

S  
i  
d  
r  
i  
s  
k  
l  
a  
s  
p  
a  
u  
s  
t  
a  
i  
s



# Informatikos ir informatinio mąstymo uždavinių rinkinys

## Nr. 14



Šiame rinkinyje pateikiama XX informatikos ir informatinio mąstymo konkurso (iššūkiu) „Bebras“ II etapo, vykusio 2024 metų žiemą, užduotys, jų atsakymai ir paaiškinimai. Visos užduotys (įskaitant grafiką ir kitą medžiagą) licencijuojamos pagal Kūrybinių bendrijų licenciją – „Creative Commons Attribution-ShareAlike 4.0 International License“.

Šis užduočių rinkinys skiriamas 9–12 klasių mokinių informatinio mąstymo gebėjimams ugdyti.

Dėkojame Daumilui Ardickui, dr. Tatjanai Jevsikovai, Alvidai Lozdienei, talkinusiems verčiant ir adaptuojant uždavinius. Taip pat dėkojame tarptautinei „Bebro“ bendruomenei ir užduočių autoriams.

Sudarė Lina Vinikienė

Konsultavo Valentina Dagienė

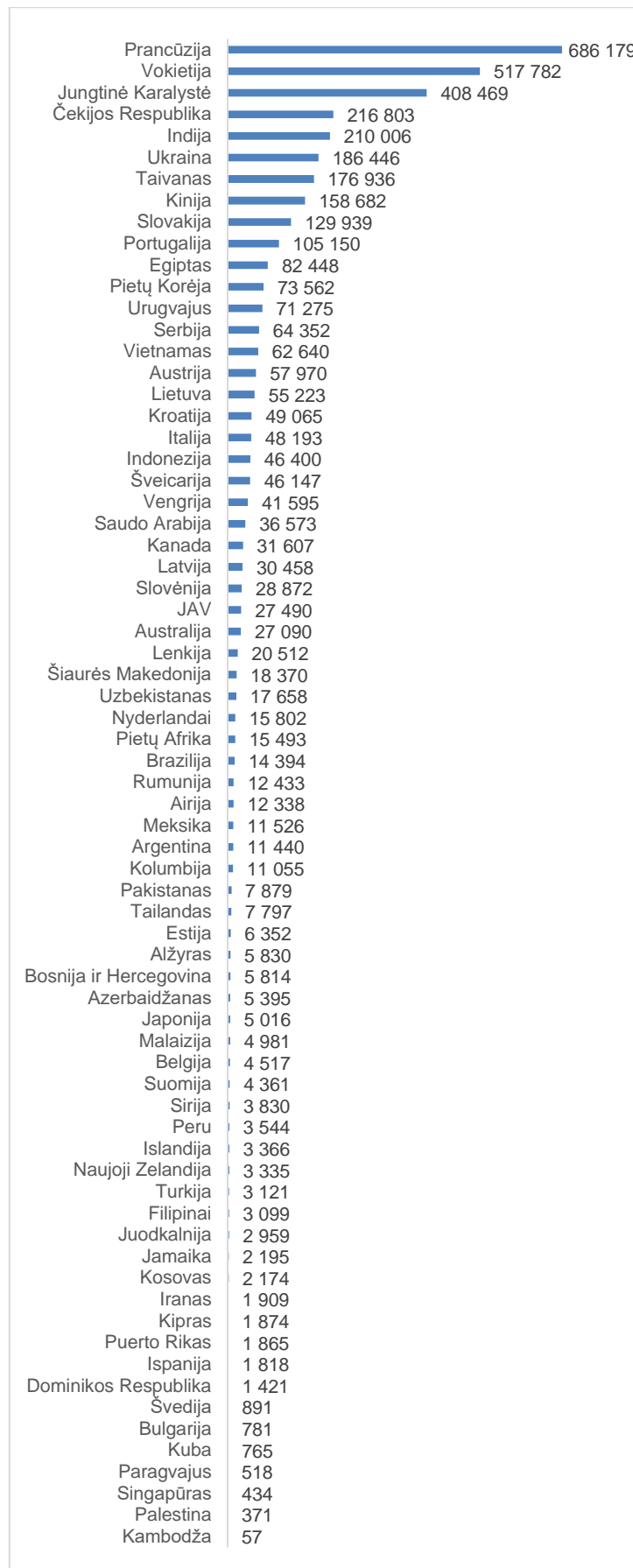
Redagavo Viktoras Dagys

Viršelį kūrė Vaidotas Kinčius



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

2023 metų „Bebro“ konkurso I etapo uždutis sprendė 3 936 642 mokiniai 71-oje valstybėje. Lietuvoje šio konkurso uždavinius sprendė 55 223 mokiniai.



2023 m. „Bebro“ dalyvių skaičius pagal šalis (71 šalies duomenys)

## Įsteigta tarptautinė asociacija „Bebras International“

2023 m. gruodžio 8 d. Vilniaus universitete (VU) buvo pasirašyta asociacijos „Bebras International“ sutartis. Asociacijos įsteigimą patvirtino gyvai susitikę ir dokumentą pasirašę įstaigų atstovai – VU rektorius prof. Rimvydas Petrauskas, Baskų krašto universiteto (Universidad del País Vasco, Ispanija) profesorius Javieras Jesus Bilbao Landache, Informatikos olimpiados fondo Nyderlanduose (Stichting Informatica Olympiade) direktorius Eljakimas Schrijversas, organizacijos „Informatika ir IT švietime“ (Computer Science and IT in Education) Belgijoje įkūrėjas prof. Sébastienas Combéfisas ir Pietų Bohemijos universiteto (Jihočeská univerzita v Českých Budějovicích, Čekija) profesorius Jiří Vaníčekas.

Pasirašant sutartį dalyvavo su „Bebras International“ veiklomis dirbanti VU profesorė Valentina Dagienė, VU Filosofijos fakulteto prodekanas dr. Vytis Silius ir Kompiuterių mokslo draugijos Vokietijoje (Bundesweite Informatikwettbewerb, BWINF) vykdančiasis direktorius dr. Wolfgangas Pohlis.



„Bebras International“ steigėjų posėdyje, kuris vyko pasirašius sutartį, prof. V. Dagienė buvo išrinkta asociacijos prezidente.

Asociacija „Bebras International“ yra tarptautinė informatikos mokslo srities organizacijų sandrauga, jos tikslas – populiarinti informatiką ir informatinį mąstymą jaunimui, mokytojams ir visuomenei. Siekdami šio tikslo, asociacijos nariai savo šalyse organizuoja lengvai prieinamus ir motyvuojančius iššūkius.

Šaltinis: <https://naujienos.vu.lt/pasirasyta-asociacijos-bebras-international-steigimo-sutartis/>

Daugiau informacijos apie „Bebro“ konkursą pateikiama interneto svetainėse:

- Tarptautinė „Bebro“ iššūkio svetainė: [www.bebbras.org](http://www.bebbras.org)
- Lietuvos „Bebro“ svetainė: [bebras.lt](http://bebras.lt)
- Konkurso sistema: [lt.bebbras.lt](http://lt.bebbras.lt)

2024 m. vasario 3 d. Lietuvoje vykusiamame „Bebro“ konkurso II etape dalyvavo 351 jaunis (9–10 klasių mokiniai) ir 265 kolegos (11–12 klasių mokiniai). Jie sprendė po 15 uždavinių.



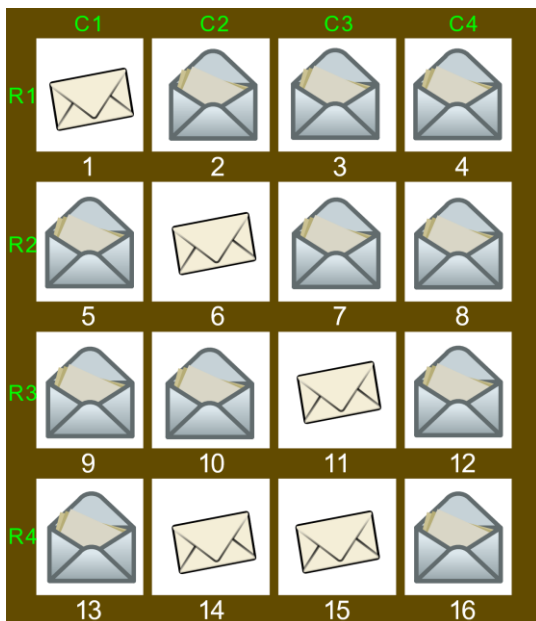
Lentelėje pateikiamas XX konkurso II etapo uždavinių skirstymas pagal amžiaus grupes. Skaičius langelyje prie uždavinio nurodo sudėtingumo lygį:

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

Nr.	Uždavinio pavadinimas	Uždavinio identifikatorius	Jauniai	Kolegos
1	Slapti laišakai	2023-PH-04	6	6
2	Kilimo audimas	2022-TR-02	6	
3	Akvarelė	2023-VN-04	6	
4	Dramblys šaldytuve	2021-IE-06	6	
5	Vizitai	2019-RO-01	6	
6	Keturios kortelės	2022-BE-02	9	6
7	Darbų paskirstymas	2023-IR-02	9	
8	Lobio dėžutė	2022-MK-01	9	
9	Bebrų duomenys	2022-IT-03a	9	
10	Garvežiai	2023-RO-01	9	
11	Tiltų statyba	2023-NZ-01	12	9
12	Į kalnus	2023-TW-04	12	9
13	Magiška šalis	2022-CA-02	12	9
14	Gražiausias brangakmenis	2022-CA-04	12	9
15	Automobilių numeriai	2023-LT-07	12	12
16	Naujas žaislas	2023-ME-03b		6
17	Ponia Begalybė	2022-IR-03a		6
18	Salų jungimas	2022-LT-01		6
19	Kriauklės ir akmenukai	2022-IT-02		9
20	Rikiavimas	2022-UA-03b		12
21	Bebrų dirbtinis intelektas	2022-DE-04		12
22	Akmenys	2022-CH-09c		12
23	Šnipai	2015-SI-06		12

## 1. Slapti laiškai

Bebrijos Respublikoje saugoma slaptų laiškų spintelė. Iš spintelėje esančių 16 laiškų vokų, sunumeruotų nuo 1 iki 16, 10 yra atplėšti, o 6 laiškai vis dar yra užklijuotuose vokuose. Vieną vakarą prieš šnipas įsiskverbė ir atplėšė vieną iš neatvertų vokų. Tačiau pamiršo jį vėl užklijuoti. Kitą rytą, pastebėjus, kad dabar yra 11 atplėštų vokų (žr. paveikslą), Bebrijos Respublikoje pradėtas tyrimas.



Respublikos apsauginis neprisimena visų detalių, tačiau yra tikras, kad prieš šnipui patenkant į vidų, buvo taip:

- Atplėštų vokų skaičius C2 ir C4 stulpeliuose kartu sudėjus buvo lyginis;
- Atplėštų vokų skaičius C3 ir C4 stulpeliuose kartu sudėjus buvo lyginis;
- Atplėštų vokų skaičius R2 ir R4 eilutėse kartu sudėjus buvo lyginis;
- Atplėštų vokų skaičius R3 ir R4 eilutėse kartu sudėjus buvo lyginis.

Kurį voką atplėšė šnipas?

1. 5
2. 9
3. 10
4. 13
5. Informacijos neužtenka, kad būtų identifikuotas atplėštas vokas.



## Paaiškinimas

Teisingas atsakymas – 13.

Galima samprotauti taip:

- Pagal apsauginio prisiminimus, C2 ir C4 stulpeliuose kartu sudėjus yra lyginis skaičius atplėštų vokų. Kadangi šnipas atplėšė tik vieną voką, tai reiškia, kad šnipas atplėštas laiško vokas turi būti C1 arba C3 stulpelyje.
- C3 ir C4 stulpeliuose kartu sudėjus yra lyginis skaičius atplėštų vokų. Turint omenyje ankstesnį teiginį, daroma išvada, kad šnipas atplėštas vokas turi būti C1 stulpelyje.
- R2 ir R4 eilutėse kartu paėmus yra nelyginis skaičius atplėštų vokų, o tai nedera su apsauginio liudijimais. Tai reiškia, kad šnipas atplėšė laišką, esantį arba R2, arba R4 eilutėje.
- R3 ir R4 eilutėse kartu sudėjus yra nelyginis skaičius atplėštų vokų, o tai neatitinka apsauginio pasakojimo. Turėdami omenyje ankstesnį teiginį, darome išvadą, kad šnipas atplėštas vokas turi būti R4 eilutėje.

Taigi laiškas, kurio voką atplėšė šnipas, yra C1 stulpelyje ir R4 eilutėje, o tai reiškia, kad teisingas atsakymas yra 13.

## Tai informatika!

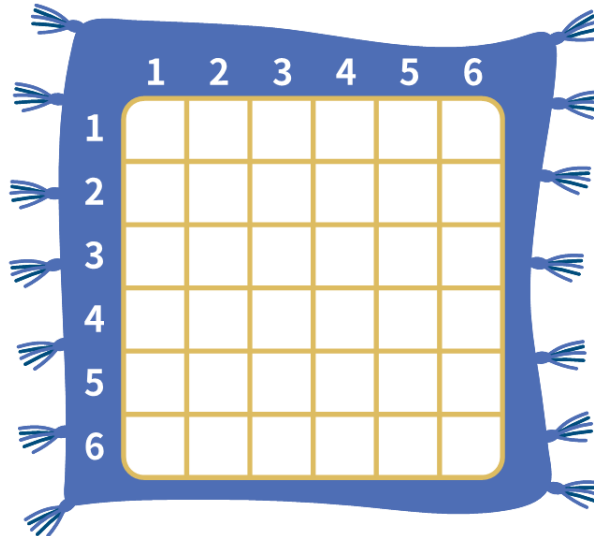
Ši užduotis supažindina su klaidų taisymo kodo sąvoka. Duomenys skaitmeniniu pavidalu iš esmės yra bitų seka (1 ir 0). Perduodant kompiuterių tinklais, jie gali būti sugadinti dėl triukšmo arba dėl nesankcionuotų veiksmų. Duomenys, saugomi DVD diskuose, taip pat gali būti sugadinti, jei diskas subraižytas. Todėl svarbu turėti schemą, kuri padėtų ne tik aptikti, kad įvyko pažeidimas (klaidų aptikimas), bet ir ištaisyti pažeistus bitus (klaidų taisymas).

Pirmąjį šiuolaikinį klaidų taisymo kodą 1950 m. pristatė Richardas Hammingas. Šiame uždavinyje konkrečiai pasiskolinta [15, 11] Hammingo kodo idėja, kai užklijuoti ir atplėšti laiškų vokai atitinka 0 ir 1. Šioje kodavimo schemoje imami 11 bitų duomenys ir įterpiami 4 lyginumo bitai. Šie lyginumo bitai užima 2, 3, 5 ir 9 lizdus, o likusius lizdus (išskyrus pirmą) užima pradinių duomenų bitai. Lyginumo bitai parenkami taip, kad 2 ir 4 stulpeliuose, 3 ir 4 stulpeliuose, 2 ir 4 eilutėse bei 3 ir 4 eilutėse būtų lyginis skaičius vienetų. Galiausiai 1 lizde esantis bitas parenkamas toks, kad iš viso būtų lyginis skaičius vienetų.

