliko 6 sukeitimus ir gautas rezultatas **ogramapr**:

|          |                      |          |
|----------|----------------------|----------|
| programa | Sukeitė 1 raidę su 3 | orpgrama |
| orpgrama | Sukeitė 2 raidę su 4 | ogprrama |
| ogprrama | Sukeitė 3 raidę su 5 | ogrrpama |
| ogrrpama | Sukeitė 4 raidę su 6 | ograprma |
| ograprma | Sukeitė 5 raidę su 7 | ogramrpa |
| ogramrpa | Sukeitė 6 raidę su 8 | ogramapr |

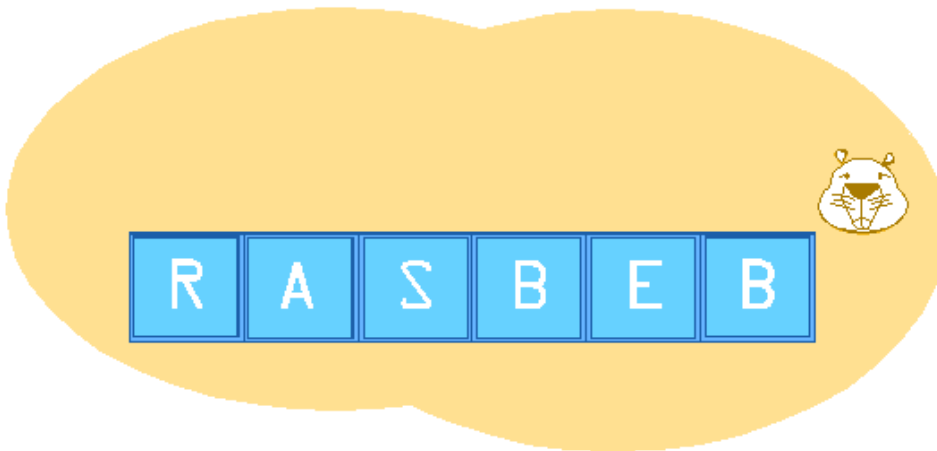
Koks funkcijos **VisiškasSukeitimas (4, BEBRAS)** rezultatas?





## Paaiškinimas

Teisingas atsakymas:



Atsakymas gaunamas atliekant veiksmus, analogiškus pateiktiems uždavinio sąlygoje.

BEBRAS

REBBAS

RABBES

RASBEB

### Tai informatika!

Programuojant dviejų kintamųjų sukeitimas – tai šių kintamųjų reikšmių tarpusavio sukeitimas.

Programuojant aukšto lygio kalbomis funkcijos yra labai svarbios. Tai labai naudingas būdas, kai kelis kartus reikia atlikti tuos pačius veiksmus. Funkcijos supaprastina programų tekstų suvokimą. Funkcijos yra integrali programavimo pagrindų dalis.

Raktiniai žodžiai: procedūra, funkcija, programa, taikomoji programa, eilutė.

## 7. Abėcėliniai vardai

Vardas vadinamas abėcėliniu, jei gali būti sudaromas pagal šį algoritmą:

- Surikiuoti vardo raides pagal abėcėlę.
- Imti pirmąją sąrašo raidę.
- Imti tolesnę sąrašo raidę ir pridėti ją vardo kairėje arba dešinėje.
- Šiuos veiksmus daryti tol, kol sąrašas taps tuščias.

LABYS yra abėcėlinis vardas: surikiavę jo raides gauname ABYL, toliau raidės dėliojamos pagal algoritmą:

**A -> AB -> ABY -> LABY -> LABYS**

JONĖ nėra abėcėlinis vardas: surikiavę gauname JĖNO. Raidės Ė negalima niekaip pridėti prie pirmosios raidės J, nes abi raidės J ir Ė nėra gretimos žodyje JONĖ.

**Kuris iš šių vardų yra abėcėlinis?**

JONAS

MIGLĖ

DANILĖ

TOMAS

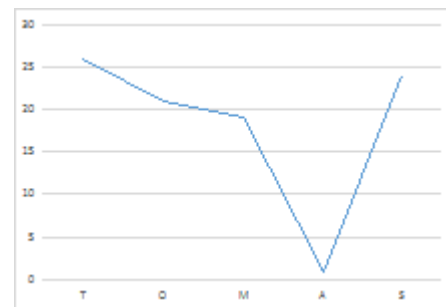
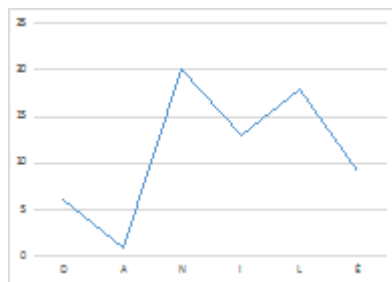
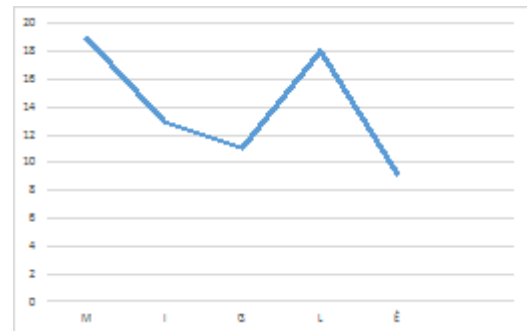
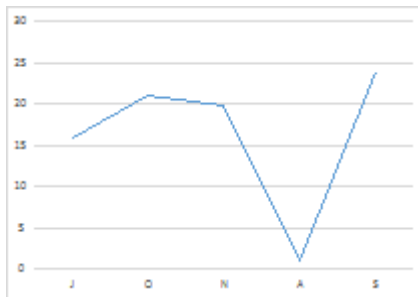
## Paaiškinimas

Teisingas atsakymas: TOMAS.

Galime sudaryti kiekvieno vardo diagramas. Vardo raidės iš eilės išdėstomos horizontaliojoje ašyje. Vardo raidžių numeriai abėcėlėje išdėstomi vertikaliuoju ašyje. Jungiant taškus nubrėžiama kreivė.

Jeigu diagramoje yra tik vienas staigus kreivės lūžis, panašus į V, vardas yra abėcėlinis. Jei diagrama panaši į apverstą V (^), tai žodis neabėcėlinis.

Panagrinėkime diagramų pavyzdžius:



### Tai informatika!

Spręsdami šį uždavinį neišsiversime be teksto kodavimo ir vizualizavimo.

Koduojama pagal taisyklės – algoritmą. Taikant algoritmą arba vaizdesnį būdą – diagramas – galima patikrinti, ar esamas vardas yra abėcėlinis. Metodų vizualizavimas labai svarbus informatikoje, todėl ieškoma vis geresnių vizualizavimo būdų.

Reikšminiai žodžiai: abėcėlinis rikiavimas, algoritmas.

## 8. Maisto ruošimas

Matas turi naują viryklę ir nori suprogramuoti patiekalo ruošimą žingsnis po žingsnio pagal recepte pateiktus nurodymus. Kiekvienas nurodymas pradedamas skaičiumi. Jei reikalingas kuris nors ingredientas, skliaustuose nurodoma šio ingrediento raidė. Kiekvienas nurodymas rašomas atskiroje eilutėje.

Pavyzdžiui, nurodymai „Sumaišyti miltus ir aliejų“ ir „Virti tam tikrą laiką“ galėtų būti suprogramuoti taip:

4 (M, A)

2

| Nurodymai                | Ingredientai  |                  |
|--------------------------|---------------|------------------|
| 1. Sudėti, supilti       | S – svogūnai  | G – grietinė     |
| 2. Virti tam tikrą laiką | P – paprika   | M – miltai       |
| 3. Kepti                 | V – vanduo    | A – aliejus      |
| 4. Sumaišyti             | VŠ – vištiena | PR – Prieskoniai |
| 5. Išjungti              |               |                  |

Matas ketina virti vištieną su paprikomis pagal tokį receptą:

- Kepti svogūnus aliejuje.
- Supilti vandenį, sudėti paprikas ir vištieną.
- Virti tam tikrą laiką.
- Dubenyje sumaišyti grietinę ir miltus.
- Supilti sumaišytą grietinę ir miltus į keptuvę.
- Sudėti prieskonius.
- Virti tam tikrą laiką.
- Išjungti.

Kuri iš šių nurodymų sekų aprašo vištienos su paprikomis receptą?

| A)      | B)          | C)          | D)          |
|---------|-------------|-------------|-------------|
| 2       | 3(A, S)     | 3(A, S)     | 3(A, S)     |
| 3(A, S) | 1(P, V, VŠ) | 1(P, V, VŠ) | 1(P, V, VŠ) |
| 2       | 2           | 2           | 2           |
| 4(G, M) | 4(G, VŠ)    | 4(G, M)     | 4(G, VŠ)    |
| 2       | 1 (G, M)    | 1 (G, M)    | 1           |
| 5       | 2           | 1 (PR)      | 5           |
|         | 5           | 2           | 2           |
|         |             | 5           | 5           |

## Paaiškinimas

Teisingas atsakymas: C.

A – aiškiai netinkamas, nes pradedama nuo virimo dar prieš sudedant ingredientus.

B – ketvirtoji programos eilutė nurodo 4(G, VŠ), tačiau reikia ne vištienos, o miltų, todėl atsakymas neteisingas.

D – pagal komandą šeštoje eilutėje išjungiama viryklė, todėl toliau virti neįmanoma.

## Tai informatika!

Struktūriniam programavimui būdingi trys algoritminiai konstruktai: sekos, išrinkimas ir iteracijos. Ši užduotis yra sekos pavyzdys. Visos komandos (ir paprogramės) vykdomos iš eilės viena po kitos. Struktūrinio programavimo paradigma siekia aiškumo, kokybės per optimalų laiką, naudojantis plačiomis paprogramių galimybėmis, programų blokų struktūromis, „for“, „while“ ciklais ir kitomis priemonėmis.

Kalbant apie programavimą, paprogramės gali būti vadinamos procedūromis, funkcijomis, klasių metodais (kalbant apie objektinį programavimą), programomis arba moduliais, priklausomai nuo programavimo kalbos. Paprogramės iškviečiamos perduodant joms parametrus arba parametrų neperduodant.

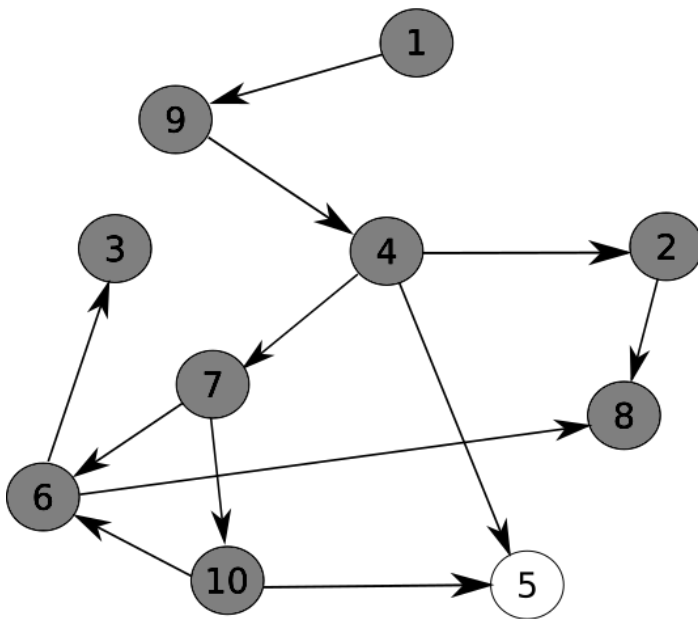
Šio uždavinio procedūroms „Sudėti“, „Maišyti“ ir „Kepti“ reikalingi skirtingi parametrai, priklausantys nuo ingredientų pasirinkimo. Ta pati paprogramė gali būti naudojama ne vieną kartą, pateikiant skirtingus duomenis (parametrus), tai labai naudinga programuojant.