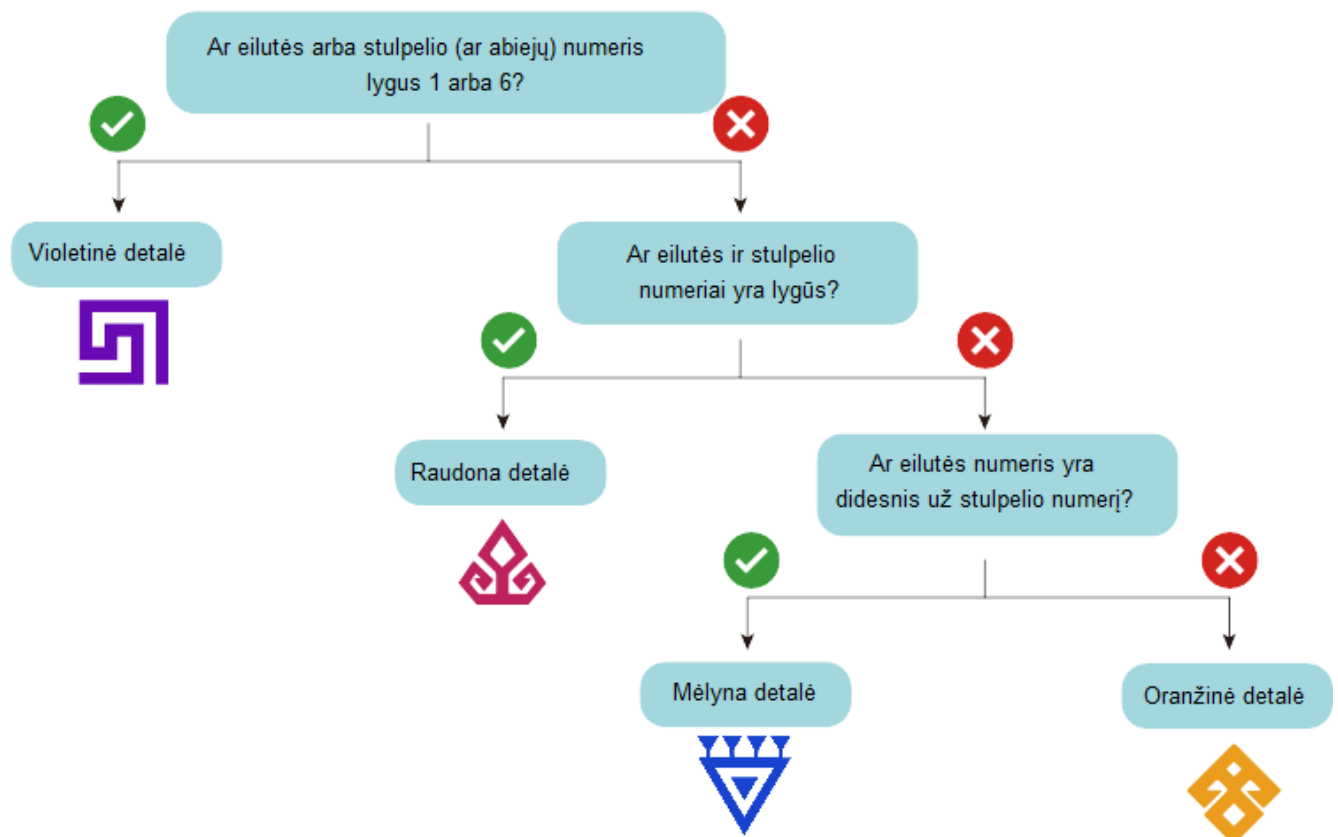
Kaip matyti iš pateikto sprendimo, pagal šią schemą galima identifikuoti ir vėliau ištaisyti pažeistą bitą, jei yra ne daugiau kaip 1 pažeistas bitas. Nepaisant to, kad norint ištaisyti 2 ar daugiau sugadintų bitų, reikia sudėtingesnių klaidų taisymo kodų, Hammingo kodo efektyvumas lemia, kad jis ir toliau naudojamas NAND atmintinės lustuose, įmontuotuose į mobiliuosius telefonus ir puslaidininkinius diskus.

## 2. Kilimo audimas

Audžiamas turkiškas kvadratinis kilimas, sudarytas iš 6 eilučių ir 6 stulpelių langelių:



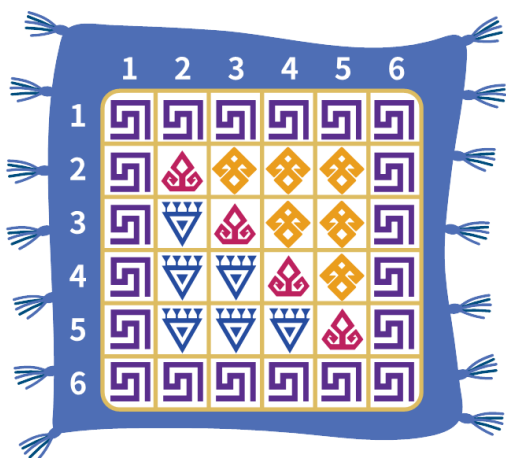
Kiekviename langelyje audžiama tokia ornamento detalė, kuri gaunama atsakius į šiuos klausimus:



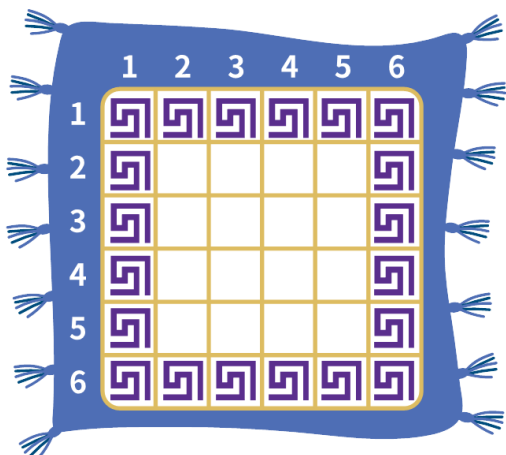
Kaip atrodys išaustas kilimas?

## Paaiškinimas

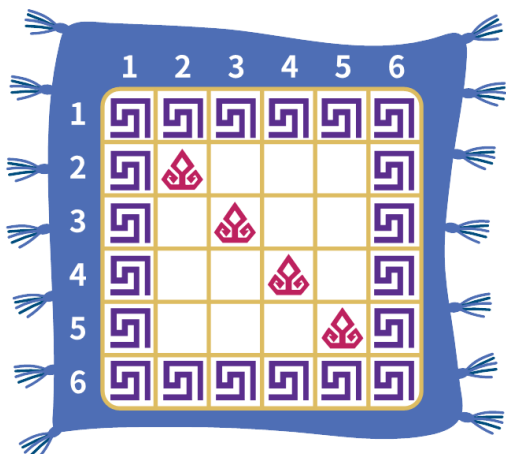
Teisingas atsakymas:



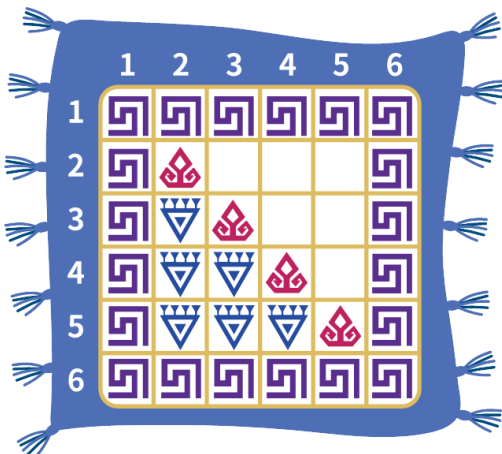
Pirmuoju klausimu nustatoma, kad 1-osios ir 6-osios eilučių bei 1-ojo ir 6-ojo stulpelių langeliuose audžiamos violetinės detalės. Taip gaunamas toks simbolių išdėstymas:



Antrasis klausimas rodo, kad pagrindinės įstrižainės langeliuose audžiamos raudonos detalės, nes įstrižainėje eilučių ir stulpelių numeriai yra vienodi:



Tolesnis klausimas rodo, kad visuose langeliuose į kairę nuo įstrižainės eilučių numeriai yra didesni už stulpelių numerius. Taigi šiuose langeliuose audžiama mėlyna detalė:



Galiausiai matome, kad visuose į dešinę nuo įstrižainės langeliuose eilučių numeriai nėra didesni už stulpelių numerius, todėl jie užpildomi oranžinėmis detalėmis ir taip gaunamas B variantas.

Tai informatika!

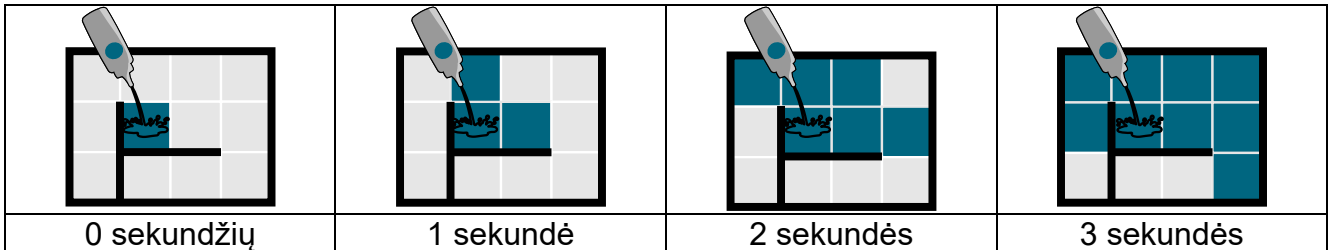
Matematikoje ir informatikoje algoritmas – tai baigtinė aiškiai apibrėžtų komandų (instrukcijų) seka, naudojama konkrečioms uždaviniams spręsti ar skaičiavimams atlikti. Algoritmai naudojami kaip skaičiavimų ir duomenų apdorojimo specifikacijos.

Sprendimų medis laikomas vienu iš būdų algoritmo struktūrai pavaizduoti. Algoritmas paprastai sudaromas iš sąlyginių sakinių, pavyzdžiui, *if-e/se*, kartojimo ir kitų komandų. Tai reiškia, kad kiekvieną sprendimų medžio šaką sudaro klausimas arba teiginys: jei teiginys teisingas, algoritmas toliau veikia pagal vieną medžio šaką, priešingu atveju algoritmas veikia pagal kitą medžio šaką. Taip einama toliau tikrinant šakų teiginius ir atliekant komandas, kol pasiekiamas algoritmo rezultatas.

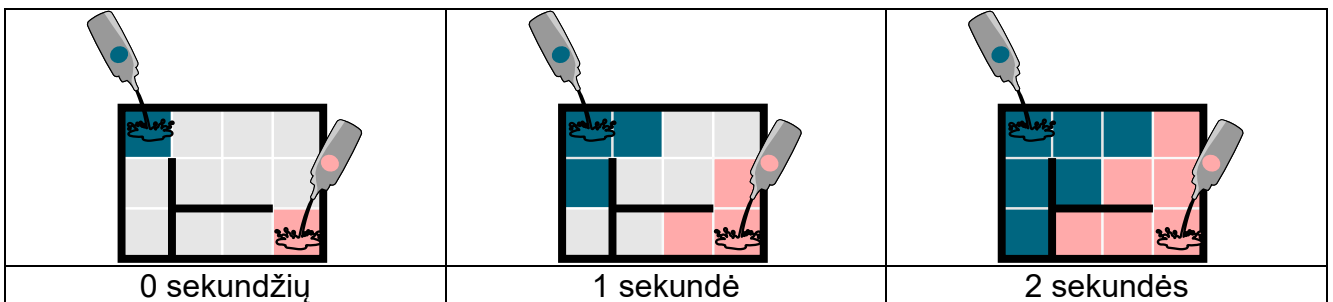
Pasinaudodami dirbtiniu intelektu, algoritmai gali atlikti automatinius išvedimus (vadinamus automatiniu samprotavimu) ir naudoti matematinius ir loginius tikrinimus, kad nukreiptų programos atlikimą įvairiais keliais (tai vadinama automatiniu sprendimų priėmimu).

### 3. Akvarelė

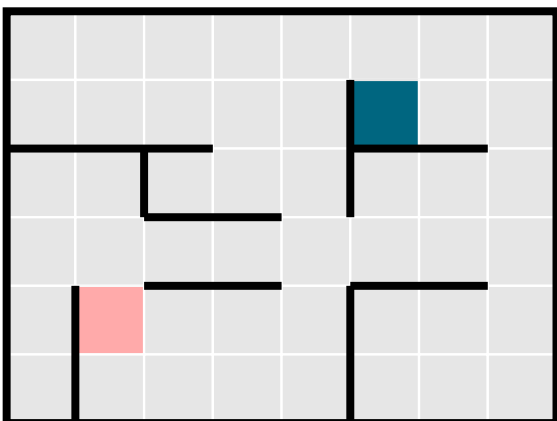
Bebrai į labirintą pila akvarelės dažus. Kas sekundę dažai pasklinda į gretimus tuščius langelius. Dažai negali plisti per sienas.



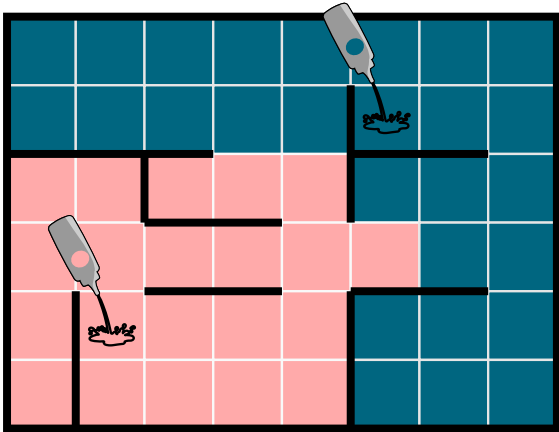
Jei bebrai į labirintą pila kelių spalvų dažus, pirmoji spalva visiškai užpildo pasiektą tuščią langelį. Jei kelios spalvos langelį pasiekia vienu metu, jis užpildomas tamsesne spalva.



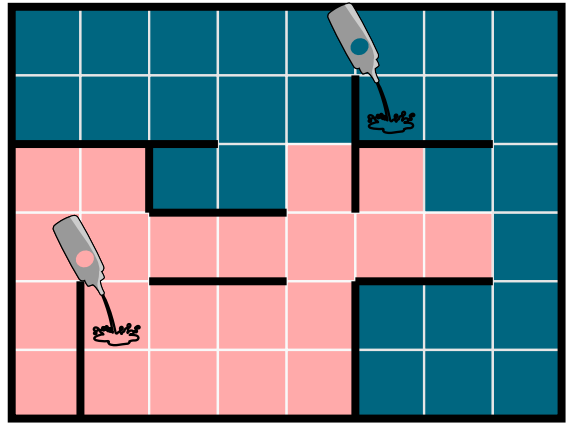
Bebrai į šį labirintą įpylė dviejų spalvų akvarelės dažų. Kaip atrodys labirintas, kai visi langeliai bus nuspalvinti?



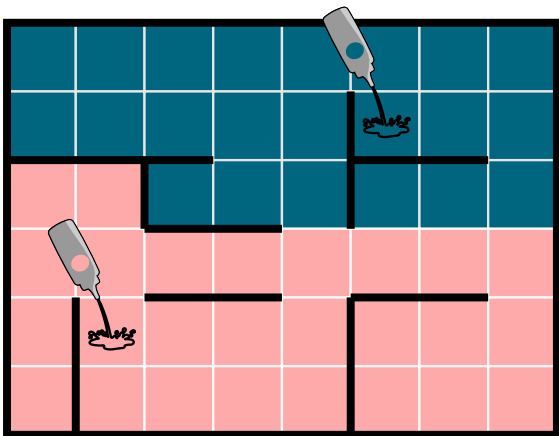
A



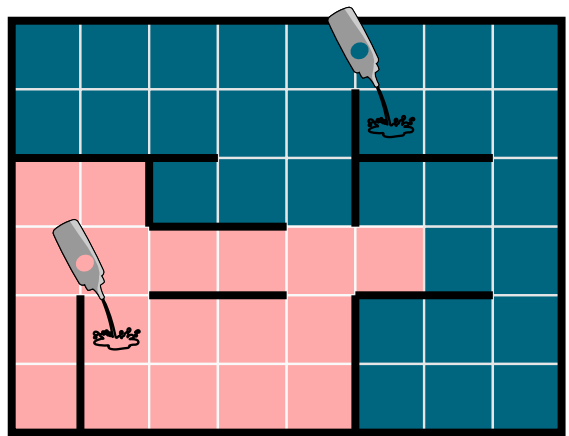
B



C

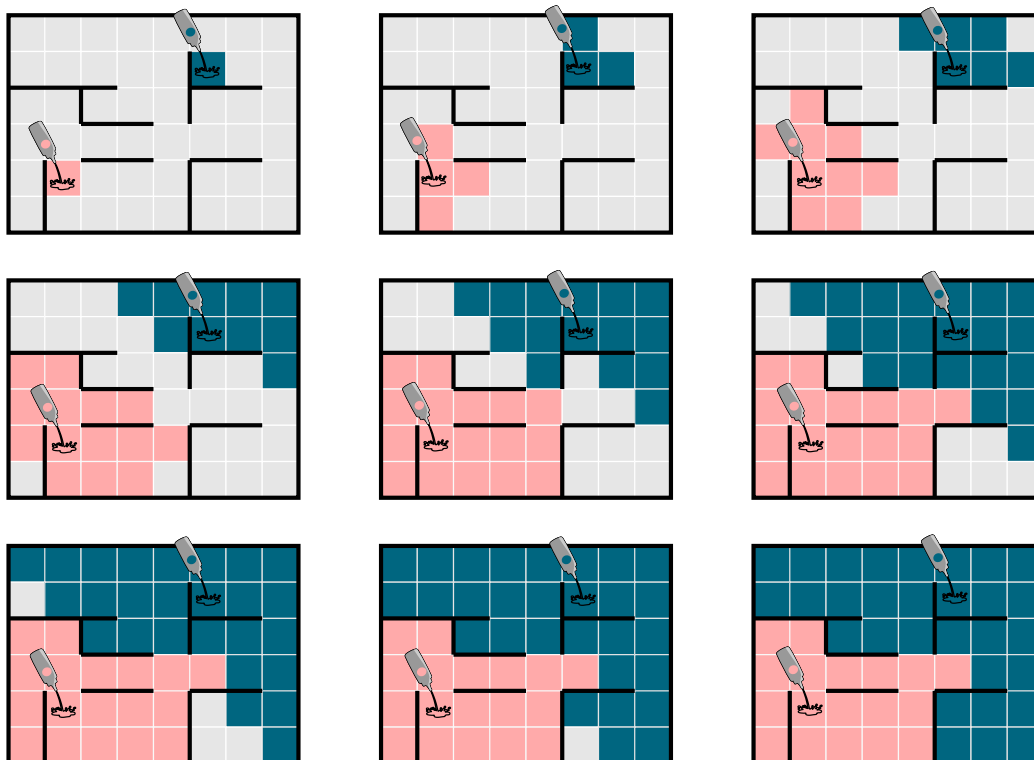


D



## Paaiškinimas

Teisingas atsakymas: D.



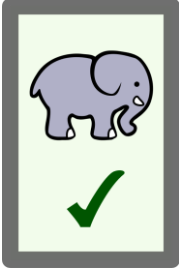

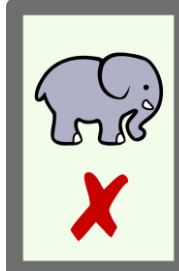
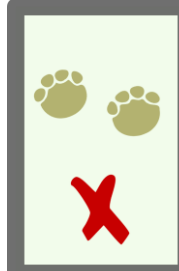
Tai informatika!

Užduotis primena dvimatį masyvą, vaizduojamą lentelę, kuri sudaryta iš eilučių ir stulpelių. Toks duomenų pateikimo būdas yra gana funkcionalus sekundę po sekundės imituojant labirinto būseną ir gaunant rezultatą. Taip pat tai galima pavaizduoti grafu, kuriame kiekvienas langelis yra sujungtas su savo kaimynais. Toks grafas leidžia greitai nustatyti, kuri spalva pasieks kvadratėlį, nmodeliuojant viso scenarijaus sekundę po sekundės. Šis metodas naudojant grafus yra toks pat, kaip ir paieškos į plotį vykdymas.

Paieškos į plotį algoritmas (angl. *breadth-first search*, BFS) yra vienas iš būdų ieškoti medyje ar grafe viršūnės, atitinkančios tam tikrus kriterijus. Vykiant paieškos į plotį algoritmą grafo viršūnės lankomos pagal griežtą taisyklę: pradėjus nuo pasirinktos viršūnės, apšaukiamos visos viršūnės, kurios pasiekiamos iš pasirinktos viršūnės vienu briauna (vienu ėjimu), tuomet – pasiekiamos dvejomis briaunomis (dviem ėjimais) ir t. t. Naudojant paieškos į plotį medžio apėjimo būdą išanalizuojami visos to paties lygio viršūnės, o tada jau pereinama prie kito lygio viršūnių. Paieška į plotį gali būti naudojama daugeliui grafų teorijos uždavinių spręsti.

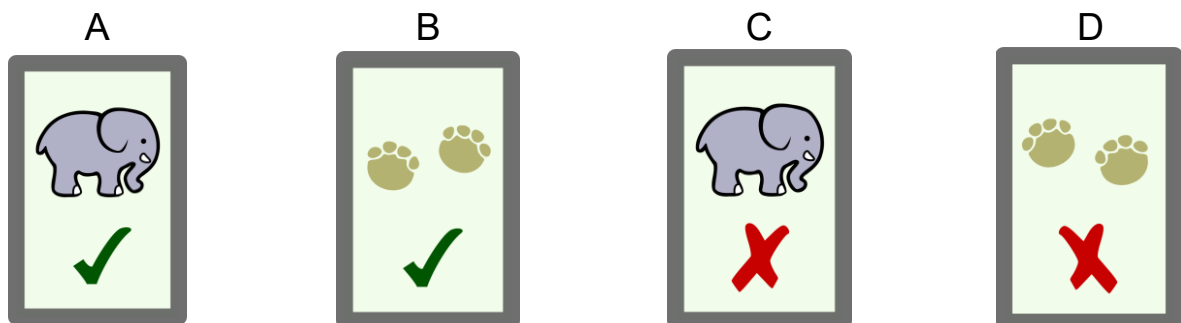
## 4. Dramblys šaldytuve

Jonei tėtis davė keturias žaidimo korteles, kurios vaizduoja tai, kas nutiko šaldytuve. Kiekviena kortelė turi dvi puses: vienoje pusėje parodyta, ar dramblys lankėsi šaldytuve, kitoje – ar svieste matomi pėdsakai. Štai šios keturios kortelės:

			
Dramblys apsilankė šaldytuve	Svieste matomi pėdsakai	Dramblys nebuvo šaldytuve	Svieste pėdsakų nėra

Jonės tėtis sako: „Jei į šaldytuvą užsuko dramblys, tai svieste yra pėdsakų“. Jonė abejoja, ar tai visada tiesa.


Spustelkite visas korteles, kuriomis Jonė galėtų įrodyti, kad jos tėtis klysta.




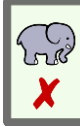



## Paaiškinimas

Teisingas atsakymas:  ir .

Jei Jonė apverčia kortelę  ir atverstoje pusėje parodyta, kad svieste pėdsakų nėra, ji sėkmingai įrodo, kad tėtis klydo. Todėl negalima tvirtinti, kad tėtis visada teisus. Žinoma, gali būti parodyta ir priešingai, tačiau tai nepatvirtina, kad tėtis visada teisus, nes tai tik viena iš dviejų galimybių (pėdsakai yra arba ne).

Jei Jonė apsiverčia šią kortelę , nesvarbu, ką ji mato atverstoje pusėje. Jei parodyta, kad dramblys apsilankė, tai dar kartą patvirtina jos tėčio teiginį. Jei parodyta, kad joks dramblys nesilankė, tai neprieštaruoja tėčio teiginiui, nes jis nieko nesakė apie tai, kas nutiks, jei dramblys nesilankys šaldytuve. Todėl su šia kortele negalima paneigti tėčio tvirtinimo.

Jei Jonė apverčia šią kortelę , visai nesvarbu, kas joje parodyta, nes tėtis nieko nesakė apie tai, kas nutiktų, jei šaldytuve nesilankytų joks dramblys.

Jei Jonė apverčia šią kortelę  ir atverstoje pusėje parodyta, kad šaldytuve apsilankė dramblys, ji sėkmingai įrodo, kad tėtis klydo. Žinoma, gali būti atverstoje kortelėje parodyta ir priešingai, tačiau tai nepatvirtina, kad tėtis visada teisus, nes tai tik viena iš dviejų galimybių (šaldytuve dramblys buvo arba nebuvo).

### Tai informatika!

Logika atlieka labai svarbų vaidmenį informatikos teorijoje ir taikyme – nuo paprastos Būlio logikos iki šiuolaikinėse dirbtinio intelekto sistemose naudojamų logikos reiškinių. Šio uždavinio teiginys vadinamas implikacija („Jei A, tai B“, dažnai rašoma „ $A \Rightarrow B$ “). Jis naudojamas ekspertinėse sistemose, pavyzdžiui, Prologe (rašoma „ $A \_ B$ “). Šios sistemos buvo vienos iš pirmosios kartos sėkmingų dirbtinio intelekto sistemų. Kompiuteriai gali tokius teiginius apdoroti ir sujungti juos su kitais pateiktais teiginiais, t. y., atlikti matematinį

informacijos apdorojimą ir taip gauti naujus ar netikėtus rezultatus. Implikacija gali būti vaizduojama kaip dvejetainė operacija, todėl ją galima pavaizduoti teisingumo lentele, panašiai kaip AND, NOT ir OR:

A	B	$A \Rightarrow B$
NETIESA	NETIESA	TIESA
NETIESA	TIESA	TIESA
TIESA	NETIESA	NETIESA
TIESA	TIESA	TIESA

Šią užduotį galima įvairiai perfrazuoti, jei tik laikomasi bendros formos „Jei A, tai B“.

Įdomu tai, kad jei implikacija pakeičiama į socialinę sutartį, tyrimai rodo, jog tokį uždavinį išspręsti lengviau. Tai Vasono atrankos uždavinio pavyzdys (Peter Cathcart Wason, 1966). Tokios užduotys – loginiai galvosūkių, paprastai susidedantys iš keturių kortelių, kuriais tikrinamas loginės implikacijos suvokimas. Paprastai užduotis pateikiama kaip bendros formos teiginys: Jei A yra tiesa, tai ir B yra tiesa.

Mokiniam pasakoma, kad kiekviena kortelė simbolizuoja stebėtą situaciją. Vienoje kortelės pusėje nurodoma, ar toje situacijoje A yra tiesa, ar netiesa, o kitoje – ar toje situacijoje B yra tiesa, ar netiesa. Mokiniai mato tik vieną pusę. Kortelės išdėstomos matomomis pusėmis taip:

1 kortelė – A yra tiesa

2 kortelė – A yra netiesa

3 kortelė – B yra tiesa

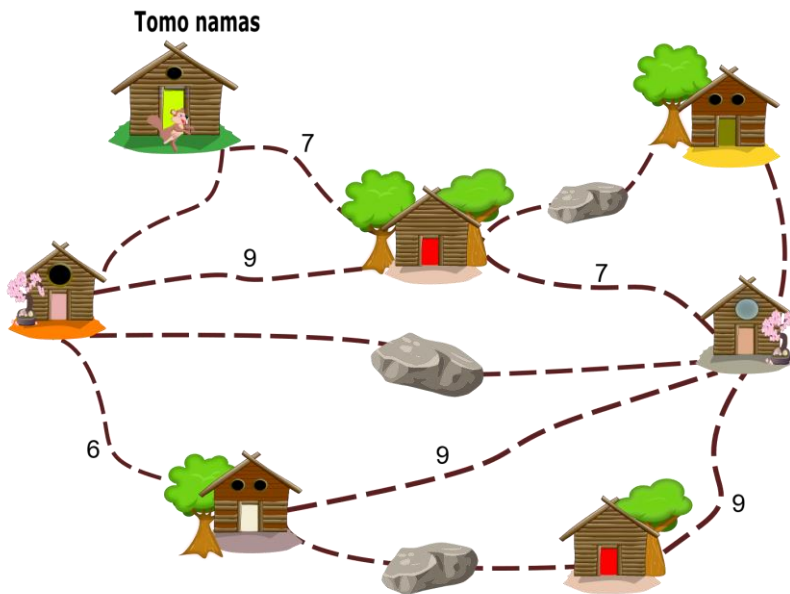
4 kortelė – B yra netiesa

1 ir 4 kortelėse visada yra atsakymas (kortelės, žinoma, sumaišomos), nepriklausomai nuo to, kas parašyta priešingoje pusėje. Tai vienintelės kortelės, galinčios iš tikrųjų paneigti loginį teiginį, kad jei A yra tiesa, tai ir B turi būti tiesa.

Vasono atrankos uždaviniams būdinga įdomi savybė – atrodo, kad žmonės juos daug lengviau supranta, kai pateikta situacija susijusi su socialinėmis sutartimis, pavyzdžiui, „jei pažeisi įstatymą, eisi į kalėjimą“ (Cosimides ir Tooby, 1992).

## 5. Vizitai

Mažasis Tomas nuobodžiauja namuose ir nusprendžia aplankyti visus savo giminaičius. Norėdamas naudotis kai kuriais keliais, jis turi sumokėti mokestį (nurodytą paveikslėlyje). Jei jis naudojasi keliu daugiau nei vieną kartą, mokėti nebereikia. Kai kurie keliai yra užversti akmenimis, todėl jais negalima naudotis.