Raktiniai žodžiai: seka, paprogramė, parametras, komanda.

## 9. Apsirengimo tvarka

Bebriukas Haris šiandien atsikėlė vėliau, nei planavo, ir dabar skuba į mokyklą. Mokykloje vyks renginys, lankysis žinomas architektas, todėl Hariui svarbu gerai atrodyti.

Kad apsirengtų, Haris turi atlikti 10 veiksmų. Kai kuriuos veiksmus Haris gali atlikti tuo pačiu metu, kad sutaupytų laiko ir spėtų į renginį. Tačiau kai kuriuos veiksmus jis gali atlikti tik baigęs ankstesnius. Pavyzdžiui, apsiauti batus galima tik tada, kai jau užsimautos kelnės ir kojines.



Haris gali apsirengti įvairiai derindamas veiksmus. Tačiau viena iš pateiktų sekų nepadės Hariui apsirengti. Kuri?

- A) 1, 9, 4, 2, 7, 10, 6, 8, 3, 5
- B) 1, 9, 4, 7, 10, 6, 2, 5, 8, 3
- C) 1, 9, 4, 7, 6, 5, 2, 10, 8, 3
- D) 1, 9, 4, 7, 2, 10, 6, 3, 5, 8

## Paaiškinimas

C – neteisinga seka.

10-as veiksmas turi būti atliktas prieš pradėdant 6-ą veiksmą. Visi kiti variantai teisingi.

### Tai informatika!

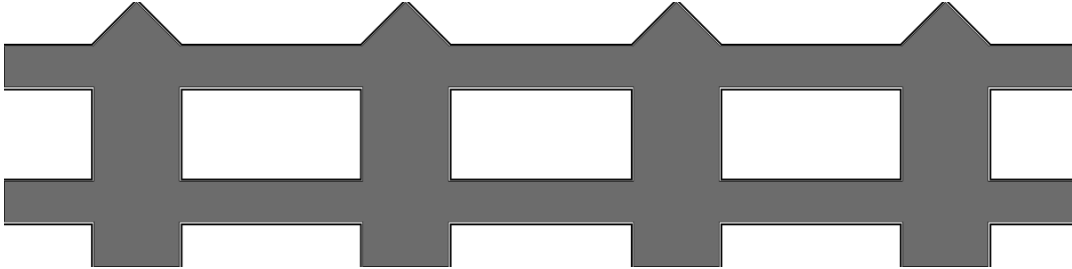
Labai dažnai užduotys turi išankstinių sąlygų. Vienas iš gerai žinomų pavyzdžių – apsirengimas. Reikia pirma apsivilkti apatinius, o tik tada mautis kelnes ir vilktis marškinius. Švarkas apsivelkamas dar vėliau. Kita vertus, nesvarbu, kada užsimausite kojines – prieš kelnes ar po jų. Kai tik tam tikros užduotys turi tarpusavio priklausomybių, jos sudaro vadinamąją dalinę tvarką. Ši tvarka dažnai pateikiama orientuotu grafu be ciklų, kur kiekviena užduotis yra viršūnė, o kiekviena išankstinė sąlyga – lankas. Jei grafas turėtų ciklą, atlikti veiksmų nepavyktų.

Norint rasti tvarką, kuri tenkintų išankstinius reikalavimus, galima naudoti topologinio rikiavimo algoritmą. Jis fiksuoja visas viršūnes, kurių išankstinės sąlygos yra tenkinamos, išrenka vieną iš jų ir sudeda visas viršūnes, kurių visos išankstinės sąlygos šiuo metu patenkintos. Tai pavyzdys godžiojo algoritmo, kurio kiekviename žingsnyje pasirenkamas geriausias variantas.

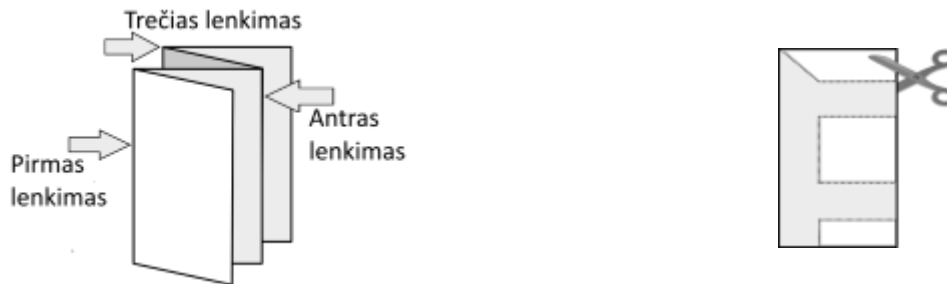
Raktiniai žodžiai: tvarka, eiliškumas, dalinė tvarka, grafas, orientuotas grafas be ciklų, topologinis rikiavimas.

## 10. Tvorelė

Norima iškarpyti štai tokią tvorelę:



Daroma šitaip: imama popieriaus juostelė ir sulankstoma zigzagu (žr. paveikslą kairėje). Šiame pavyzdyje lankstant padarytos 3 klostės (lenkimai). Paskui iškerpama figūra (paveikslas dešinėje).



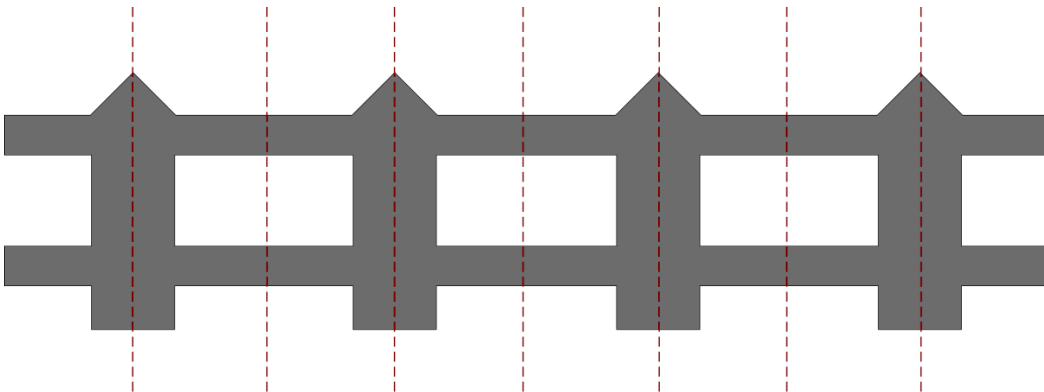
Kiek klosčių reikia padaryti, norint iškirpti uždavinio pradžioje pateiktą tvorelę?



## Paaiškinimas

Reikia padaryti 7 popierinės juostelės klostes.

Pateiktas piešinys, kuriame klostės pažymėtos raudonomis linijomis.



Uždavinys sprendžiamas ieškant simetrijos ir kartojamų veiksmų.

## Tai informatika!

Sprendžiant šį uždavinį reikalingi veiksmų kartojimo ir užduoties skaidymo į mažesnes dalis įgūdžiai.

Informatikoje kartojimo veiksmai nusakomi ciklais. Beveik visos programavimo kalbos turi ciklą. Kur kas lengviau nurodyti kompiuteriui „atlik šiuos veiksmus 100 000 kartų“ nei 100 000 kartų nurodyti atlikti „šiuos veiksmus“.

Uždavinio skaidymas į mažesnes dalis, arba dekompozicija, padeda padalyti didelę užduotį ar ją supaprastinti – išskirti pagrindinius komponentus. Šiame uždavinyje toks pagrindinis komponentas yra tvorelės dalis, apimanti dvi gretimas klostes. Programuotojai paprastai didelius, sudėtingus uždavinius skaido į mažesnius komponentus, nes mažesnes dalis lengviau programuoti, mažiau daroma klaidų, taip pat lengviau testuoti ir galima greičiau pasiekti rezultatą.

Raktiniai žodžiai: simetrija, kartojimas, dekompozicija.

## 11. Spartusis kėlimas laipsniu

Reikia apskaičiuoti  $2^{37}$ , o skaičiuotuvo įjungimo mygtukas neveikia. Norėdamas padėti, draugas pateikė tokias užuominas:

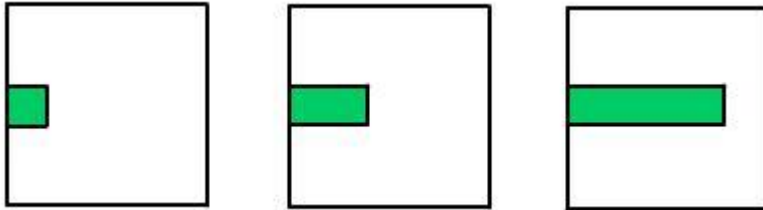
- Jei laipsnio rodiklis yra lyginis, apskaičiuok 2, pakeltą laipsniu, lygiu pusei esamo laipsnio rodiklio, tada padaugink rezultatą iš jo paties, pavyzdžiui:  $2^5 \times 2^5 = 2^{10}$ .
- Jei laipsnio rodiklis yra nelyginis, apskaičiuok 2, pakeltą laipsniu, kuris yra vienetu mažesnis už esamą, t. y., lyginis, o tada padaugink rezultatą iš 2, pavyzdžiui:  $2^{10} \times 2 = 2^{11}$ .

**Kiek daugybės veiksmų, remiantis pateiktomis užuominomis, reikia atlikti, norint apskaičiuoti  $2^{37}$ ?**

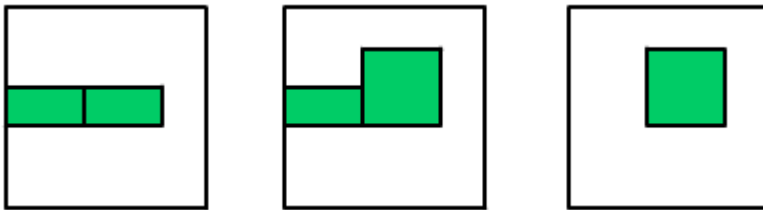


## 12. Augalo gyvenimas

Bebrui patinka augalai. Jis sukūrė vaizdinę programavimo kalbą, kurioje augalo augimas vaizduojamas paveikslėliais. Kiekvienas paveikslėlis pradedamas kvadratu  $a$ . Vaizdiniai objektai kuriami trimis operacijomis: *augti()*, *daugintis()* ir *sunykti()*. Skliaustuose nurodoma augimo kryptis (rytai, vakarai, šiaurė, pietūs). Pateikiamas penkių operacijų programos pavyzdys:



*a. augti (rytai); a. augti (rytai);*



*b, c = a. daugintis(); c. augti (šiaurė); b. sunykti ();*

Stačiakampiam objektui sukurti gali būti taikoma operacija *daugintis()* – sukuriami du padalinti per vidurį dvigubai trumpesni augalai, negu prieš tai buvęs vienas augalas. Kvadratiniam objektui sukurti operacijos *daugintis* negalima taikyti.