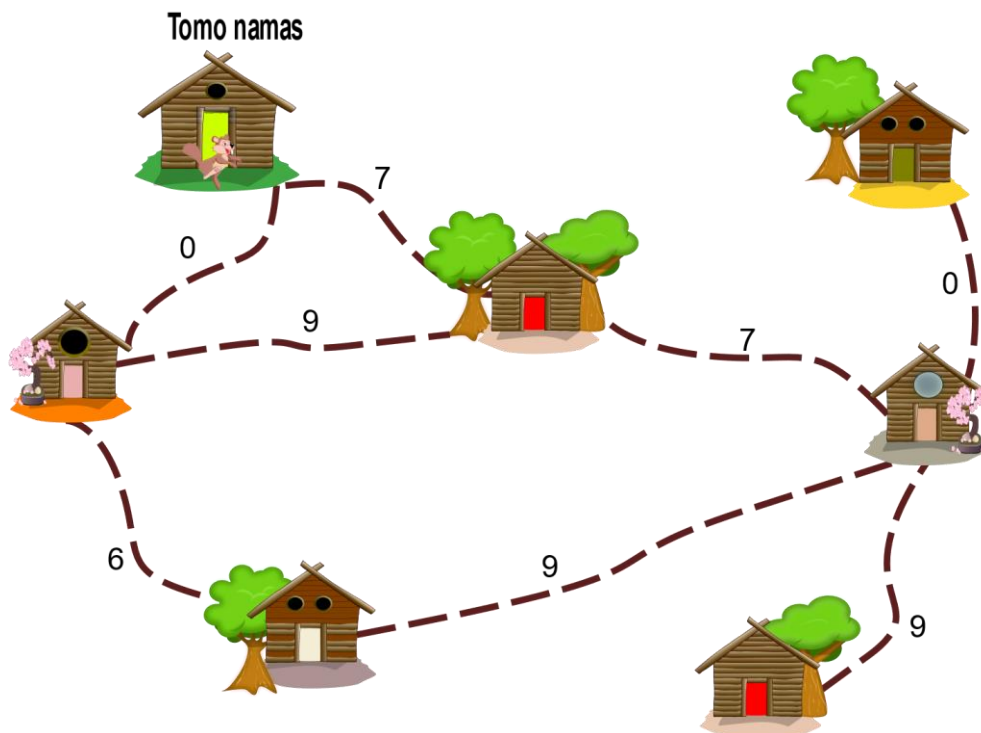
Kokią mažiausią pinigų sumą turi turėti Tomas, kad galėtų aplankyti visus savo giminaičius?

## Paaiškinimas

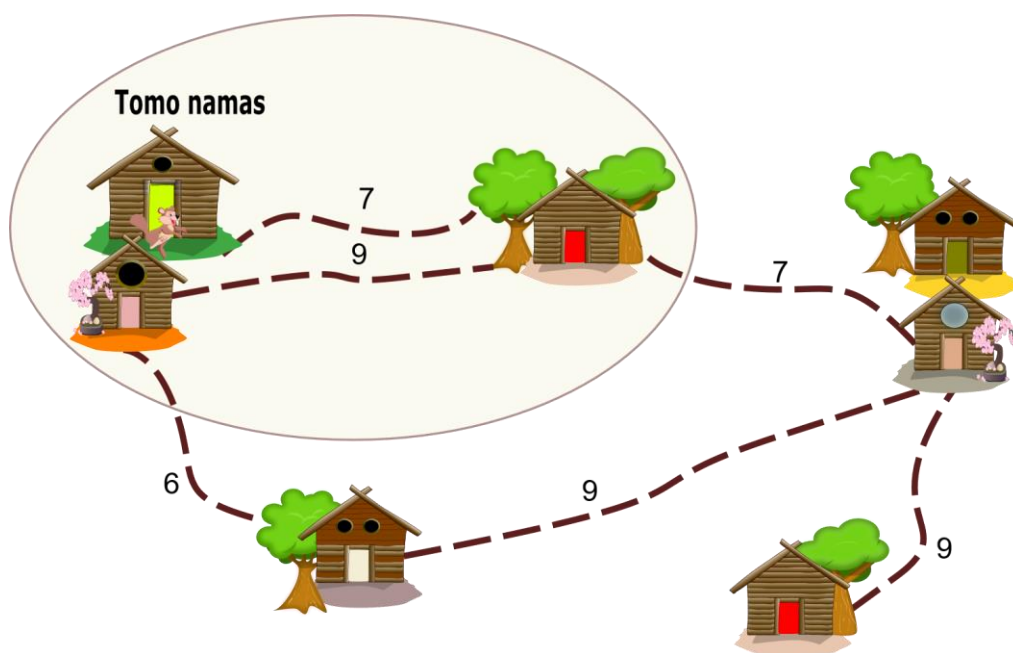
Teisingas atsakymas: 29.

Tarkime, kad kelių, už kuriuos nėra mokesčio, kaina yra lygi 0, o kelių su akmenimis nėra.

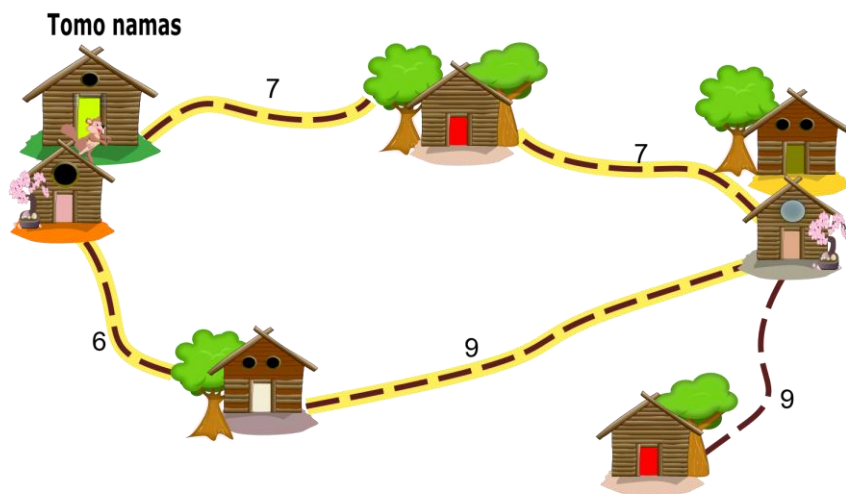
Taigi, žemėlapis atrodo taip:



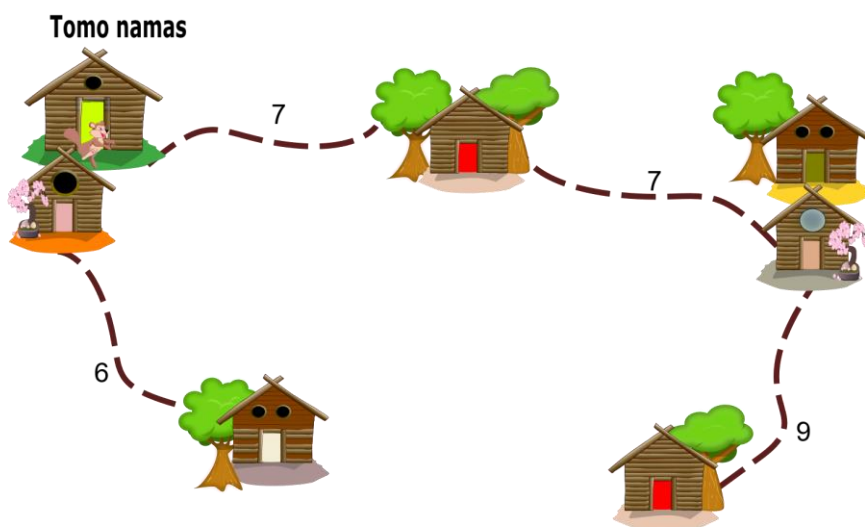
Vietovės, esančios tarp kelių, kurių mokestis yra 0, gali būti laikomos viena vietove, nes tarp jų galima laisvai judėti be mokesčio. Perkėlus jas vieną šalia kitos, žemėlapis atrodo taip:



Tarp apvestų vietovių yra du brangūs keliai, kurių vieno galima atsisakyti:



Atsisakoma vieno brangaus kelio ir visi giminaičiai bus aplankyti:



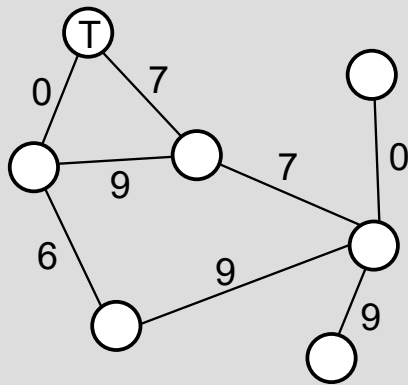
Susumuojame žemėlapyje likusių kelių mokesčius:  $6+7+7+7+9=29$ .

Tai informatika!

Dažnai kompiuteriai naudojami siekiant rasti geriausią maršrutą keliauti tarp vietovių (objektų). Geriausias gali reikšti „greičiausias“, „trumpiausias atstumas“ arba „pigiausias“, kaip ir šioje užduotyje, o vietovė gali reikšti „miestą“, „namą“ arba net „kompiuterį tinkle“. Tai trumpiausio kelio uždaviniai arba optimizavimo uždaviniai.

Šiai užduočiai spręsti atsisakoma nereikalingos paveikslėlyje pateiktos informacijos. Namai, jų forma, spalvos, kelių pavaizdavimas, net Tomo vardas, yra perteklinė informacija.

Kompiuterių programos atsirenka tik būtiną informaciją ir ją vaizduoja kaip schemą, kuri vadinama grafu. Pateiktoje schemeje yra visa reikalinga informacija.



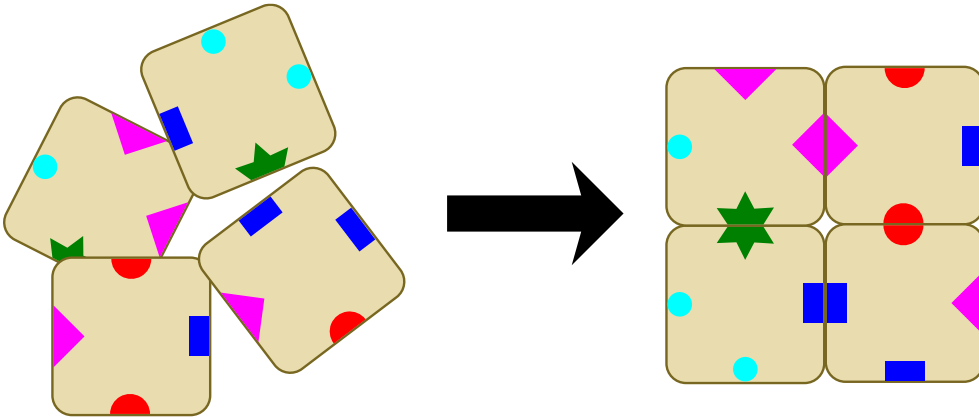
Informatikai šį uždavinį vadina minimalaus jungiamojo medžio (angl. *minimum spanning tree*, MST) uždaviniu. Minimalus jungiamasis medis – tai briaunų poaibis, jungiantis visas viršūnes mažiausiomis įmanomomis sąnaudomis. Klasikinis minimalaus jungiamojo medžio radimo būdas yra Kruskalo algoritmas.

## 6. Keturios kortelės

Keturias korteles reikia sudėlioti į  $2 \times 2$  kvadratą laikantis šios taisyklės:

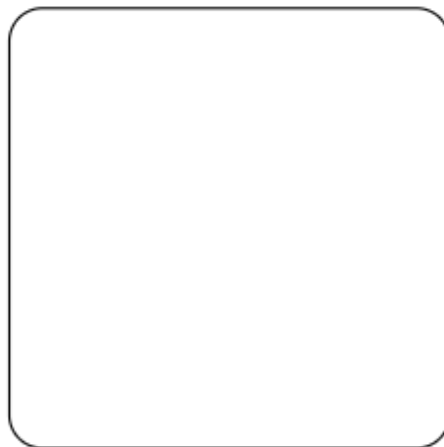
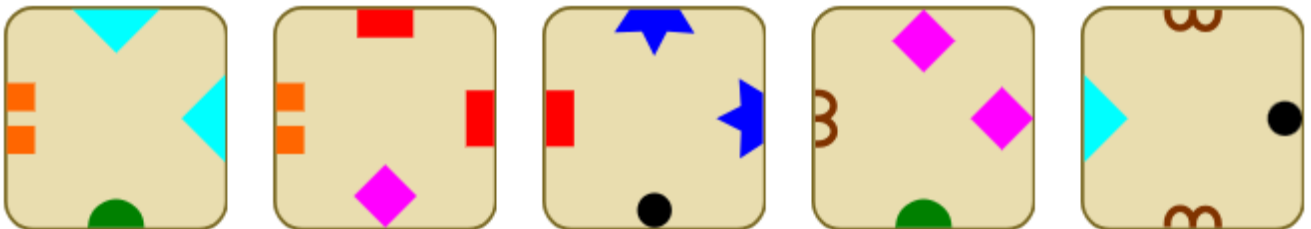
**Kortelės gali liestis tik tada, jei jų šonai sutampa.**

Pateikiame pavyzdį:



Duotos 5 kortelės. Keturias iš šių penkių kortelių turite sudėti į  $2 \times 2$  kvadratą pagal anksčiau nurodytą taisyklę. (Šiuo atveju yra tik viena galimybė.)

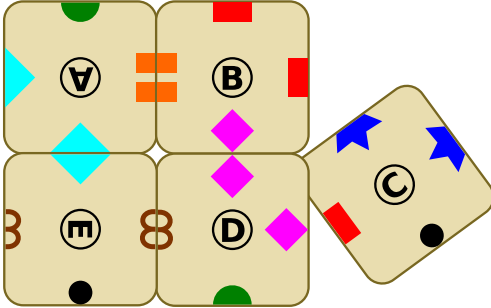
Spustelkite korteles, kad jas pasuktumėte, ir nuvilkite jas į kvadratą.



## Paaškinimas

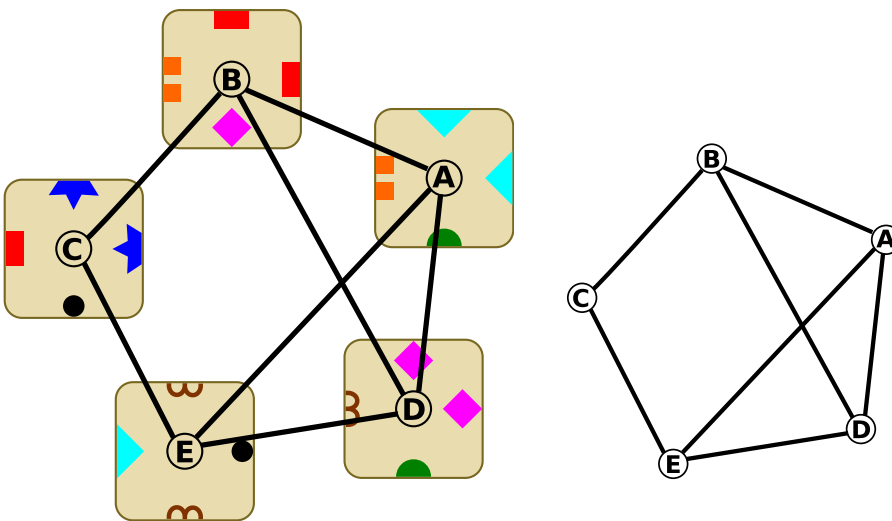
Teisingas atsakymas: C.