Bebras nori parašyti programą, kuri padėtų kairinį paveiksluką transformuoti į dešinįjį:



**Kokios turėtų būti pirmosios keturios operacijos?**

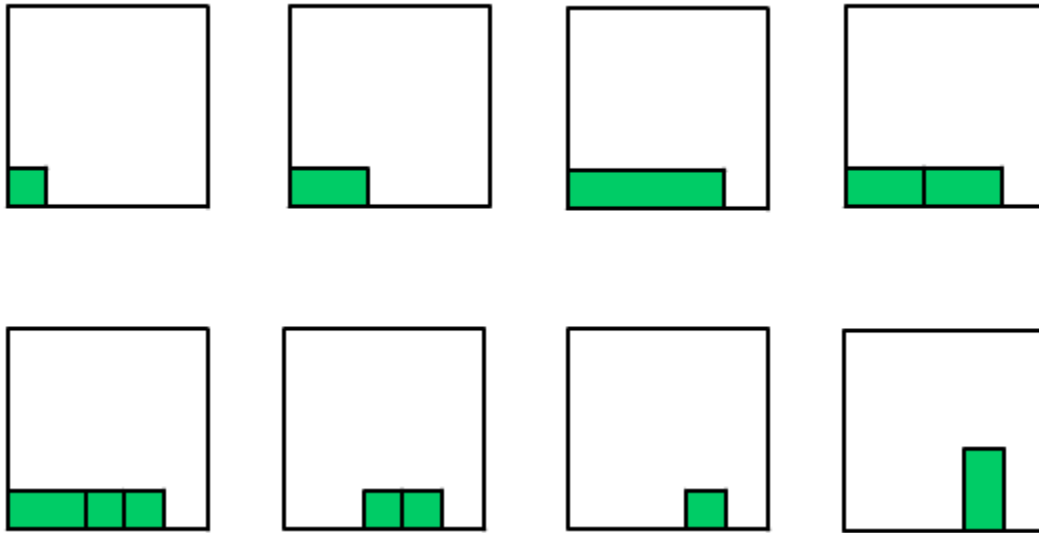
- A. *a. augti (rytai); a. augti (rytai); b, c = a. daugintis (); b. sunykti ();*
- B. *a. augti (šiaurė); a. augti (rytai); a. augti (rytai); b, c = a. daugintis ();*
- C. *a. augti (rytai); a. augti (rytai); a. augti (šiaurė); a. sunykti();*
- D. *a. augti (rytai); b, c = a. daugintis (); c. augti (šiaurė); c. augti (rytai);*

## Paaiškinimas

Teisingas atsakymas: A.

Štai, kaip veikia programa:

a.augti(rytai); a.augti(rytai); b,c = a.daugintis(); d,e = c.daugintis(); b.sunykti(); d.sunykti();  
d.augti(šiaurė);



## Tai informatika!

Kompiuterių programos, klasių bibliotekos ir programavimo kalbos dažnai remiasi intuityviomis metaforomis. Tai palengvina pradedančių programuotojų mokymąsi ir padeda greičiau perprasti abstrakčias idėjas. Pavyzdžiui, Logo grafikos sistema grindžiama judančiu vėžliuku, kuris valdomas programiniu būdu. Vėžliukas turi pieštuką, kuris piešia nueitą kelią.

Šiame uždavinyje panaudota augmenijos idėja. Augalai negali vaikščioti iš vienos vietos į kitą, jie neturi kojų ir raumenų. Tačiau augalai gali augti, daugintis, vegetuoti ir sunykti. Su šiomis operacijomis galima įgyvendinti tam tikrą judėjimą, atliekant operacijas, kurias galima pavaizduoti piešiniais.

### 13. Algoritmavimo menas

Rekursija – tai algoritmo kreipimasis į save patį. Gyvenimiškas rekursijos pavyzdys – du vienas prieš kitą sustatyti veidrodžiai. Jie vaizdą rekursiškai atspindi be galo daug kartų.

Štai rekursinio algoritmo pavyzdys:

Komanda *Piešti kvadratą*( $x, y, s$ ) nurodo kompiuteriui atlikti tokius veiksmus:

Piešti kvadratą, kurio kraštinės ilgis yra  $s$ , o jo centro koordinatės yra  $x, y$ .

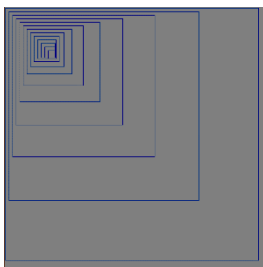
Jei kvadrato kraštinė  $s$  ilgesnė negu 2 taškai, reikia atlikti veiksmus:

*Piešti kvadratą*( $x+s/2, y, s/2$ ) (piešti mažesnį kvadratą dešinėje);

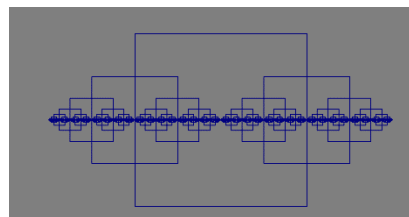
*Piešti kvadratą*( $x-s/2, y, s/2$ ) (piešti mažesnį kvadratą kairėje).

Kuris iš pateiktų piešinių buvo sukurtas pagal komandą *Piešti kvadratą*?

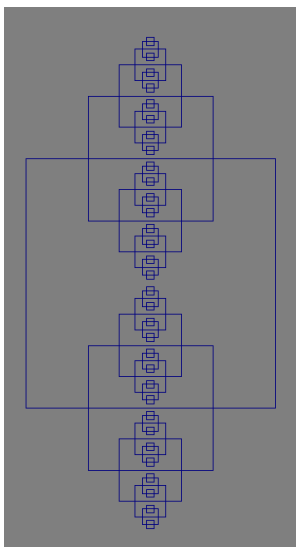
A.



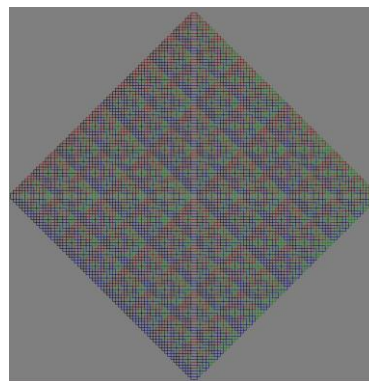
B.



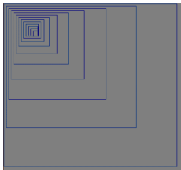
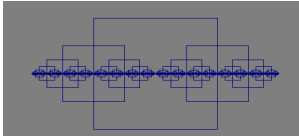
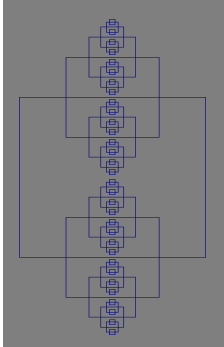
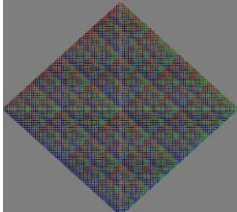
C.



D.



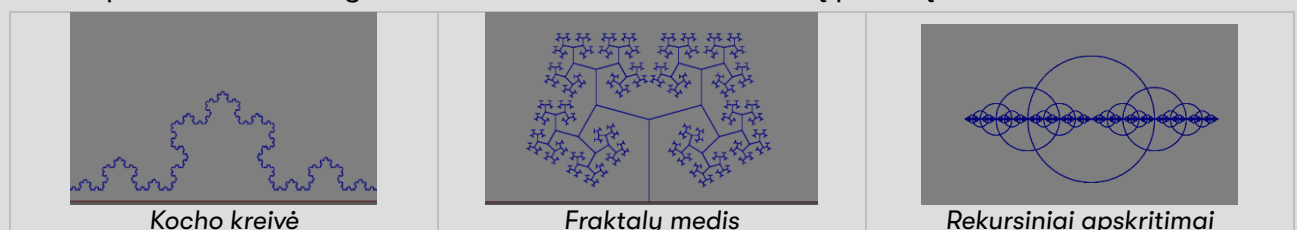
## Paaiškinimas

|    |   |  |
|----|---|--|
| A. |    | <p>Neteisinga.</p> <p>Kvadratai nupiešti, bet nėra kvadratų kairėje ir dešinėje.</p>   |
| B. |    | <p>Teisinga.</p> <p>Yra vieno didelio kvadrato piešinys ir ant kiekvieno kvadrato šoninių kraštinių po vieną dvigubai mažesnį kvadratą.</p>                          |
| C. |    | <p>Neteisinga.</p> <p>Yra vieno didelio kvadrato piešinys, bet mažesni kvadratai yra ant didesniųjų kvadratų viršutinių ir apatinių kraštinių, o ne ant šoninių.</p> |
| D. |  | <p>Neteisinga.</p> <p>Yra vieno didelio kvadrato piešinys, bet mažesni kvadratai yra ant kiekvienos didesniojo kvadrato kraštinės, ne tik ant šoninių.</p>           |

## Tai informatika!

Daugelis programavimo kalbų turi rekursiją. Jų procedūros gali kreiptis pačios į save, kad galėtų išspręsti šiek tiek supaprastintą uždavinį. Tai gali būti naudinga dirbant su sudėtingomis duomenų struktūromis, pavyzdžiui, sąrašais.

Vienas iš informatinio mąstymo rezultatų yra rekursiniai fraktalai. Tai sudėtingi ir įdomūs piešiniai, kuriems generuoti pakanka trumpų, tik kelių eilučių ilgio programų. Kai kurie tokiomis programomis sukurti piešiniai laikomi algoritmavimo menu. Štai keletas tokių piešinių:



Skaitmeninėje grafikoje, naudojantis algoritmavimo menu arba fraktalų geometrija, sukuriama skaitmeniniai piešiniai. Skirtingai nuo klasikinės geometrijos, čia gaunami ne visi taisyklingi vaizdai, primenantys gamtos objektus: snaiges, medžius, lapus. Visa tai gali būti naudojama kompiuteriniuose modeliuose, kuriant dinaminis ir tikroviškus objektus.

Raktiniai žodžiai: fraktalai, rekursija, algoritmai.

## 14. Šokinėjanti beždžionė

Lapuotis medis  stovi tarp dviejų spygliuočių medžių  ir dviejų palmių  .



Medžiuose padėti penkių skirtingų rūšių bananai, tarkim, P, Q, R, S, T – viename medyje viena rūšis. Beždžionė šokinėja tarp medžių, kas kartą suėsdama po vieną bananą. Ji sugaišta (šokdama ir ėsdama bananą) 3 sekundes, kol iš lapuočio medžio persoka į bet kurį kitą, 2 sekundes – iš spygliuočio į palmę arba atvirkščiai, 7 sekundes – iš vieno spygliuočio į kitą spygliuotį arba iš vienos palmės į kitą palmę.

Beždžionė šokinėja ir ėda bananus tokia tvarka: P, Q, S, R, T, R, P.

**Kokios rūšies bananai gali būti padėti lapuotyje medyje, siekiant, kad visas beždžionės sugaištas laikas būtų trumpiausias.**

- A) P arba Q, arba T
- B) P arba S, arba T
- C) Q arba S, arba T
- D) Q arba R, arba S



## Paaiškinimas

Teisingas atsakymas: P arba S, arba T.

Sekoje P, Q, S, R, T, R, P ėdami visų rūšių bananai, atliekami šeši šuoliai P–Q, Q–S, S–R, R–T, T–R, R–P. Siekiant sugaišti mažiausiai laiko, reikia atlikti keturis 2 sekundžių ir du 3 sekundžių trukmės šuolius. Bendra visų šuolių trukmė – 14 sekundžių.