Iš kortelių A, B, D ir E, taikant minėtą taisyklę, galima sudaryti štai tokį 2×2 kvadratą:



Galima spręsti išbandant įvairias kombinacijas, kol viena iš jų tiks. Tačiau yra būdas sumažinti tikrintinų dėliojimų skaičių. Tai atliekame dviem žingsniais.

1. Sudarome penkių kortelių schemą ir sujungiame jas linija, jei jos turi bent vieną bendrą simbolį šonuose. Schemos paaškinimas: C ir E jungiamos linija, nes abiejų kortelių šonuose yra juodas skritulys; A ir C nejungiamos, nes jų šonuose nėra nei vieno bendro simbolio. (Greta nubraižyta supaprastinta schema – grafas.)



Keturių kortelių dėliojimas sudarant kvadratą ir laikantis lietimosi šonų derinimo taisyklės, atitinka 4 žingsnių ciklą. Sąvokos „4 žingsnių ciklas“ paaškinimas: tai kurioje nors kortelėje prasidedantis kelias, einantis pagal linijas ir po 4 žingsnių vėl grįžtantis į kortelę, iš kurios pradėta.

2. Naudodamiesi schema (grafu), surandame visus galimus 4 žingsnių ciklus. Čia yra trys 4 žingsnių ciklai:

a) pirmasis eina iš A į B, iš D į E ir atgal į A (šis 4-žingsnis ciklas atitinka mūsų sprendinį);

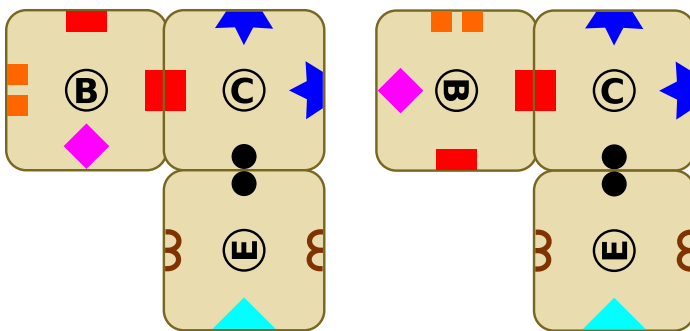


b) antrasis yra A-B-C-E-A;

c) trečiasis yra B-C-E-D-B.

Žinoma, visi cikliniai penkių raidžių perstatiniai (pradedant nuo bet kurios raidės, bet išlaikant eilės tvarką) yra ekvivalentūs, įskaitant ir atvirkštinius. Pavyzdžiui, A-B-C-E-A, B-C-E-A-B ir E-C-B-A-E yra ekvivalentūs.

Ar gali būti, kad antrasis arba trečiasis 4 žingsnių ciklai gali padėti rasti kitą mūsų galvosūkiio sprendimą? Abiejuose cikluose yra B-C-E ir yra tik dvi galimybės sujungti šias korteles tokia tvarka:



Nei A, nei D kortelės negali būti panaudotos 2x2 kvadratui užbaigti (ir, žinoma, atitikti taisyklę).

Tai informatika!

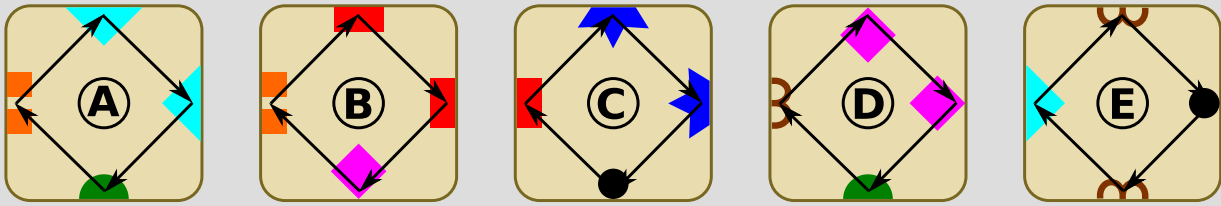
Sprendžiant sudėtingą uždavinį dažnai naudinga (laikiniai) nekreipti dėmesio į kai kurias detales. (Tai vadinama abstrakcija ir yra svarbus informatinio mąstymo komponentas).

Pirmajame žingsnyje nekreipėme dėmesio į simbolių formą, spalvas ir jų padėtį ant kortelių šonų. Domino tik tai, ar dvi kortelės turi bendrą simbolį, ar ne. Gauta diagrama (grafas) suteikė mums pakankamai informacijos, kad sprendimo paieška būtų šiek tiek trumpesnė. Liko tik trys galimi sprendimai.

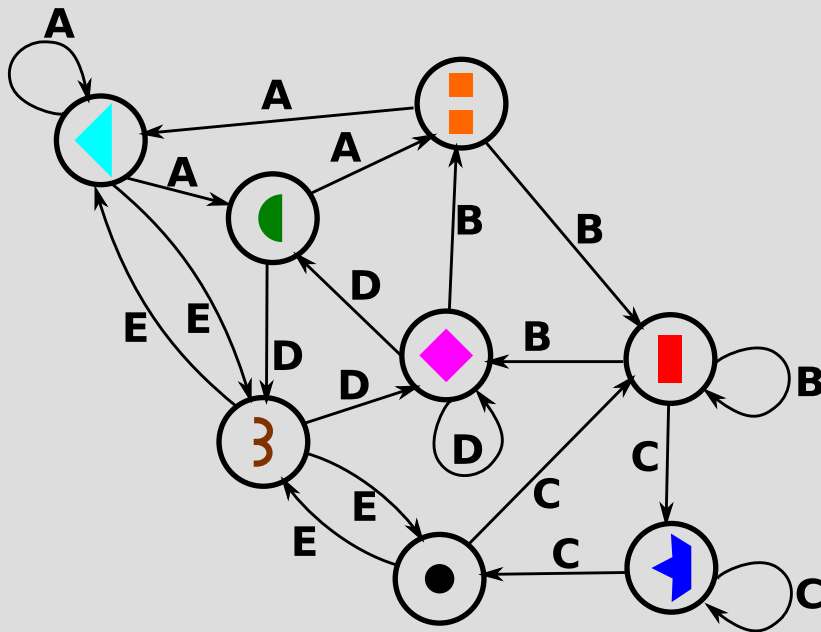
Grafai dažnai naudojami informatikoje kaip apdorojamų duomenų abstrakcija. Istorijos bėgyje daugybė žmonių sugalvojo gudrių ir greitų metodų, kaip nustatyti įvairias tokių grafų savybes, pavyzdžiui, lengvai rasti visus 4 žingsnių ciklus. Norint toliau spręsti šį uždavinį reikėjo antrojo žingsnio, kuriame vėl buvo naudojamas grafas. Sudėtingo uždavinio skaidymas į atskirus žingsnius, kuriuos galima lengviau išspręsti, vadinamas dekompozicija – tai vėl svarbus informatinio mąstymo komponentas.

*Alternatyvus sprendimas*

Galime naudoti kitokią schemą, kurią konstruojame taip: tos pačios kortelės šonų simbolius sujungiame rodykle tokiu būdu:



Dabar naudokime šias rodykles kaip jungtis naujoje scheme, kurioje kiekvienas simbolis pasitaiko tik vieną kartą (ši schema vadinama orientuotu grafu):

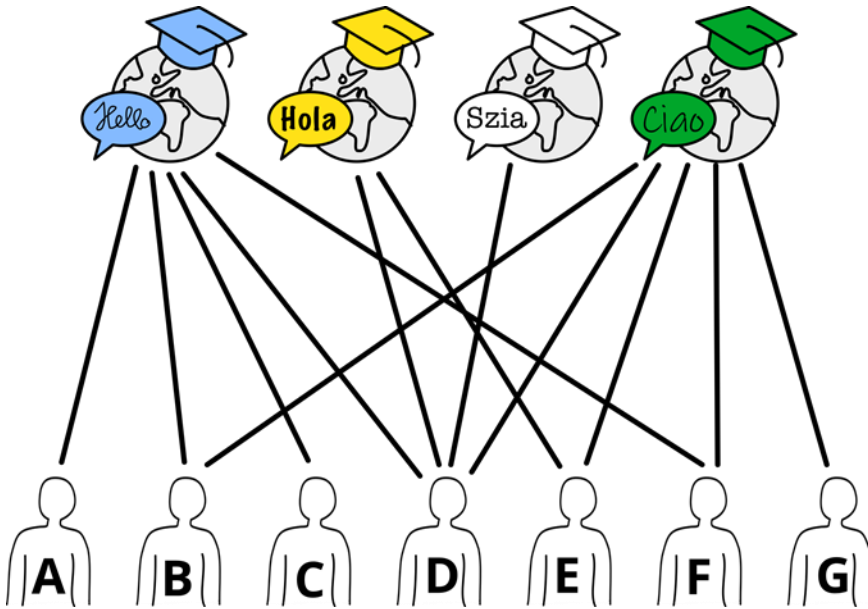


Ant atitinkamos rodyklės užrašėme kortelės pavadinimą (A, B, C, D, E). Teisingą sprendinį šioje diagramoje sudaro 4 žingsnių ciklas, tačiau šį kartą visos rodyklės turi eiti 4 žingsnių ciklo kryptimi. Randame ciklą A-E-D-B, kuris atitinka mūsų sprendinį.

Taip pat yra ciklas E-C-B-D, kuris yra „beveik sprendinys“, išskyrus tai, kad D rodyklė eina ne ta kryptimi. Yra kelias A-B-C-E, bet jis nesibaigia ten, kur prasidėjo (taigi nėra ciklas), o C rodyklė bet kuriuo atveju nukreipta ne ta kryptimi. Todėl pirmasis sprendinys yra vienintelis teisingas.

## 7. Darbų paskirstymas


Užsienio kalbų mokykla planuoja keturis vasaros kursus. Esami mokytojai gali mokyti šių kalbų (žr. schema):




Kiekvienas kursas turi turėti mokytoją ir kiekvienas mokytojas gali mokyti tik vieną kursą. Mokytojai, kurie nemoko jokio kurso, paskiriami dirbti kitur.

Kuris iš šių teiginių yra neabejotinai teisingas?

a) Jei B, F ir G nėra, vienas kursas turi būti atšauktas.


b) D mokys ispanų kalbos  kursą.

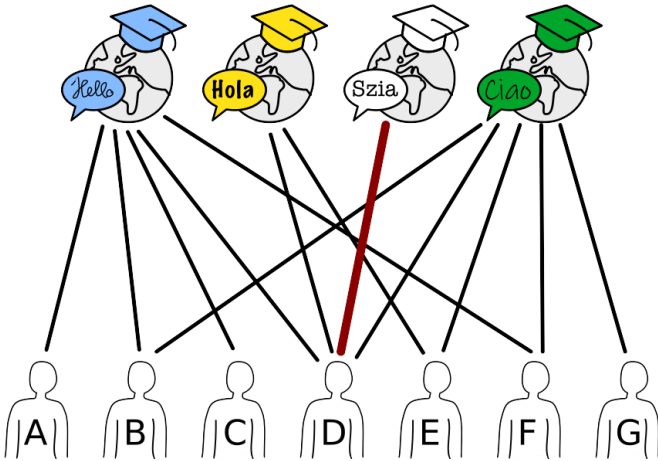
c) E mokys italų kalbos  kursą.


d) 4 mokytojai neturi kurso, kurį galėtų mokyti.

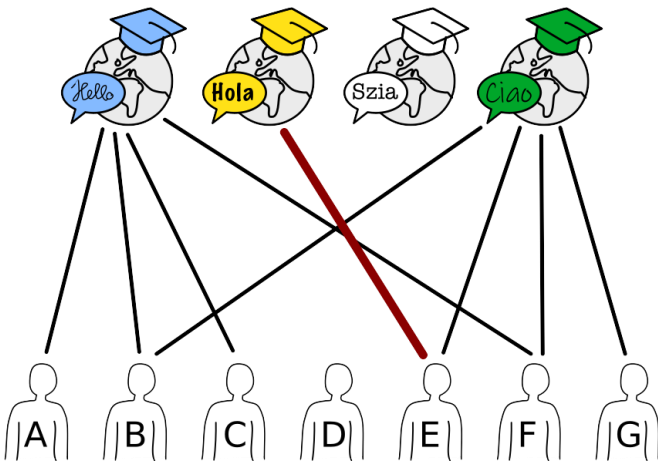
## Paaiškinimas

Teisingas atsakymas: A.

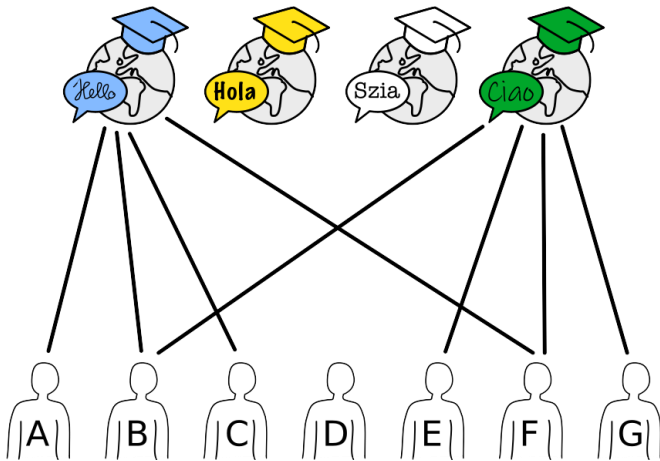
D yra vienintelis mokytojas, kuris gali mokyti vengrų  kalbos kursą. Taigi šis kursas užimtas ir D negali mokyti kitų kalbų kursų. (Atsakymas B yra klaidingas.)



E yra vienintelis mokytojas, kuris gali mokyti ispanų  kalbos kursą. Taigi šis kursas užimtas ir E negali mokyti kitų kalbų kursų. (Atsakymas C yra klaidingas.)



Likusius du kursus galima laisvai rinktis. Pavyzdžiui, A galėtų mokyti anglų ir B – italų. Tai reiškia, kad yra 4 mokytojai, kurie moko 4 kursus, taigi lieka  $7-4 = 3$  mokytojai, kurie neturi jokio kurso. (Atsakymas D yra klaidingas.).



Dabar panagrinėkime variantą A. Dar kartą pažvelgę į ankstesnį paveikslėlį pastebėsime, kad D turi mokyti vengrų kalbos, o E – ispanų kalbos, todėl likę italų kalbos mokytojai yra B, F ir G. Jei „pašalinsime“ B, F ir G, neliks italų kalbos mokytojo, taigi A variantas yra teisingas.

Tai informatika!

Kalbų kursų ir mokytojų duomenų struktūrai pavaizduoti šioje užduotyje panaudotas dvidalis grafas (angl. *bipartite graph*, *bigraph*). Tai yra ribojimų tenkinimo uždavinys, kai kintamiesiems priskiriant reikšmes atmetamos kitos galimybės. Uždavinį galima pavaizduoti naudojant tradicinį aibės užrašą, pavyzdžiui, ispanų = {D, E}.

Ribojimų tenkinimo uždaviniai būdingi daugeliui realaus pasaulio išteklių paskirstymo uždavinių ir galvosūkių (pvz., sudoku).