R yra keturiuose šuoliuose: S–R, R–T, T–R, R–P. R–T ir T–R trunka tiek pat laiko. Į R arba iš R yra mažiausiai trys skirtingi šuoliai. Todėl R negali būti lapuotis medis, nes trys šuoliai truktų 9 sekundes ( $3 \times 3$ ) ir sugaištas laikas nebūtų trumpiausias. Toks samprotavimas atmeta D atsakymą.

T gali būti lapuotis medis, nes T–R ir R–T šuoliai truktų po 3 sekundes, o kiti šuoliai – po 2 sekundes.

Jei Q yra lapuotis medis, tai šuoliai P–Q ir Q–S trunka po 2 sekundes. Tokiu atveju bendras visų šuolių laikas būtų minimalus, jei ir kiti šuoliai – S–R, R–T, T–R ir R–P – būtų 2 sekundžių trukmės. Tačiau tai yra neįmanoma ir tuo galima įsitikinti pažvelgus į paveikslėlį, kuriame medžiai pažymėti raidėmis S, R ir T. Taigi Q negali būti lapuotis medis, todėl telieka galimybės P arba S, arba T.

## Tai informatika!

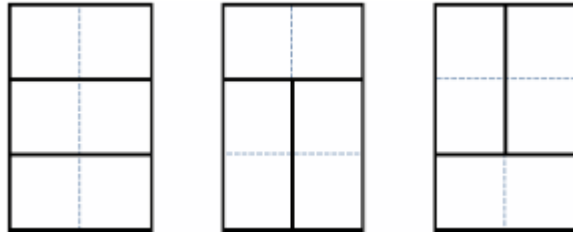
Sprendami šį uždavinį iš pradžių turime pasitelkti dedukcinį metodą (arba dedukcinį samprotavimą), kad iš bendro atvejo išskirtume atskirus, logiškai pagrįstus atvejus. Taikome ir simetrijos principą. Taip pat siekiame optimalaus sprendimo, kad sunaudotume kuo mažiau resursų. Šiuo atveju optimizuojame laiko sąnaudas. Informatikai nuolat sprendžia optimizavimo uždavinius, tam sukurta nemažai standartinių algoritmų.

Raktiniai žodžiai: dedukcinis samprotavimas, optimizavimas.

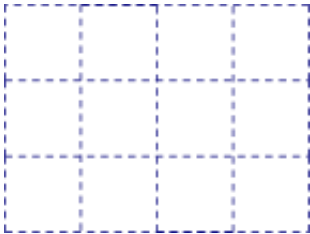
## 15. Plytelių dėjimas

Bebras turi daug vienodų plytelių, panašių į domino kauliukus, t. y., jų ilgis yra dvigubai didesnis nei plotis.

Bebras deda plyteles į dėžutę. Plytelė dedama tik vertikaliai arba horizontaliai. Pavyzdžiui, jei dėžutės matmenys  $3 \times 2$ , tai plyteles galima sudėti 3 skirtingais būdais:



Keliais skirtingais būdais galima sudėti plyteles į didesnę, t. y.,  $3 \times 4$ , dėžutę?



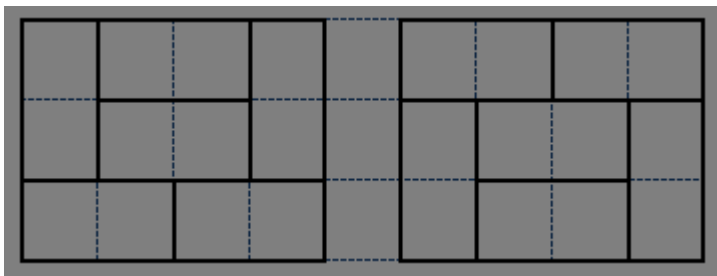
- A) 6
- B) 8
- C) 9
- D) 11

## Paaiškinimas

Teisingas atsakymas: 11.

Pirmiausia atkreipkime dėmesį, kad didesnę dėžutę galime padalyti į dvi mažesnes ir pritaikyti anksčiau nagrinėtą atvejį. Kadangi į mažesnę, t. y.,  $3 \times 2$ , dėžutę plyteles dėjome trimis skirtingais būdais, tai plyteles dėdami į didesnę dėžutę šiuos būdus turėsime kombinuoti po du. Gausime  $3 \times 3 = 9$  būdus.

Pagalvoję pastebėsime, kad galimi dar du papildomi būdai:



Taigi plyteles į  $3 \times 4$  dėžutę galima sudėti 11 skirtingų būdų, t. y.  $9 + 2 = 11$ .

## Tai informatika!

Plytelių dėjimas grindžiamas geometrinio plano sudarymu arba mozaikos kūrimu. Domino kauliukas (plytelė) yra geometrinė figūra, sudaryta iš dviejų kvadratų. Plytelių dėjimas į dėžutę – tai kurios nors srities uždengimas viena ar keliomis figūromis, kai nepaliekama plyšių ir plytelės viena kitos neuždengia.

Ieškant sąryšio tarp mažesnės, t. y.,  $3 \times 2$ , ir didesnės, t. y.,  $3 \times 4$ , dėžučių, būtinas indukcinis samprotavimas. Toliau reikia ištirti, ar dedant plyteles į didesnę dėžutę neatsirado papildomų galimybių – taip išsiaiškinami dar du skirtingi plytelių dėjimo būdai.

Raktiniai žodžiai: geometrinis išdėstymas, indukcija, domino, mozaika.

## 16. Raidžių žaidimas

Bebras Bronius žaidžia raidžių sekų žaidimą. Jis yra sukūręs keletą taisyklių, kaip sudaryti naujas raidžių sekas:

1. Kiekviena seka pradedama raide S.
2. Tinkamą seką sudaro tik raidės A ir (arba) B  
(sudarant seką gali būti pasitelkiamos raidės S ir X.)
3. Raidė S gali būti keičiama į AX.
4. Raidė X gali būti keičiama į AXB.
5. Raidė X gali būti keičiama į B.

Pavyzdžiui, raidžių seka AABB gali būti gaunama tokiu būdu:

S (1 taisyklė) → AX (3 taisyklė) → AAXB (4 taisyklė) → AABB (5 taisyklė).

**Kuri iš raidžių sekų sudaryta pagal šio žaidimo taisykles?**

- a) AX
- b) A
- c) AAAABBBB
- d) AABBAABB

## Paaiškinimas

- (a) seka neteisinga, nes pagal 2 taisyklę seką gali sudaryti tik raidės A ir (arba) B.
- b) seka neteisinga, nes trumpiausių seką sudaro dvi raidės:  $S \rightarrow AX \rightarrow AB$  (kiekviena taisyklė seką daro ilgesnę už ankstesnę).
- c) AAAABBBB yra teisinga seka ( $S \rightarrow AX \rightarrow AAXB \rightarrow AABB$ ).
- d) seka neteisinga, nes nėra taisyklės, pagal kurią raidę A būtų galima užrašyti dešinėje AB.

## Tai informatika!

Pateiktomis taisyklėmis kuriama laisvojo konteksto kalba (angl. *context free language*). Šią kalbą atitinka tokia gramatikos taisyklių aibė:

1.  $S \rightarrow AX$
2.  $X \rightarrow AXB$
3.  $X \rightarrow B$

Laisvojo konteksto kalbos generavimą galima nusakyti taip: paimamas vienintelis simbolis S, pasirenkama taisyklė, pagal kurią kairėje sekos pusėje yra S. Šis simbolis keičiamas eilute, esančia ženklo „→“ dešinėje. Veiksmai kartojami, kol eilutėje nelieka simbolių, pagal taisyklę esančių ženklo „→“ kairėje.

Ženklo „→“ kairėje esantys simboliai vadinami neterminaliniais, o ženklo „→“ dešinėje – terminaliniais simboliais.

Neterminaliniai simboliai žymi baigtinio automato būsenas. Baigtinis automatas veikia kaip kalbos atpažinimo įtaisas.

Galima sukurti nedeterministinį automatą, kuriuo nustatoma, ar seka yra žodis.

## 17. Rask vagį!

O, ne! Šiandien iš muziejaus buvo pavogtas ypatingas mėlynasis deimantas: vagis vietoj jo paliko pigią žalios spalvos deimanto imitaciją.



Šiandien deimanto ekspoziciją aplankė 2000 žmonių. Jie buvo įėję į deimanto ekspozicijos kambarį vienas po kito. Inspektorius Bebras privalo rasti vagį, todėl apklausia kai kuriuos iš šių lankytojų. Jis turi 2000 ekspoziciją aplankusių žmonių sąrašą, sudarytą ta eile, kuria jie ėjo į deimanto kambarį. Inspektorius kiekvienam lankytojui pateikia tą patį klausimą: „Jūsų matytas deimantas buvo žalios ar mėlynos spalvos?“ Visi turėtų atsakyti teisingai, išskyrus vagį, kuris sakys, kad deimantas jau buvo žalias.

Inspektorius Bebras yra labai protingas. Jis naudos strategiją, kad žmonių, kuriuos reikia apklausti, skaičius būtų kuo mažesnis.

**Kurį iš pateiktų teiginių jis gali suformuluoti nemeluodamas?**

- A) Garantuoju, kad rasiu vagį apklausęs mažiau kaip 20 žmonių.
- B) Neužteks apklausti 20 žmonių (nebent man pasiseks), bet tikrai pavyks atlikti darbą apklausus mažiau kaip 200 lankytojų.
- C) Tai sudėtingas darbas: man reikės apklausti ne mažiau kaip 200 žmonių, gali būti, kad net 1999.
- D) Nieko negaliu prognozuoti. Jei nepasiseks, man gali tekti apklausti kiekvieną lankytoją.

## Paaiškinimas

Teisingas atsakymas: A.

Inspektorius Bebras turės apklausti tik nedidelę dalį lankytojų. Pirmiausia jis turi sunumeruoti lankytojus nuo 1 iki 2000 tokia eile, kokia jie buvo įėję į kambarį.

Tuomet jis kartodamas atlieka tokius žingsnius:

- Inspektorius Bebras prašo lankytoją Nr. 1000 atsakyti, kokios spalvos buvo deimantas, kai šis jį matė.
- Jei jis atsako, kad mėlynas, daroma išvada, kad vagis atėjo po 1000-ojo lankytojo ir turi eilės numerį nuo 1001 iki 2000.
- Jei jis atsako, kad deimantas buvo žalias, tai vagis turės eilės numerį nuo 1 iki 1000. (Atkreipkite dėmesį, kad deimantą galėjo pavogti lankytojas, kurio numeris 1000!)

Abiem atvejais potencialių vagių skaičius sumažinamas nuo 2000 iki 1000, kitaip tariant, per pusę.

Tada inspektorius Bebras užduoda klausimą asmeniui, kuris yra likusio sąrašo viduryje (t. y., pirmu atveju – numeris 1500, antru atveju – numeris 500). Tokiu būdu inspektorius vėl galės padalyti įtariamųjų sąrašą per pusę.

Taip skaičiuodamas, inspektorius galės sumažinti įtariamųjų skaičių iki 500, 250, 125, 63, 32, 16, 8, 4 ir 2. Kai liks 2 įtariamieji, jis turės paklausti pirmojo. Jei jis atsakys „žalias“, jis ir yra vagis. Priešingu atveju vagis yra kitas įtariamasis. Taip inspektorius ras deimantą pavogusį asmenį apklausęs tik 11 žmonių.

### Tai informatika!

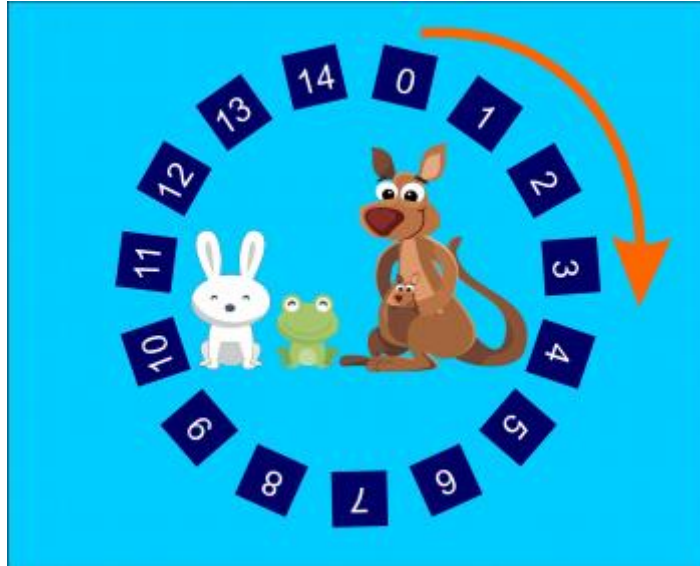
Aibės kartotinio dalijimo per pusę technika populiari informatikoje. Vienas žinomiausių pavyzdžių – dvejetainė paieška, kuria ieškoma tam tikro įrašo sutvarkytame sąraše.

Mūsų atveju ieškomas pirmas žalias elementas mėlynų ir žalių elementų sąraše, kur visi mėlyni elementai yra sąrašo pradžioje. Faktas, kad lankytojų sąraše yra vagis, kuris meluos atsakydamas į klausimą, mums leidžia ieškoti spalvos pasikeitimo. Atkreipkime dėmesį, kad problema būtų neišsprendžiama, jei iš anksto nežinotume, kokios spalvos deimanto imitacija vagis pakeitė originalų deimantą.

Raktiniai žodžiai: dvejetainė paieška.

## 18. Šuolių čempionatas

Trys draugai – zuikis, varlė ir kengūra – dalyvauja šuolių čempionate. Šuolių trasa sudaryta iš 15 šuolių aikštelių, įrengtų ratu ir sunumeruotų nuo 0 iki 14.



Per sekundę zuikis atlieka 4 šuolius, varlė – 6, o kengūra – 10 šuolių. Iš pradžių visi trys stovi vienoje aikštelėje, pažymėtoje 0.

**Po kiek sekundžių visi draugai vėl susitiks vienoje iš aikštelių?**



## Paaiškinimas

Teisingas atsakymas: 30.

Per sekundę kiekvienas iš draugų atlieka nurodytą skaičių šuolių: zuikis – 4, varlė – 6, kengūra – 10. Kadangi jų šuolių skaičiai nėra vienas kito kartotiniai, tai visi trys draugai galės susitikti tik pradinėje aikštelėje. Reikia atkreipti dėmesį, kad uždavinį galima išspręsti taikant mažiausio bendrojo kartotinio – 30 – algoritmą.

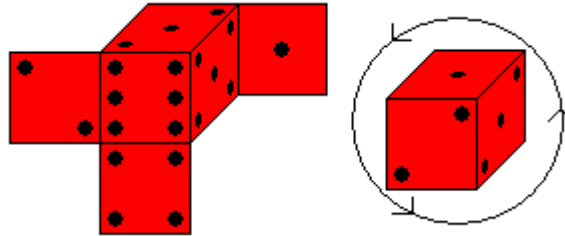
## Tai informatika!

Sprendžiant uždavinį, pirmiausia būtina įvertinti situaciją ir nustatyti skaičiavimų pobūdį, galimą metodą, ar nėra žinomo algoritmo ir pan. Jei iš karto nenumatoma mažiausio bendrojo kartotinio algoritmo galimybė, uždavinį galima spręsti modeliuojant šuolių kartojimą. Įžvelgus galimybę taikyti mažiausio bendrojo kartotinio algoritmą, uždavinys tampa lengvai išsprendžiamas matematiškai būdu.

Raktiniai žodžiai: žingsnis (šuoelis), kartojimas, mažiausias bendrasis kartotinis, modeliavimas.

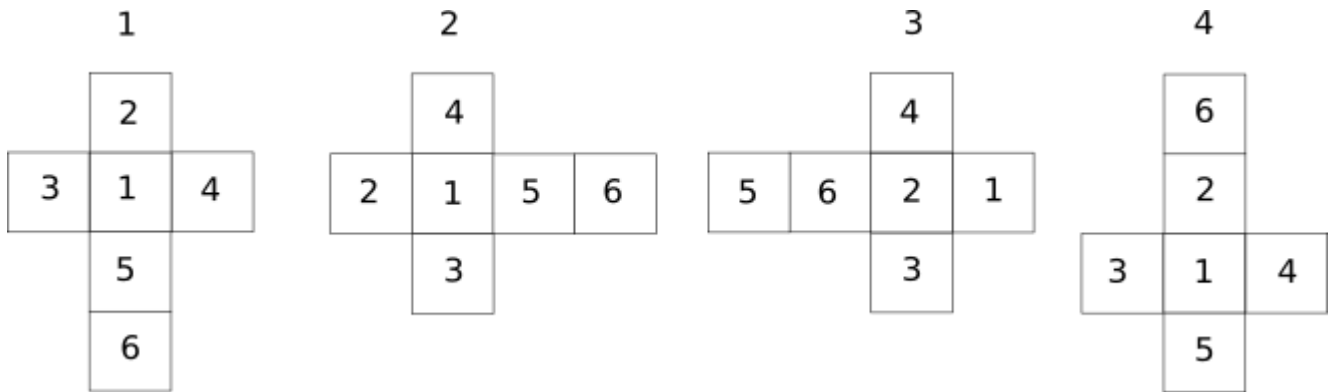
## 19. Penki kauliukai

Vaikai nori žaisti YATZY žaidimą, kuriam reikia 5 vienodų kauliukų. Žaidimo kauliukus reikia pasidaryti iš popieriaus lapų. Nupiešiama visų šešių kauliuko sienų išklotinė, pavaizduojami taškai. Vaikai iškarpo ir sulanksto išklotinę – štai ir kauliukas!

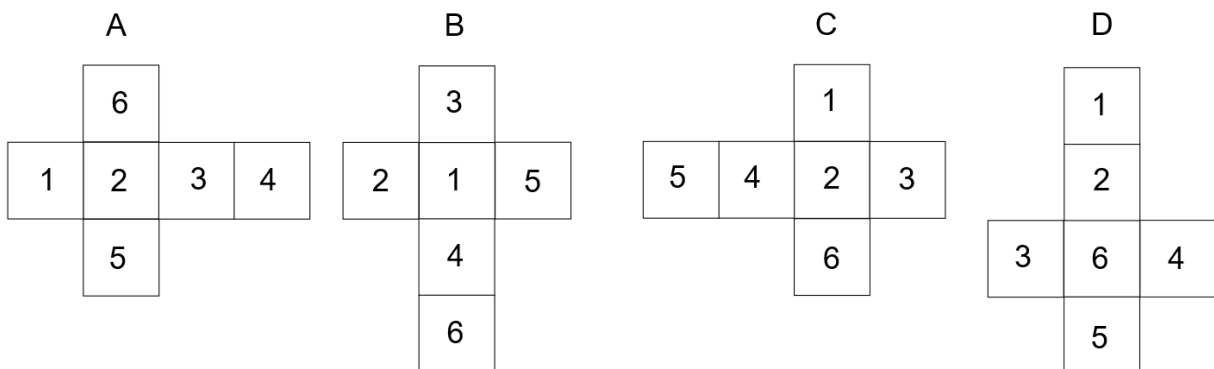


Žaidimo kauliuko savybė: priešingų sienų taškų suma lygi 7 ( $1 + 6$ ,  $4 + 3$  ir  $5 + 2$ ). Be to, sienos su 1, 2, 3 taškais, žvelgiant iš kampo, turi būti išdėstytos prieš laikrodžio rodyklę.

Vaikai padarė 4 kauliukus, bet dar trūksta penktojo:



Kuri iš šių išklotinių tinka trūkstamam kauliukui?



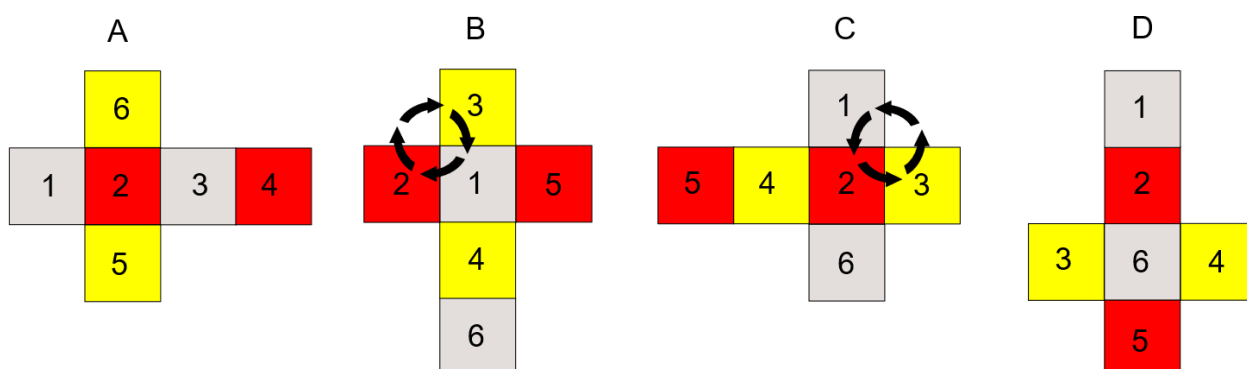
## Paaiškinimas

Išklotinė C yra vienintelė, tinkanti kauliukui pagaminti. Priešingų sienų taškų suma lygi 1 (paveikslėlyje pažymėta ta pačia spalva), o sienos su 1, 2, 3 taškais išdėstytos prieš laikrodžio rodyklę.

A – neteisingas atsakymas, nes priešingų sienų taškų suma nėra lygi 7.

B – taip pat neteisingas atsakymas, nes sienos su 1, 2, 3 taškais nėra išdėstytos prieš laikrodžio rodyklę, o atvirkščiai.

Sukonstravęs trimatį atsakymo D kauliuką ir pažvelgęs į sienas su 1, 2, 3 taškais, pamatysi, kad šie taškai išdėstyti pagal laikrodžio rodyklę, o ne prieš, kaip reikalaujama.