Šios užduoties esmė – mokytojams paskirstant kursus naudoti dvidalius grafus. Dvidaliai grafai naudoja dvi viršūnių kategorijas, kurios leidžia polinominio laiko algoritmu spręsti viršūnių atitikimo uždavinį.

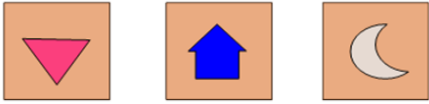
Informatinis mąstymas

Užduotyje naudojamas informatinio mąstymo konceptas – dekomponavimas nagrinėjant kiekvieno mokytojo darbą atskirai ir nepriklausomai nuo kitų darbų.

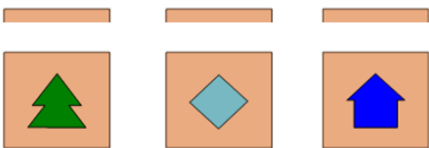
Sprendžiant užduotį naudojamas loginis mąstymas, teigiant „jei kurį nors kursą moko tik vienas mokytojas, jis turi mokyti šį kursą, net jei jis gali mokyti ir kitus kursus“. Tuo remiantis randamas algoritmas šiam uždaviniui išspręsti, o tuo ugdomas informatinis mąstymas.

## 8. Lobio dėžutė

Marija rado užrakintą dėžutę, kurioje paslėptas lobis. Norėdama atrakinti dėžutę, ji turi panaudoti teisingą trijų figūrų kombinaciją. Remdamiesi figūrų kombinacijų dešinėje parašytomis užuominomis, padėkite Marijai atrakinti dėžutę.



1. Viena figūra yra teisinga ir tinkamoje vietoje.



2. Nei viena figūra į atrakinančią kombinaciją neįeina.



3. Dvi figūros teisingos, bet yra netinkamoje vietoje.

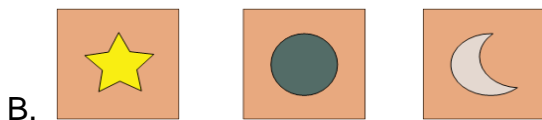


4. Viena figūra teisinga, bet netinkamoje vietoje.






5. Viena figūra teisinga, bet netinkamoje vietoje.


Kuri iš pateiktų kombinacijų atrakins lobio dėžutę?



## Paaiškinimas

Teisingas atsakymas: B.

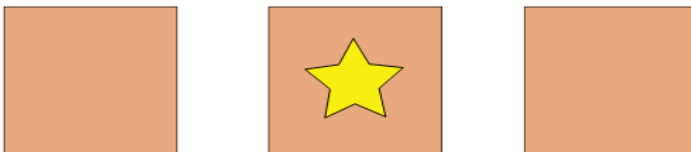
Pirmiausia eliminuosime figūras, kurios netinka kombinacijai, atrakinančiai dėžutę. Antroje užuominų rinkinio eilutėje matome, kad nei viena jos figūra į atrakinančią kombinaciją neįeina. Tai reiškia, kad eglutė , rombas  ir rodyklė  nepadės atrakinti dėžutės.

Paskutinės eilutės užuominoje rašoma, kad viena figūra teisinga, bet ji yra neteisingoje vietoje. Mes jau padarėme išvadą, kad eglutė ir rodyklė – netinkamos figūros, taigi likusi tinkama figūra yra žvaigždutė , tačiau ji yra neteisingoje vietoje.

Galimos žvaigždutės pozicijos:



ARBA



Galime tęsti kitų dviejų figūrų paiešką. Iš pirmos eilutės matome, kad viena figūra yra teisinga ir teisingoje vietoje. Atmetame rodyklę, kadangi ji nepriklauso kombinacijai, taigi teisinga bus viena iš šių kombinacijų.



ARBA



Trečioje eilutėje matome, kad dvi figūros teisingos, bet yra netinkamose vietose. Į galutinę kombinaciją turi būtinai patekti žvaigždutė, tačiau jos pozicija ne viduryje. Taigi vienintelė galima teisinga pozicija vaizduojama šiame paveiksle:



Radome vieną figūrą ir jos vietą. Tęsiame likusių dviejų figūrų paiešką.

Pirmoje eilutėje turėjome





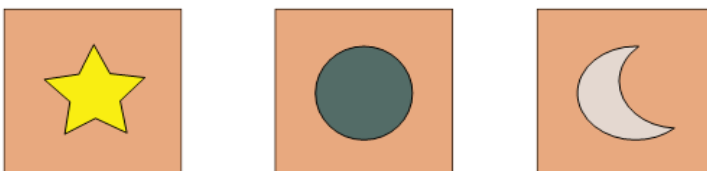
ARBA



Kadangi jau radome žvaigždutę ir jos vietą kairėje, trikampis negali būti pirmoje vietoje. Iš to gauname, kad mėnulis yra teisingoje vietoje.

Ketvirtoje eilutėje matome, kad teisinga viena figūra, tačiau ji yra neteisingoje vietoje.

Trikampis  jau yra atmestas, todėl vienintelė galima vieta yra viduryje. Širdelė  negali būti tinkama figūra, kadangi ji yra viduryje. Taigi darome išvadą, kad viduryje turi būti skritulys



Tai informatika!

Šablonų susiejimas informatikoje – tai duotos elementų sekos tikrinimas ieškant joje tam tikro šablono. Mūsų uždavinyje sekdami užuominas sukūrėme šabloną – radę teisingą šabloną, atrakinsime dėžutę. Realiame gyvenime šablonų susiejimas naudojamas, pavyzdžiui, tikrinant, ar žodžių rašyba teisinga.



## 9. Bebrų duomenys

Bebrų kaime gyvena keliolika šeimų. Bebras informatikas Bronius sukūrė kaimo gyventojų duomenų bazę, įrašydamas duomenis apie kiekvieną bebrą 16 bitų seka nuo b15 (kairėje) iki b0 (dešinėje):

- b15 \_ b12: keturi bitai šeimos numeriui;
- b11: vienas bitas lyčiai (0 = moteris, 1 = vyras);
- b10 \_ b4: septyni bitai svoriui (kilogramų skaičius);
- b3 \_ b2: du bitai darbuotojo kvalifikacijai (00 = namelių statyba, 01 = užtvankų statyba, 10 = maisto sandėliavimas, 11 = jaunų bebrų ugdymas);
- b1 \_ b0: du bitai mėgstamam valgiui (00 = medžių žievė, 01 = vandens augalai, 10 = žolės, 11 = viksvos).



Pavyzdžiui, seka 0100 0 0100101 10 01 reiškia, kad tai bebrė ir ji priklauso 4-ai šeimai, yra patelė (moteriškos lyties), sveria 37 kilogramus, yra įgudusi maisto sandėliavimo darbininkė ir mėgsta maitintis vandens augalais.

Informatikas Bronius užklausas duomenų bazėje formuluoja loginiais reiškiniais (0 = *false*, 1 = *true*).

Duotas toks reiškinys:

$$b_{11} \text{ and not } (b_{10}) \text{ and } b_9 \text{ and } b_8 \text{ and not } (b_3 \text{ and } b_2)$$

Kuris bebrų aprašas atitinka šį reiškinį?

- Patelės, sveriančios ne mažiau kaip 16 kg, kvalifikuotos maisto sandėliavimo darbuotojos.
- Patinai, sveriantys ne mažiau kaip 64 kg, kvalifikuoti namelių ar užtvankų statybininkai.
- Patinai, sveriantys nuo 40 iki 63 kg, kvalifikuoti namelių ar užtvankų statybininkai arba maisto sandėliavimo darbuotojai.
- Patinai, sveriantys ne daugiau kaip 39 kg, kvalifikuoti užtvankų statybininkai.

## Paaiškinimas

Teisingas atsakymas: C.

$b_{11} = 1$  reiškia „vyras“ (patinas),  $b_{10} = 0$  reiškia „sveria ne daugiau kaip 63 kg“,  $b_9 = 1$  ir  $b_7 = 1$  reiškia „sveria ne mažiau kaip  $(32 + 8)$  kg“,  $b_3 = 0$  arba  $b_2 = 0$  rodo, jog neįtraukta tik jaunų bebrų ugdymas ( $b_3 = 1$  ir  $b_2 = 1$ ).

(De Morgano dėsnis nusako  $\text{not}(b_3 \text{ and } b_2)$  ekvivalentiškumą reiškiniui  $\text{not}(b_3)$  or  $\text{not}(b_2)$ .)

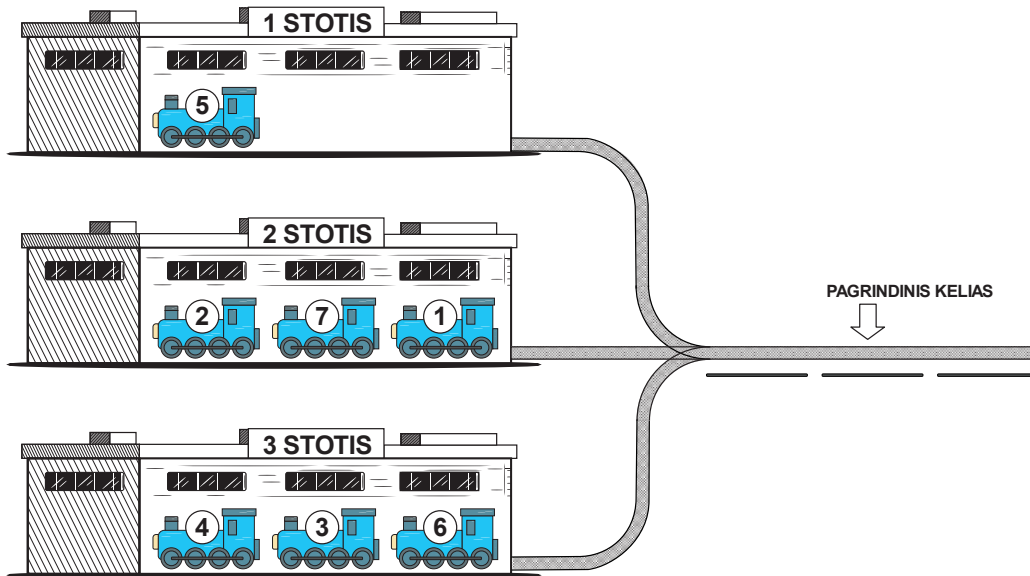
Atsakymas A neteisingas, nes jau pirmas reiškinių elementas  $b_{11} = 1$  netinka, todėl ir visas reiškinys, nepriklausomai nuo kitų elementų, yra neteisingas (*false*). Atsakymas B neteisingas, nes  $b_2 = 0$ , kas reiškia, jog tinka ne tik kvalifikuoti namelių arba užtvankų statybininkai, bet ir maisto sandėliavimo darbuotojai. Atsakymas D neteisingas, nes neteisingai aprašyta ne tik kvalifikacija, bet ir svoris, nes  $b_9 = 1$  ir  $b_7 = 1$  reiškia „sveria ne mažiau kaip 40 kilogramų“.

Tai informatika!

Aprašytoje paprastos duomenų bazės užklausoje gali būti naudojamas loginis reiškinys (teiginių logikos formulė), t. y., jis gali būti laikomas paieškos raktu. Paprastai tokia formulė identifikuoja duomenų poaibį (tokias situacijas galima grafiškai pavaizduoti Veno diagramomis). Pavyzdžių galima rasti 1943–1944 m. Konrado Cūzės (Konrad Zuse) darbuose. (Konradas Cūzė buvo vokiečių statybų inžinierius, tačiau didžiausias jo pasiekimas – pirmasis programa valdomas elektromechaninis kompiuteris, pradėjęs veikti 1941 m.). Dauguma šiuolaikinių programavimo kalbų turi unarinį loginį operatorių *not* ir dvinarinius loginius operatorius *and*, *or*. Daugelio kompiuteriuose naudojamų skaitmeninių komponentų (multiplexorių, sumatorių, aritmetinių loginių blokų ir t. t.) pagrindas yra loginės grandinės (schemos), sudarytos iš atitinkamų loginių elementų. 1913 m. Henris Šeferis (Henry Maurice Sheffer) parodė, kad visus loginius operatorius galima išvesti iš vieno (*nand*):  $A \text{ nand } B$  yra ekvivalentus  $\text{not}(A \text{ and } B)$ , taigi  $A \text{ nand } A$  yra ekvivalentus  $\text{not}(A)$ . 1880 m. Čarlzas Pirsas (Charles Sanders Peirce) pirmasis tyrinėjo loginius reiškinis, siedamas juos su semiotika. Kaip ir *nand* elementai, *nor* elementai ( $A \text{ nor } B$  ekvivalentus  $\text{not}(A \text{ or } B)$ , taigi  $A \text{ nor } A$  ekvivalentus  $\text{not}(A)$ ) yra universalūs loginiai elementai; juos galima sujungti į bet kokius kitus loginius elementus. Daugelis elektroninių grandinių buvo sukurtos tik iš *nand* arba *nor* elementų.

## 10. Garvežiai

Jonas turi 7 žaislinius garvežius (sunumeruotus nuo 1 iki 7) ir 3 stotis, kuriose jis gali pastatyti savo garvežius. Dabartinė garvežių išdėstymo schema parodyta paveikslėlyje.



Jonas gali atlikti šiuos veiksmus:

OUT(X) – perkelti pirmą garvežį nuo X stoties dešinės pusės į pagrindinį kelią, kad jis taptų kairiausiu garvežiu pagrindiniame kelyje.

IN(X) – perkelti kairiausią garvežį iš pagrindinio kelio į X stotį, kad jis taptų pirmuoju garvežiu stoties dešinėje.

Vienoje stotyje, o taip pat ir pagrindiniame kelyje, vienu metu gali būti ne daugiau nei 3 garvežiai.

Kuris veiksmų rinkinys leidžia garvežius su numeriais 1, 2, 3 išdėstyti iš kairės į dešinę 1-oje stotyje?

- A. OUT(1) - OUT(2) - IN(1) - OUT(2) - OUT(2) - IN(1) - OUT(3)
- B. OUT(1) - OUT(2) - IN(1) - OUT(2) - OUT(2) - IN(1) - OUT(3) - IN(2) - OUT(3) - IN(1)
- C. OUT(1) - OUT(2) - IN(1) - OUT(2) - IN(1) - OUT(3) - IN(2) - OUT(3) - IN(1)
- D. OUT(1) - OUT(2) - IN(1) - OUT(2) - OUT(2) - IN(1) - OUT(3) - OUT(3) – IN(1)

## Paaiškinimas

Teisingas atsakymas: B.