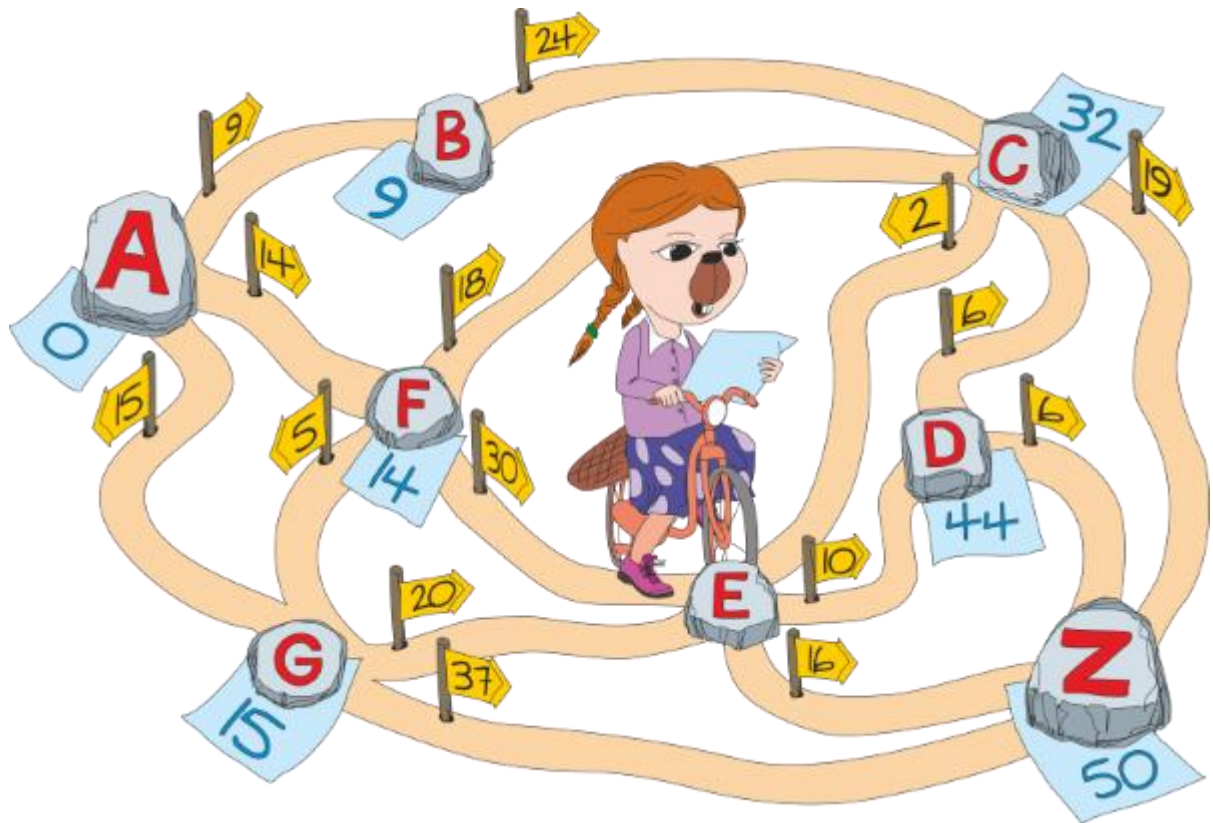
## Tai informatika!

Žaidimo kauliuko padarymas grindžiamas vaizdų atpažinimu. Kauliuko išklotinė – dvimatis vaizdas, iš jos gaunamas trimatis objektas. Sukurta įvairių būdų, kaip trimačius objektus kuo tiksliau perteikti dvimačiais vaizdais. Dažnai sprendžiamas uždavinys – kaip iš daugelio nuotraukų nustatyti, kuriose iš jų yra tas pats žmogus.

Raktiniai žodžiai: vaizdų atpažinimas, žaidimo kauliukas, savybė, išklotinė.

## 20. Trasų tyrinėjimas

Bebrė Onutė yra dviratininkė. Ji tyrinėja vienakryptes trasas, einančias per kaimus, pažymėtus raidėmis nuo A iki Z. Nurodytos visų trasų kryptys ir ilgiai. Šie duomenys pateikti geltonose vėliavėlėse.



Po kiekvienos kelionės bebrė po akmeniu palieka žymą: mėlyną raštelį su užrašytu tam tikru skaičiumi.

**Ką reiškia skaičiai, surašyti mėlynuose lapeliuose, paliktuose po akmenimis?**

- A) trumpiausias atstumas nuo kaimo A iki kito kaimo, aplankant kuo mažiau kaimų;
- B) trumpiausias atstumas nuo kaimo A iki kito kaimo;
- C) trumpiausias atstumas nuo kaimo A iki kito kaimo, kiekvienoje sankryžoje pasukant į kairę;
- D) trumpiausias atstumas nuo kaimo A iki kito kaimo, kiekvienoje sankryžoje pasukant į dešinę.

## Paaiškinimas

Norint rasti teisingą atsakymą, atstumai iki kiekvieno kaimo, atsižvelgiant į skirtingas nuorodas nuo A iki D, turi būti skaičiuojami taip:

A – neteisingas, nes  $D = 45$ ,  $Z = 52$ ;

C – neteisingas, nes  $C = 33$ ,  $D = 45$ ,  $Z = 52$ ;

D – neteisingas, nes  $C = 51$ ,  $D = 45$ ,  $Z = 52$ .

Taigi skaičius, užrašytas mėlyname lapelyje, nurodo trumpiausią atstumą nuo kaimo A iki kito kaimo (B atsakymas).

Ieškant trumpiausio kelio, galima naudoti Dijkstra'os algoritmą:

1. Kiekvienam atstumui tarp sankryžų (kaimų), pradedant nuo pradinio kaimo A, priskirti nulines reikšmes.
2. Pažymėti, jog A yra pradinė sankryža.
3. Pradedant nuo pradinės sankryžos apskaičiuoti visus atstumus iki kiekvienos neaplankytos sankryžos. Apskaičiuoti kiekvienos gretimos sankryžos visus atstumus iki kitų sankryžų ir palyginti naujus rezultatus su esamais, o pradinės padėtį suteikti mažiausią reikšmę turinčiai sankryžai.
4. Atlikus visus skaičiavimus pažymėti, jog buvusi pradinė sankryža jau yra aplankyta. Aplankytoji sankryža nebus daugiau lankoma.
5. Pasirinkti neaplankytą sankryžą su mažiausia reikšme, pažymėti, jog ji yra pradinė sankryža ir pereiti prie 3-io žingsnio.
6. Jeigu paskutinė sankryža (Z) tampa aplankytąja, tada sustoti. Algoritmas baigtas.

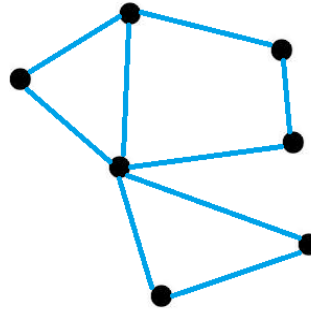
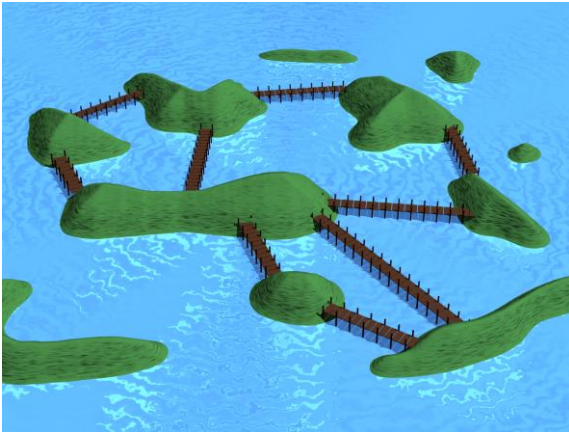
## Tai informatika!

Trumpiausio kelio (maršruto) radimo uždavinys yra viena iš svarbių informatikos sąvokų. Trumpiausio kelio algoritmas taikomas atstumui tarp vietovių rasti, pavyzdžiui, maršruto paieškai „MapQuest“ ar „Google Maps“ svetainėse. Dijkstra'os algoritmas yra vienas populiariausių algoritmų, kuriuo ieškoma trumpiausio maršruto. Šiame algoritme priskiriamos pradinės atstumų reikšmės, o tada pažingsniui ieškoma geresnių sprendinių. Sujungus vietas keliais, gaunamas grafas – labai svarbi duomenų struktūra.

Raktiniai žodžiai: algoritmas, trumpiausio kelio problema, Dijkstra'os algoritmas, grafas.

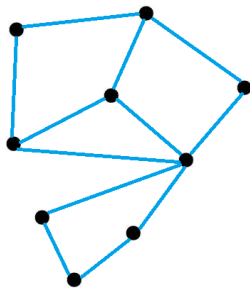
## 21. Salos ir tiltai

Bebrų bendruomenės gyvena salose ir nori tarp jų nutiesti tiltus. Inžinierius pateikė schemą, kurioje taškai žymi salas, o linijos – jas jungiančius tiltus.

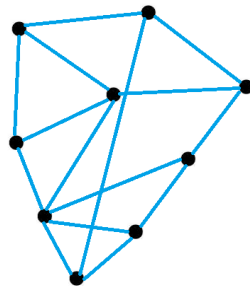


Tačiau bebrai pageidauja schemos, kurioje taškai vaizduotų tiltus, o linijos – salas. Kaip atrodytų tokia schema?

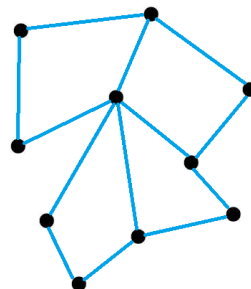
A)



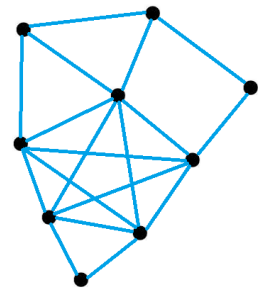
B)



C)



D)



## Paaiškinimas

Teisingas atsakymas: D.

C schemą iš karto atmetame, nes tiltų yra 9, o šioje schemoje 10 viršūnių (salų).

Toliau pastebime, kad yra vienas tiltas, iš kurio tiesiai galime pasiekti 6 tiltus – taip tereikia tik B ir D schemose. Toliau galime ieškoti tiltų, iš kurių tiesiogiai galima pasiekti 5 tiltus. Tačiau paprasčiau pastebėti, kad yra 2 tiltai, iš kurių tiesiogiai galima pasiekti tik 2 kitus tiltus – to nėra B schemoje. Vadinas, belieka, jog teisinga D schema.

## Tai informatika!

Realių situacijų modeliavimas grafais tapo itin reikšmingas, atsiradus kompiuteriams ir daugeliui grafų skaičiavimo algoritmų. Algoritmų, kurie apdorotų grafus, kūrimas yra viena iš daug dėmesio sulaukiančių informatikos sričių. Informatikoje grafais vaizduojami komunikacijų tinklai, duomenų organizavimas, skaičiavimo procesai įrenginiuose, informacijos srautai ir pan. Pavyzdžiui, tinklalapio ryšių struktūrą pavaizdavus grafu, galima optimizuoti paiešką ir atlikti kitokius veiksmus. Kuriant ar taikant grafų algoritmus, dažnai reikia transformuoti turimą grafą – keisti viršūnes ir briaunas vietomis. Tam sukurta formalių metodų. Grafo transformavimo procesas reikalauja gana nemažų abstrahavimo pastangų.

Raktiniai žodžiai: abstrahavimas, grafas, modeliavimas, transformavimas.

## 22. Funkcijų skaičiavimas

Bebro kompiuteris apdoroja informaciją naudodamas tik keletą operacijų:

- $(R f (x_1, x_2, \dots, x_n))$  pateikia rezultatą  $x_1 f x_2 f \dots f x_n$   
Pavyzdžiui,  $(R + (1, 2, 3, 4))$  pateikia  $1 + 2 + 3 + 4$ , t. y., 10.
- $(M f (x_1, x_2, \dots, x_n))$  pateikia rezultatą  $(f(x_1), f(x_2), \dots, f(x_n))$   
Pavyzdžiui, jei  $q(x) = -x$ , tada  $(M q (1, 2, 3, 4))$  pateikia rezultatą  $(-1, -2, -3, -4)$ .  
(f yra funkcija, o  $x_i$  – bet kurie skaičiai)

Kompiuteris gali vartoti bet kurias kitas funkcijas.

Pavyzdžiui, imkime  $t(x) = 3x + 2$ , tada  $(M t (1, 2, 3))$  pateiks rezultatą  $(5, 8, 11)$ .

Tarkime:  $t(x) = 3x + 2$  ir  $q(x) = -x$ .

**Kokia reikšmę gausime, atlikę šią operaciją:**

$(R + (R + (M t (0, 2, 4))) (R + (M q (M t (3, 5))))))$

- A. 7            B. 0            C. -7            D. -4



## Paaiškinimas

Tarpinių skaičiavimų rezultatai apjungiami ir gaunamas galutinis funkcijų taikymo rezultatas (pabrauktoji dalis reikalinga kitai eilutei gauti):

$$\begin{aligned} & (R + (R + \underline{(M t (0, 2, 4))}) (R + (M q \underline{(M t (3, 5))}))) \\ \Rightarrow & (R + \underline{(R + (2, 8, 14))}) (R + \underline{(M q (11, 17))}) \\ \Rightarrow & (R + 24 \underline{(R + (-11, -17))}) \\ \Rightarrow & \underline{(R + 24 -28)} \\ \Rightarrow & -4 \end{aligned}$$

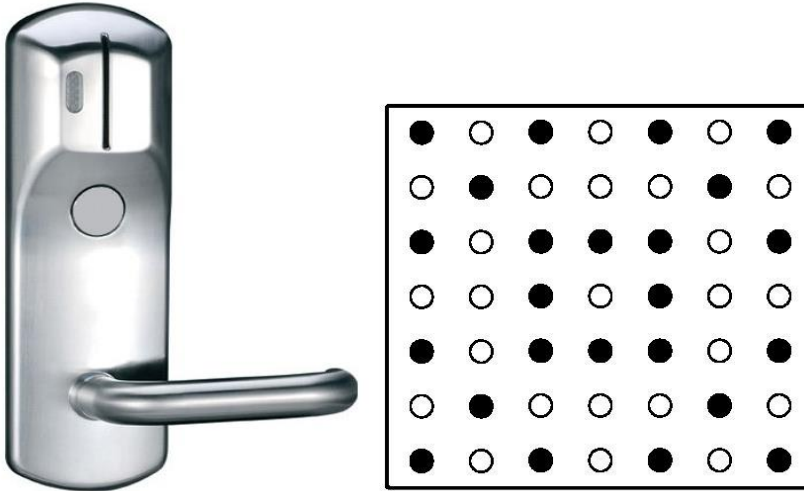
## Tai informatika!

Šis uždavinys paaiškina, kaip funkcija taikoma reikšmių aibei ir kaip atliekant operacijas mažinamas reikšmių skaičius, kol gaunamas rezultatas. Funkcijų įdėjimas viena į kitą ir jų reikšmių aibės mažinimas – pagrindiniai funkcinio programavimo konceptai. Pavyzdžiui, siekiate rasti aukščiausių užduočių įvertinimą. Vienas iš būdų tai formalizuoti yra mažinti kiekvienos eilutės reikšmes (suskaiciuojant užduočių įvertinimų sumas) ir tuomet taikyti maksimumo funkciją. Šios dvi funkcijos taikomos viena po kitos ir taip gaunamas reikiamas rezultatas.

Raktiniai žodžiai: funkcija, mažinimas, informacijos apdorojimas.

### 23. Viešbučio raktas

Bebrų miesto viešbutyje „Didysis Bebras“ įrengta nauja raktų sistema. Kiekvienas svečias gauna kvadratinę kortelę, su kuria atrakina savo kambarį. Visose kortelėse pažymėta po 7 x 7 taškų, kuriuose gali būti išmušamos skylutės.



Kambario spynoje įtaisytas kortelių skaitytuvas. Jis patikrina, ar kortelė priklauso to kambario svečiui.

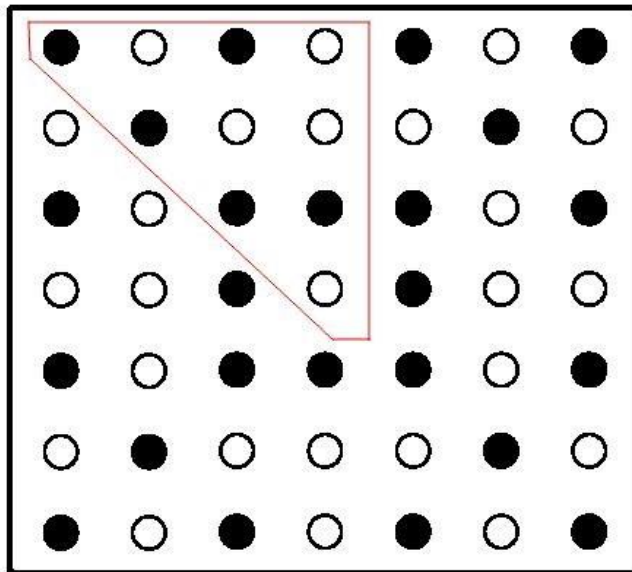
Kortelės simetriškos, todėl nesvarbu, kuriuo šonu ir kuria puse jos kišamos į skaitytuvą.

**Kiek daugiausia galima pagaminti skirtingų kortelių?**

- A. 16
- B. 49
- C. 1024
- D. 65536

## Paaiškinimas

Teisingas atsakymas: C.



### Tai informatika!

Kortelėse yra užkoduoti parametrai, kuriuos apdoroja kortelių skaitytuvas, tada durys atrakinamos arba neatrakinamos. Šiame uždavinyje kortelių skaitytuvas nuskaito kodus iš bet kurios kortelės pusės.

Kortelių su skirtingais taškų variantais gaminimas gali būti palygintas su kodavimu programavime. Kodavimas – veiksmas, kuriuo objektui suteikiamas kodas. Koduojama dėl techninių priežasčių (pavyzdžiui, rašytinis tekstas koduojamas dvejetainiais ženklais, kad jį būtų galima pateikti kompiuteriui) arba siekiant apsaugoti duomenis. Programos iškviečia ir perduoda duomenis parametrais funkcijoms, procedūroms arba programų moduliams, kur šie duomenys apdorojami ir atgal pateikiamos tam tikros reikšmės (rezultatai) arba atliekami tam tikri veiksmai.

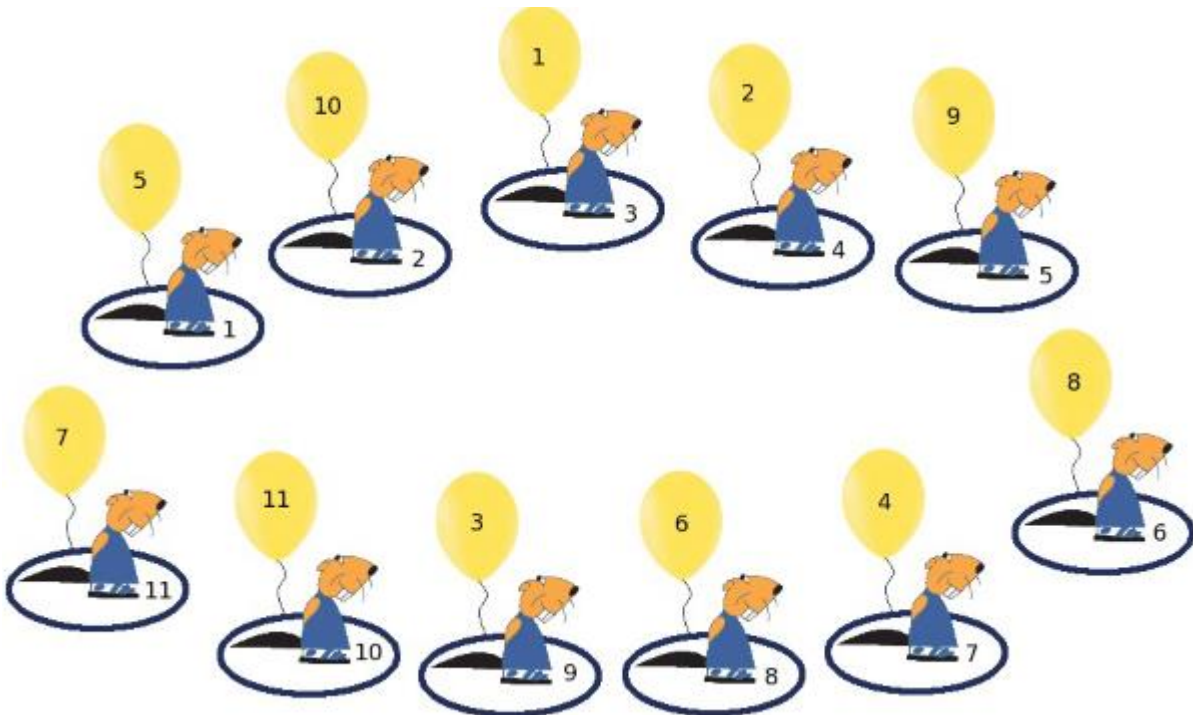
Masyvas informatikoje yra sistemingai sutvarkytų objektų duomenų struktūra, turintis indeksus ir reikšmes. Atitinkamam masyvo elementui priskiriama atitinkama reikšmė arba ji gali būti nepriskirta. Masyvai gali būti indeksuojami skaičiais arba raidėmis su tam tikrais kitais leistiniais simboliais (tokiu atveju jie vadinami asociatyviaisiais masyvais).

Šiame uždavinyje (šnekant programavimo terminais) galėtų būti funkcija, kuriai perduodamas kodas, pateiktas dvimačiu masyvu, kurio indeksai būtų skaičiai, reiškiantys tam tikras pozicijas, kur yra skylutės, o reikšmės galėtų būti loginio kintamojo reikšmės „false“ („netiesa“ – nėra skylutės) arba „true“ („tiesa“ – yra skylutė). Šios funkcijos rezultatas būtų viena reikšmė: „true“ arba „false“ (loginis kintamasis „tiesa“ arba „netiesa“), o programa, gavusi šią reikšmę, atrakintų arba neatrakintų spyną.

Raktiniai žodžiai: kodavimas, loginis kintamasis, funkcija, masyvas.

## 24. Kelionė diskais

Vienuolika bebrų pradeda žaidimą stovėdami ant vienuolikos diskų – po vieną ant kiekvieno disko. Diskai sunumeruoti nuo 1 iki 11. Prie kiekvieno disko pritvirtinti balionai su skaičiais nuo 1 iki 11, rodančiais numerį to disko, ant kurio turi pereiti bebras pagal visiems bebrams duodamą bendrą komandą. Bebrų keliavimas, vykdant vieną komandą, vadinamas ciklu. Po pirmo ciklo eina antras, po antro – trečias ir t. t., kol visi bebrai atsistoja vėl ant tų pačių diskų, ant kurių buvo žaidimo pradžioje.



Kiek ciklų užtruks žaidimas?

- A) 20
- B) 40
- C) 11
- D) 121

## Paaiškinimas

Teisingas atsakymas: A.

Pasekime bebrą, stovintį ant 1 disko. Po pirmo ciklo jis atsidurs ant 5 disko, po antro – ant 9, po trečio – ant 3, po ketvirto – vėl ant 1, t. y., pradinio disko. Panašiai bebrui, stovinčiam ant 2 disko, ten pat grįžti prireiks 5 ciklų ( $2 \rightarrow 10 \rightarrow 11 \rightarrow 7 \rightarrow 4 \rightarrow 2$ ). Bebrui, stovinčiam ant 6 disko, grįžti ant pradinio disko prireiks tik dviejų ciklų ( $6 \rightarrow 8 \rightarrow 6$ ). Jeigu patikrintume visų likusių bebrų keliones, pastebėtume, kad bet kuris iš jų pasieks savo pradinį diską per kurį nors vieną rastą ciklų skaičių: 2, 4 arba 5. teisingas atsakymas yra lygus šių trijų skaičių mažiausiam bendrajam kartotiniui – 20.