Lentelėje pateiktos garvežių būsenos (iš kairės į dešinę) kiekvienoje stotyje ir pagrindiniame kelyje po atliktų veiksmų:

Veiksmas	1 stotis	2 stotis	3 stotis	Pagrindinė linija
	5	2 7 1	4 3 6	
OUT(1)		2 7 1	4 3 6	5
OUT(2)		2 7	4 3 6	1 5
IN(1)	1	2 7	4 3 6	5
OUT(2)	1	2	4 3 6	7 5
OUT(2)	1		4 3 6	2 7 5
IN(1)	1 2		4 3 6	7 5
OUT(3)	1 2		4 3	6 7 5
IN(2)	1 2	6	4 3	7 5
OUT(3)	1 2	6	4	3 7 5
IN(1)	1 2 3	6	4	7 5

Atsakymas A neteisingas, nes į 1-ą stotį nepridėti 3 garvežiai. Atsakymas C neteisingas, nes į 1-ą stotį perkeltas 7-as garvežys. Atsakymas D neteisingas, nes atlikus veiksmus OUT(1) - OUT(2) - IN(1) - OUT(2) - OUT(2) - IN(1) - OUT(3) - OUT(3) - IN(3) pagrindinėje linijoje yra daugiau nei 3 garvežiai.

Tai informatika!

Dėklas, arba rietuvė (angl. *stack*) yra tiesinė duomenų struktūra, kurioje yra operacijos PUSH (padėti), kuri prideda elementą ant dėklo viršaus, ir POP (paimti), kuri išima viršutinį elementą iš dėklo. Kiekviena stotis ir pagrindinis kelias yra dėklai. OUT(X) yra X stoties dėklo POP operacija, po kurios įvyksta šio garvežio PUSH operacija į pagrindinio kelio dėklą. IN(X) yra pagrindinio kelio dėklo POP operacija, po kurios įvyksta šio garvežio PUSH į X stotį.

Fiksuoto gylio dėklai naudingi spręsti uždaviniams, susijusiems su ribotais resursais ir fiziniu daiktų judėjimu, pavyzdžiui, sandėliuose. Atsakymų variantai pasinaudoja traukinių stotimi ir pagrindiniu keliu kaip baigtinės būsenos sistema, kurioje kiekvienas veiksmas deterministiškai keičia būseną. Pavyzdžiui, visi atsakymai prasideda OUT(1) - OUT(2) - IN(1) - OUT(2), todėl iš šios sekos gauta būseną gali būti apskaičiuota tik kartą ir naudojama visiems atsakymams.

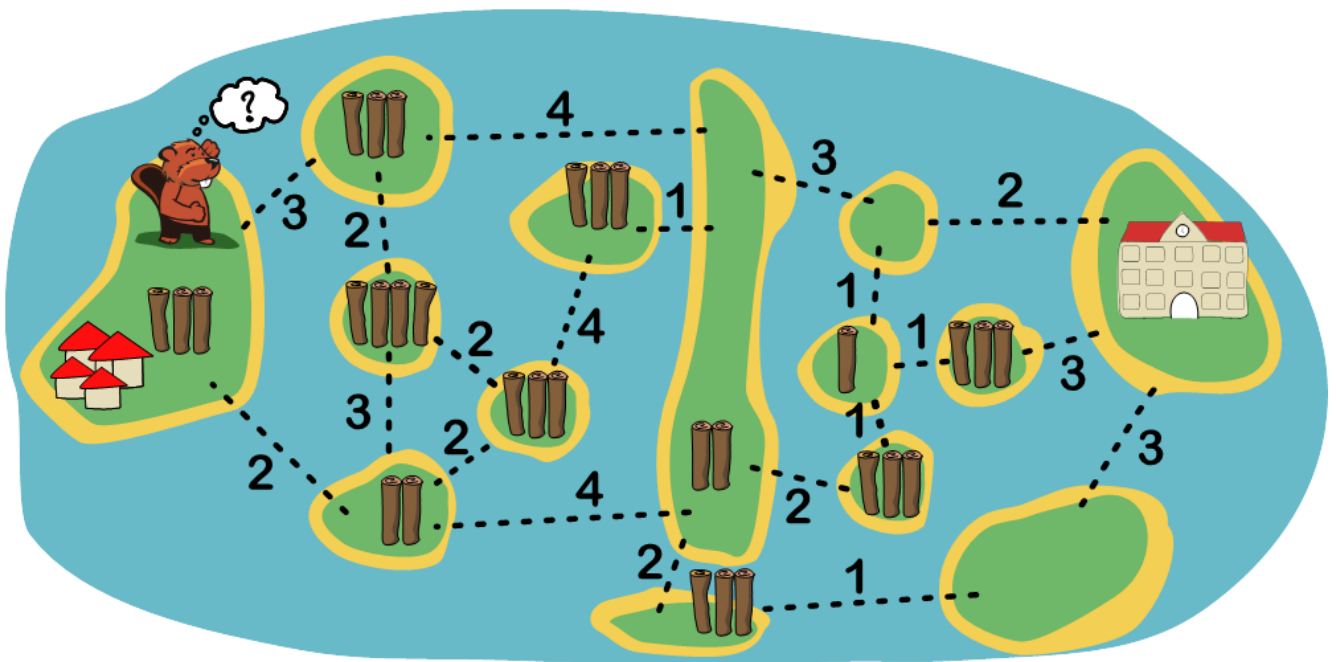
Informatinis mąstymas

Šiai užduočiai būdingas algoritminis mąstymas, nes perkėlimų skaičiavimas gali būti traktuojamas kaip algoritmas: rasti pirmą garvežį ir jį pridėti prie pirmos stoties, tada antrą garvežį ir trečią garvežį, naudojant operacijas, duotas dėklo sąrašė.

## 11. Tiltų statyba

Bronius stato tiltus. Naujas jo projektas – pastatyti tiltus, kad Broniaus kaimo  mokiniai galėtų nukeliauti į naująją mokyklą .

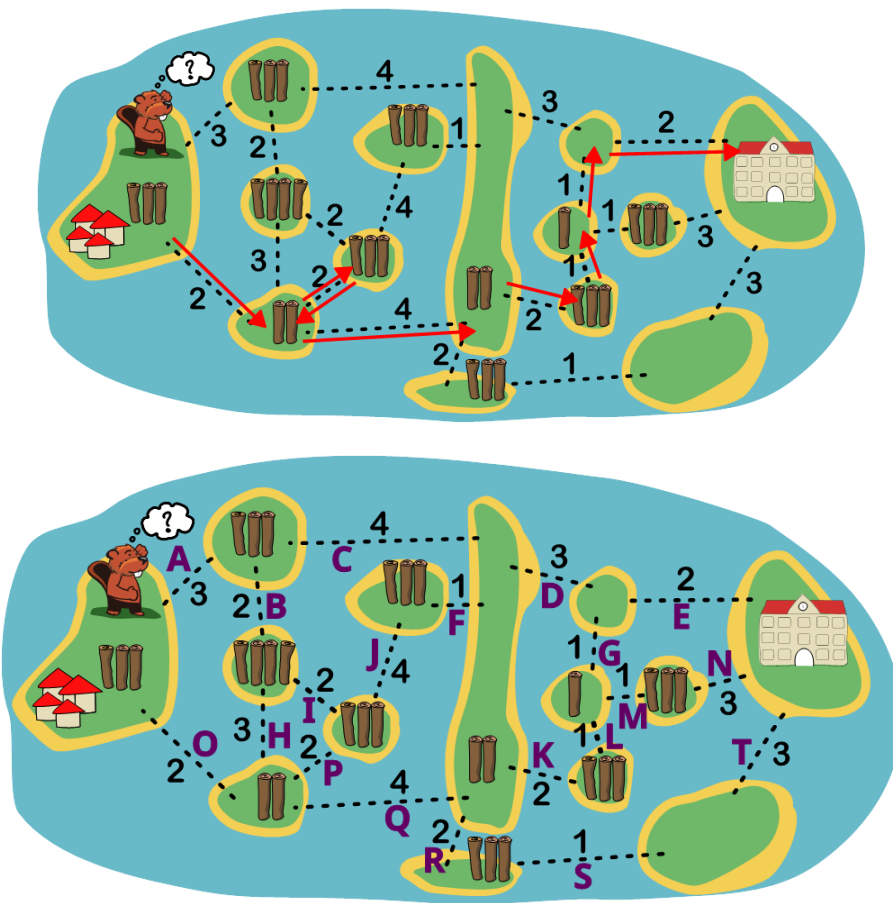
- Bronius nemoka plaukti, todėl pirmiausia jis turi pastatyti tiltą, kuriuo būtų galima pereiti į kitą salą tiek kartų, kiek jam reikės.
- Kiekvieno tilto statybai reikia tam tikro skaičiaus rąstų. Savo kaime Bronius turi tik 3 rąstus. Bronius gali keliauti jo pastatytais tiltais iš vienos salos į kitą ir pasiimti daugiau rąstų. Paveiksle pavaizduota, kiek rąstų reikia tiltui pastatyti tarp salų ir kiek rąstų yra kiekvienoje saloje.
- Kiekvienas rąstas gali būti naudojamas tik vieną kartą, tačiau nebūtina panaudoti visų turimų rąstų.



Kuriuos tiltus turėtų pastatyti Bronius, kad sunaudotų mažiausią rąstų skaičių ir mokiniai galėtų nukeliauti iš kaimo į mokyklą? Kiek rąstų jam užteks?

## Paaiškinimas

Teisingas atsakymas: 14 rąstų.



Teisingas kelias užrašyti bendrą surinktų ir panaudotų rąstų skaičių kiekviename žingsnyje būtų toks (kiekviename žingsnyje surinkti rąstai  $\geq$  panaudoti rąstai):

(3, 2) (5, 4) (8, 4) (8, 8) (10, 10) (13, 11) (14, 12) (14, 14).

Įrodysime, kad tai yra mažiausias įmanomas rąstų kiekis. Surasime visus kelius, jungiančius kaimą su mokykla, kuriuose yra  $\leq 14$  panaudotų rąstų. Svarbu tai, kad visi keliai turi eiti per didžiąją salą tvenkinio viduryje, todėl uždavinį galima padalyti į dvi dalis. Keliui iš kaimo į didžiąją salą reikia ne mažiau kaip 6 rąstų, o keliui iš didžiosios salos į mokyklą reikia ne mažiau kaip 5 rąstų (šiuos kelius vadiname apatiniais rėžiais).

Pirmiausia išsprendžiame kairiąją dalį naudodamiesi dešiniuoju apatiniu rėžiu. Tai reiškia, kad ieškome visų įmanomų kelių, kurie sunaudoja  $\leq 9$  rąstus (nes bet koks kairės dalies sprendimas  $\geq 10$  iš viso sunaudotų bent 15 rąstų).

Kairėje dalyje turime šiuos kelius, kurie jungiasi su didžiąja sala ir naudoja  $\leq 9$  rąstus:

$\{A, C\} = (6, 7)$ ,  $\{A, B, C\} = (10, 9)$ ,  $\{O, Q\} = (5, 6)$ ,  $\{O, H, Q\} = (9, 9)$ ,  $\{O, P, Q\} = (8, 8)$ ,  $\{O, P, J, F\} = (11, 9)$

Pašalinus netinkamus sprendinius, kuriuose surinktų rąstų mažiau negu panaudotų rąstų, lieka:

$$\{A, B, C\} = (10, 9), \{O, H, Q\} = (9, 9), \{O, P, Q\} = (8, 8), \{O, P, J, F\} = (11, 9)$$

Tuomet dešinę dalį sprendžiame naudodami apatinės kairiosios dalies 8 panaudotų rąstų rėžį, ir ieškome visų kelių, kuriems reikia  $\leq 6$  rąstų:

$$\{D, E\} = (2, 5), \{D, G, E\} = (3, 6), \{K, L, G, E\} = (6, 6), \{R, S, T\} = (5, 6)$$

Sujungiame kairiosios ir dešinėsios dalių sprendinius, kad rastume visas įmanomas poras, kuriose būtų  $\leq 14$  panaudotų rąstų, ir randame tik

$$\{O, P, Q\} = (8, 8) + \{K, L, G, E\} = (6, 6) \rightarrow (14, 14),$$

nes visos kitos poros turi daugiau paimtų rąstų nei panaudotų.

Tai informatika!

Žemės plotus (salas) ir galimus statyti tiltus galima pavaizduoti grafo viršūnėmis ir briaunomis. Kiekvienoje saloje surinktų rąstų ir panaudotų kiekvienam tiltui statyti rąstų skaičių galima priskirti atitinkamoms viršūnėms ir briaunoms. Broniaus kelią galima pavaizduoti kaip briaunų sąrašą tiltų pastatymo tvarka. Sprendžiant šią užduotį galima naudoti kelio paieškos algoritmus su išteklių apribojimais. Išsami visų tinkamų kelių paieška būtų rezultatyvi, tačiau užtruktų eksponentinį laiką grafo dydžio atžvilgiu. Taip pat šią užduotį galima suformuluoti kaip „skaldyk ir valdyk“ metodo paiešką, kaip tai daroma sprendimo paaiškinime. Taikant šį metodą naudojami apribojimų optimizavimo metodai – kelio nuoseklumo ir šakų bei ribų (angl. *branch-and-bound*) metodas. Šiais metodais grindžiami sudėtingiausių realaus pasaulio išteklių paskirstymo uždavinių, galvosūkių (pvz., sudoku) ir dirbtiniu intelektu paremtų žaidimų sprendimai.

Informatinis mąstymas

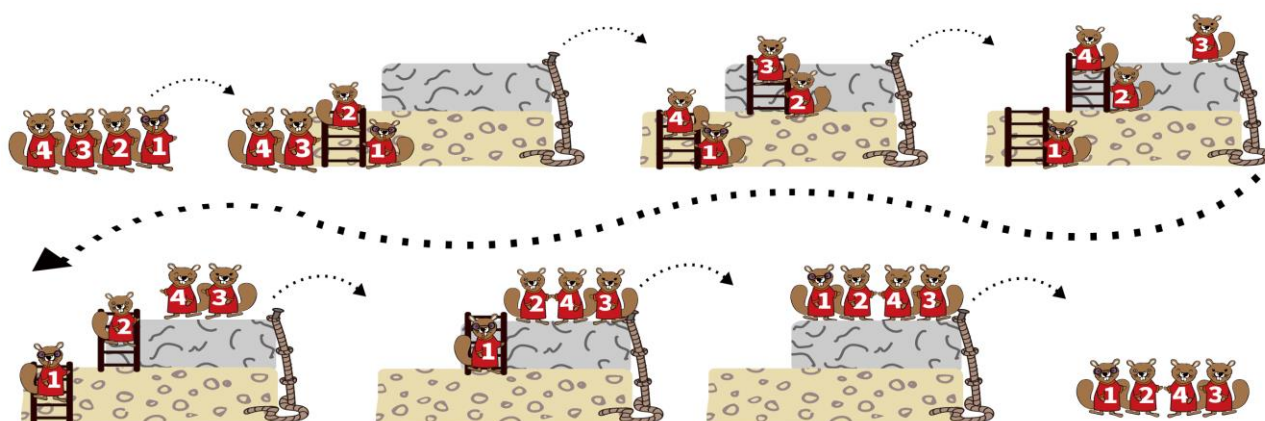
Abstrahavimas (salos dydis ar forma neturi reikšmės) – todėl tinka vaizduoti grafu.

Dekomponavimas – uždavinį galima padalyti į du panašaus pobūdžio uždavinius (kairioji ir dešinioji grafo dalys). Algoritminis mąstymas – kelio paieška – yra polinominio laiko nuoseklusis algoritmas ir reikalauja kaupiamųjų išteklių apribojimų. Taikant išsamios paieškos metodą sudėtingumas būtų eksponentinis (atsižvelgiant į tai, kad mūsų kelias yra medis, o grafe gali egzistuoti iki  $n^n - 2$  medžių) ir neefektyvus tokio dydžio uždaviniui spręsti.