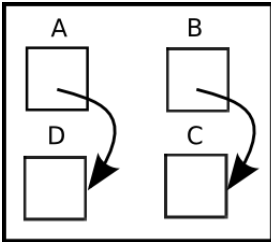
## Tai informatika!

Uždavinyje panaudoti du elementai, turintys sąsajų su programavimu: 1) sąlyginis ciklas – „ciklas vyksta, kol“ ir 2) rodyklės duomenų tipas – ant balionų užrašyti skaičiai, rodantys, kur bebrui keliauti.

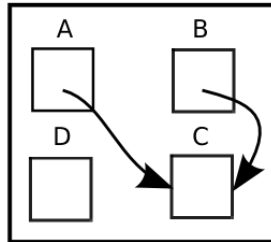
Raktiniai žodžiai: ciklas, rodyklė, mažiausias bendrasis kartotinis.

## 25. Krypties keitimas

Komandą  $A \leftarrow B$  keičia rodyklių kryptis šitaip:



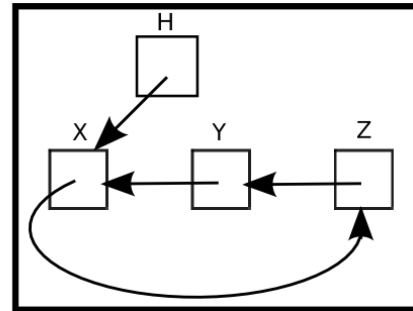
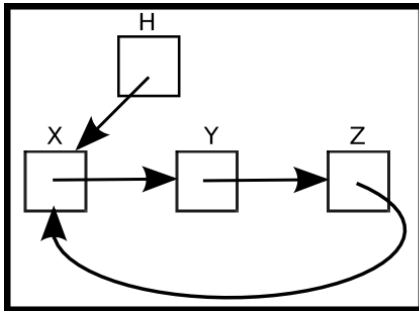
Vaizdas prieš



Vaizdas po

Dabar rodyklė iš A rodo ten, kur rodo rodyklė iš B. Reikšmei A priskiriama reikšmė B per papildomą lauką C.

Kuria komandų seka (atliekant komandas iš eilės taip, kaip jos pateiktos) iš kairiojo paveikslėlio galima gauti dešinįjį paveikslėlį?



- A)  $X \leftarrow Y$   
 $Y \leftarrow Z$   
 $Z \leftarrow X$
- B)  $X \leftarrow Z$   
 $Z \leftarrow X$   
 $Y \leftarrow H$
- C)  $Z \leftarrow Y$   
 $X \leftarrow Z$   
 $Y \leftarrow H$
- D)  $Z \leftarrow X$   
 $X \leftarrow Y$   
 $Y \leftarrow H$

## Paaiškinimas

Teisingas atsakymas: D.

A neteisingas, nes X reikšmė pakeičiama prieš atliekant komandą  $Z \leftarrow X$ .

B neteisingas, nes bandoma sukeisti X ir Z reikšmes be papildomo lauko.

C neteisingas, nes komanda  $A \leftarrow B$  interpretuojama neteisingai: „rodyklė iš A rodyti į B“.

D teisingas, nes, priešingai nei A variante, teisingai naudojamas papildomas laukas H.

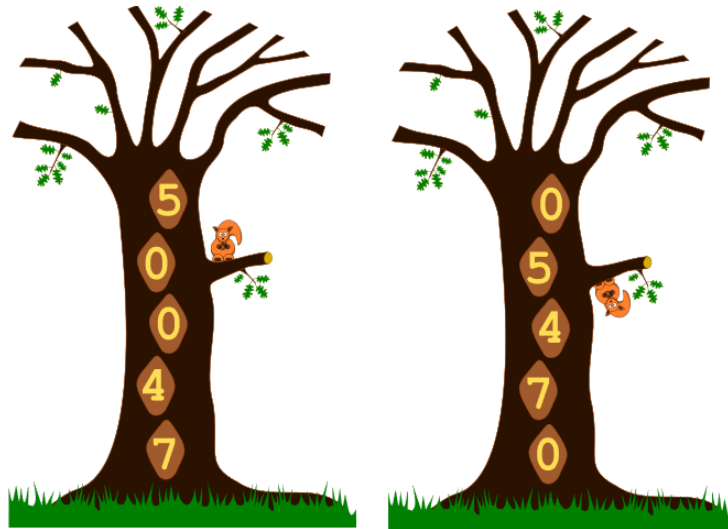
## Tai informatika!

Programuojant reikia subendrinti skirtingose vietose saugomų duomenų atmintis. Kompiuterio atmintis dažnai vaizduojama kaip popieriaus lapas, suskirstytas į langelius duomenims laikyti. Ryšiai tarp duomenų nurodomi rodyklėmis. Kompiuteryje saugomo atminties turinio valdymas yra svarbi programavimo dalis.

## 26. Savanaudės voverės

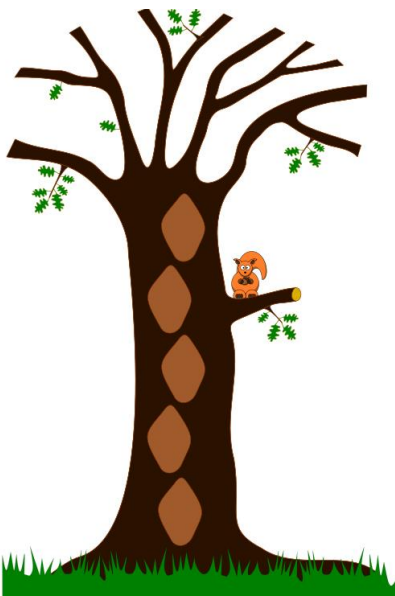
Medyje viena virš kitos yra penkios drevės. Jose gyvena 16 voverių. Kasdien kiekviena voverė apžiūri, kiek turi savo drevės kaimynų, kiek kaimynų aukščiau esančioje drevėje (jei tokia yra) ir kiek žemiau esančioje drevėje (jei tokia yra). Tada nustato, kurioje iš šių drevių yra mažiausiai kaimynų ir naktį persikelia į pasirinktą drevę (arba lieka savoje). Jei tokių drevių ne viena, voverė pirmenybę teikia savo drevi, tada aukščiau esančiai drevi ir galiausiai – žemiau esančiai drevi.

Pavyzdžiui, voverės drevėse gyvena šitaip: 5, 0, 0, 4, 7 (pradedant nuo viršaus ir einant žemyn). Rytoj visos 5 viršutinio aukšto voverės apsigyvens viena dreve žemiau (0 kaimynų yra geriau nei 4), 7 voverės iš apatinės drevės persikels viena dreve aukščiau (4 kaimynai yra geriau negu 6) ir 4 voverės iš antrosios drevės nuo apačios persikels dreve aukščiau (0 kaimynų yra geriau nei 3).



Daugeliu atvejų voverės po kurio laiko galiausiai atsидurs toje pačioje drevėje.

Surask situaciją, kai voverės negalės susirinkti vienoje drevėje. Surašyk voverių skaičius drevėse.





## Paaiškinimas

Galimi keli atsakymai, pavyzdžiui: 0 7 9 0 0, 5 0 0 11 0.

### Tai informatika!

Tai skruzdžių kolonijos elgseną arba kitaip – vadinamąjį skruzdžių intelektą – imituojantis uždavinys. Šis algoritmas remiasi tuo, kad problemos gali būti sprendžiamos net naudojant paprastus įrenginius, jei tų įrenginių yra labai daug. Pavyzdžiui, skruzdės juda pagal elementarias taisykles ir nepriklausomo viena nuo kitos. Tačiau kai skruzdžių daug, jos gali padaryti įspūdingus dalykus. Todėl skruzdžių kolonijos modelis taikomas daugeliui sudėtingų uždavinių spręsti: rasti optimalų kelią grafe, spręsti keliaujančio pirklio uždavinį, nuskabyti medžių lapus ir pan.

Šiame uždavinyje voverių drevių pasirinkimas grindžiamas paprastomis taisyklėmis, tačiau voverių elgesys toli gražu nerodo jų sumanumo. Jos norėtų gyventi kuo erdviau, tačiau susigrūda į vieną drevę. Taigi šis uždavinys gana pamokantis: skruzdžių kolonijos algoritmą reikia taikyti apdairiai, atsiminti, kad kartais geriau veikti bendradarbiaujant, o ne vien tik siekti naudos sau.

Raktiniai žodžiai: skruzdžių kolonijos elgsenos modelis, kolektyvinis intelektas, skruzdžių algoritmas.

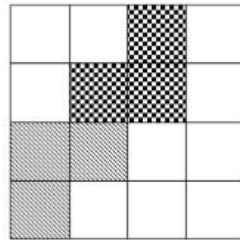
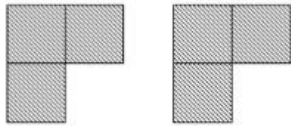
## 27. Žaidimas L figūromis

Bebrai Kikis ir Vivis žaidžia žaidimą: dėlioja L formos figūras 4×4 langelių lentoje. Figūros dedamos paėiliui pagal šias taisykles:

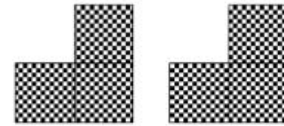
- Kikis deda aukštyn apverstas L figūras (žr. paveikslo kairėje),
- Vivis deda kairėn pasuktas L figūras (žr. paveikslo dešinėje),
- dedama figūra turi visa tilpti ant lentos,
- figūros negali dengti viena kitos.

Padėtų figūrų negalima nuimti. Žaidėjas pralaimi žaidimą, jei yra jo ėjimas, bet jis negali padėti figūros pagal taisykles. Pavyzdyje Kikis pradeda žaidimą ir gali laimėti, padėdamas figūrą apatiniajame dešiniajame kampe.

Kikis deda figūrą



Vivis deda figūrą



Galimi 9 skirtingi Kikio pradiniai ėjimai. Kiek iš jų garantuoja pergalę?

- (A) 0
- (B) 1
- (C) 2
- (D) 3





Kuriame  
Lietuvos ateitį  
2014–2020 metų  
Europos Sąjungos  
fondų investicijų  
veiksmų programa



**Vilniaus  
universitetas**

Informatinio mąstymo uždavinių rinkiniai sukurti įgyvendinant projektą „Aukštųjų mokyklų tinklo optimizavimas ir studijų kokybės gerinimas Šiaulių universitetą prijungiant prie Vilniaus universiteto“ (Nr. 09.3.1-ESFA-V-738-03-0001), finansuojamą iš Europos socialinio fondo lėšų pagal 2014–2020 metų Europos Sąjungos fondų investicijų veiksmų programos 9 prioriteto „Visuomenės švietimas ir žmogiškųjų išteklių potencialo didinimas“ įgyvendinimo priemonę Nr. 09.3.1-ESFA-V-738 „Aukštųjų mokyklų tinklo tobulinimas“.

„Bebro“ konkurso uždavinių rinkinys tinka Mokyklos pedagogikos studijų programos moduliui „Informatikos didaktika“. Studentai, būsimi mokytojai, nagrinėdami „Bebro“ uždavinius susipažįsta su įvairiais informatikos konceptais, nagrinėja sudėtingesnes informatikos sąvokas, išmoksta jas paaiškinti.