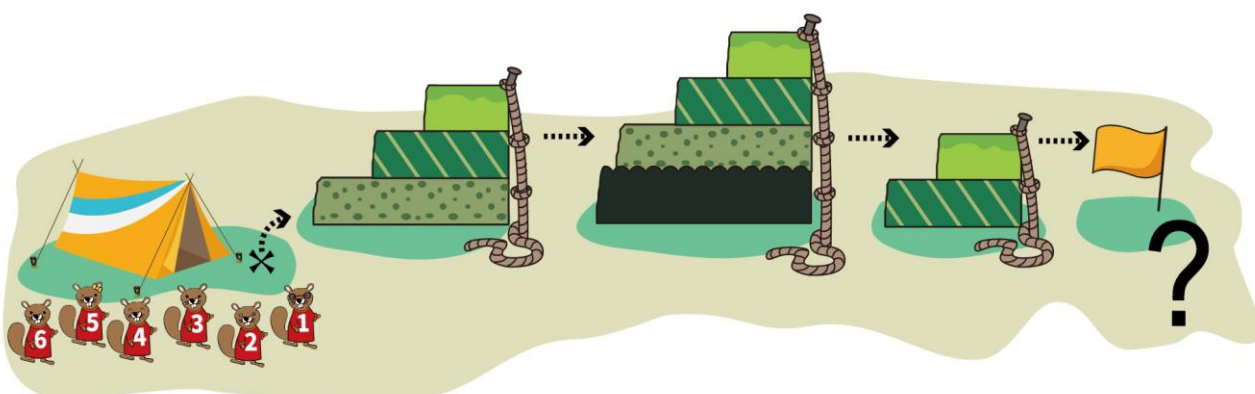
## 12. Į kalnus

Ruošdamiesi žygiui, bebrai treniruoiasi lipti stačiomis uolomis. Bebrai rikiuojasi ir eina eilute vienas paskui kitą. Kiekvienas bebras nešasi kopėčias. Kai prieina stačią uolą, pirmas eilėje esantis bebras pastato kopėčias, kad galėtų lipti kiti. Kiekvienas pastatęs kopėčias bebras laukia vietoje kol visi, kas nelaiko pakeltų kopėčių, užlipa į uolos viršūnę. Kiekvienas kopėčias pastatęs bebras savo kopėčiomis lipa į kitą lygį. Taip jie visi lipa tol, kol pasiekia uolos viršūnę. Kiekvienas bebras jungiasi į eilės pabaigą. Baigę kopti į uolą bebrai nusileidžia žemyn naudodamiesi virve ir laikydamiesi savo eilės tvarkos.

Paveiksle parodyta, kaip jie kyla.



Šeši bebrai ruošiasi į žygį vietovėje, pavaizduotoje paveiksle. Jų tikslas – vėliava ir klaustuku pažymėta vieta. Nutempkite ten bebrus jų atvykimo į tikslą eilės tvarka.





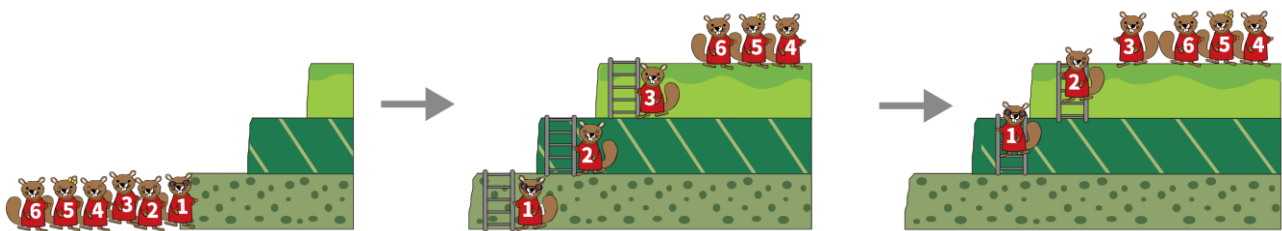
## Paaiškinimas

Teisinga bebrų eilės tvarka: 365412.

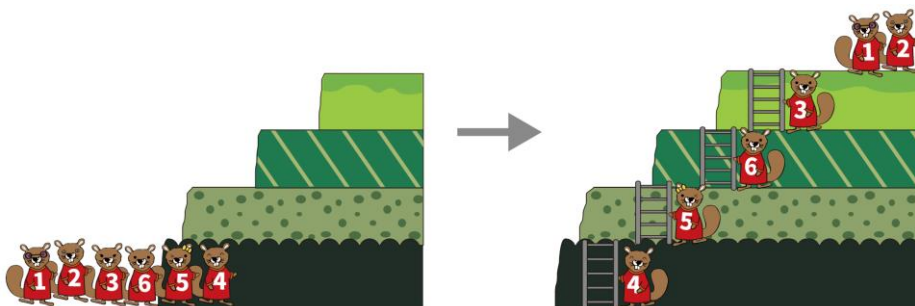
Bebrai pradeda žygį eilės tvarka: 123456. Taip jie atvyksta prie pirmosios 3-jų lygių uolos.

Bebras, kurio numeris 1, laiko savo kopėčias 1-ame lygyje, 2-as bebras laiko savo kopėčias 2-ame lygyje, 3-as bebras savo kopėčias laiko 3-ame lygyje, o 4, 5 ir 6 bebrai šia tvarka kyla į viršūnę.

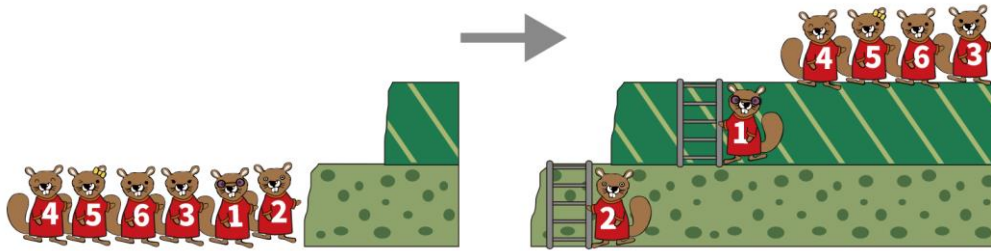
Pastariesiems bebrams pasiekus viršūnę, 3-as, 2-as ir 1-as bebrai lipa į viršų ta pačia tvarka, kol visi jie atsiduria pirmosios uolos viršūnėje. Po šio kopimo bebrų eilės tvarka yra 456321.



Tokia tvarka bebrai atvyksta prie antros 4-ių lygių uolos. 4-as bebras, būdamas pirmas eilėje, laiko kopėčias 1-ame lygyje, 5-as bebras – 2-ame lygyje, 6-as bebras – 3-ame lygyje, 3-as bebras – 4-ame lygyje, todėl 2-as ir 1-as bebrai gali šitokia tvarka užlipti į antrosios uolos viršūnę. 2-am ir 1-am bebrams pasiekus viršūnę, 3-as, 6-as, 5-as ir 4-as bebrai ta pačia tvarka lipa aukštyn, kol visi jie atsiduria ant antrosios uolos viršūnės. Dabar bebrų eilės tvarka yra tokia: 213654.



Bebrai prie paskutiniosios dviejų lygių uolos atvyksta tokia tvarka: 213654. Tad 2-as bebras yra pirmas eilėje, jis laiko kopėčias 1-ame lygyje, 1-as bebras – 2-ame lygyje, o kiti bebrai lipa į viršų eilės tvarka: 3, 6, 5 ir 4. Šiems bebrams pasiekus viršūnę, ta pačia tvarka į viršų lipa 1-as ir 2-as bebrai.



Įkopus į paskutinę uolą bebrų eilės tvarka yra 365412. Tai reiškia, kad bebrai atvyksta į pažymėtą tikslą tokiu eiliškumu: 365412.



Tai informatika!

Tai, kaip šiame uždavinyje bebrai lipa į uolas, atspindi dėklo ir eilės duomenų struktūras informatikoje.

Dėklas ir eilė yra dviejų tipų duomenų struktūros. Duomenys dėkle laikomi panašiai, kaip ir objektai dėkle realiame gyvenime. Dėklo duomenų struktūra veikia pagal principą: „paskutinis į maišą, pirmas iš maišo“, t. y., duomenys į ją dedami iš eilės tokia tvarka, kokia pateikiami, tad skaityti arba pašalinti galima tik paskiausiai įdėtą duomenį.




Į eilės duomenų struktūrą nauji duomenys dedami jų gavimo tvarka (į eilės pabaigą), o išimamas pirmasis į eilę įdėtas duomuo (esantis eilės pradžioje). Kitaip tariant, jei duomenis (objektus) iš eilės išimame vieną po kito, pirmas į eilę padėtas duomenų objektas bus išimtas pirmas.

Informatinis mąstymas

Mokiniai gali taikyti šablonų (struktūrų) atpažinimą ir išsiaiškinti, kokia tvarka bebrai pakelia kopėčias ir kokia tvarka lipa į uolas. Mokiniai taiko algoritminį mąstymą, kad galėtų vadovautis komandomis ir susekti bebrų eilės tvarką.





### 13. Magiška šalis

Magiškoje šalyje yra keturių rūšių stebuklingi daiktai: stebuklingos skrybėlės, krištoliniai rutuliai, stebuklingos knygos ir stebuklingi eliksiyrai. Magiškas skrybėles ir krištolinius rutulius galima transformuoti dviem skirtingais būdais. Lentelėje parodyta, kas sukuriama iš objektų šio proceso metu – būtent ten, kur jie buvo prieš tai, ir būtent tokia tvarka, kokia parodyta:

	<i>Prieš</i>	<i>Po</i>
1		
2		
3		
4		

Transformacijos gali vykti bet kiek kartų ir bet kokia tvarka. Taigi iš vieno stebuklingo objekto gali susidaryti ilga objektų seka.


Kuris derinys **NEGALI** atsirasti iš vienos stebuklingos skrybėlės?

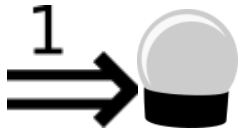
- A. 
- B. 
- C. 
- D. 

Paaiškinimas

Teisingas atsakymas: B.

Iš vienos stebuklingos skrybėlės gali susidaryti A, C ir D variantuose pateiktas išdėstymas:

Atsakymas A: 

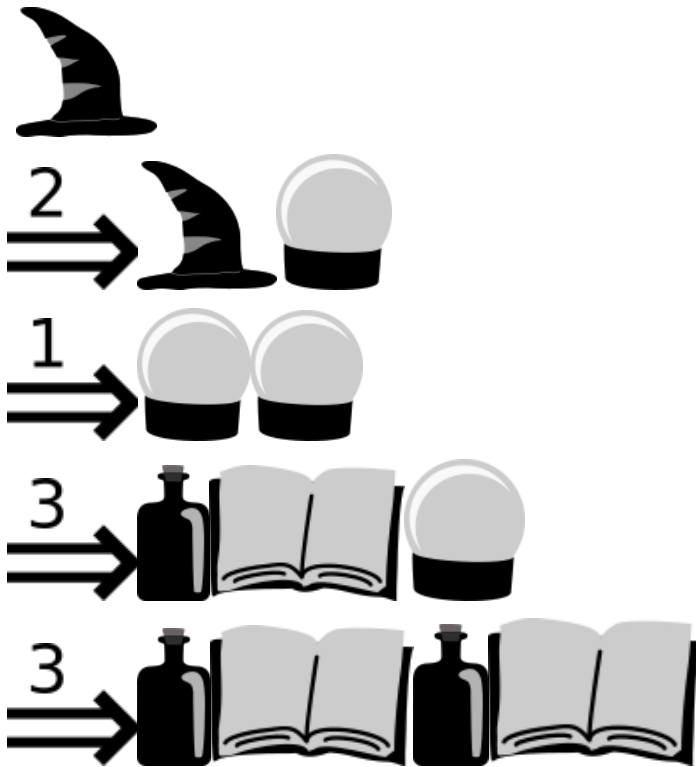


Atsakymas C:





Atsakymas D:

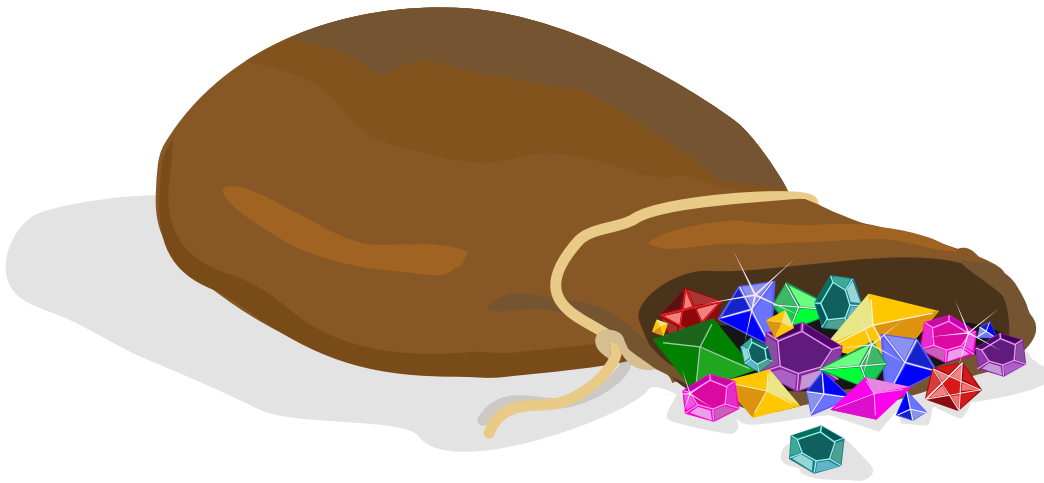


Tai informatika!

Magiškas transformacijas galima laikyti taisyklių rinkiniu, naudojamu Magiškos šalies objektų modeliams sukurti. Kompiuterių moksle bekontekstinė gramatika yra vienas iš įrankių, kurį galima naudoti taisyklėms, generuojančioms modelius, aprašyti. Konteksto neturinčiomis gramatikomis galima aprašyti kalbas (tiek formalias, tiek natūralias), o pakartotinai taikant gramatikos taisykles galima generuoti kalbą sudarančius žodžius (arba eilutes).

Šioje užduotyje reikėjo nustatyti, kuris iš pateiktų žodžių nėra Magiškos šalies kalbos dalis.

## 14. Gražiausias brangakmenis



Trojus kiekvieną savo kolekcijos brangakmenį įvertino pagal jo grožį nuo paties gražiausio iki prasčiausio.

Sara apžiūrinėja Trojaus kolekciją, tačiau nežino, kaip jis įvertino brangakmenius.

Mergaitė nori išsiaiškinti, kuris brangakmenis iš visų turimų Trojui pats gražiausias. Sara sugalvoja ir įgyvendina tokį planą:

- mergaitė pasirenka keturis Trojaus brangakmenius ir klausia: „Kuris iš šių keturių brangakmenių tau gražiausias?“
- Sara vėl pasirenka keturis brangakmenius ir užduoda tą patį klausimą antrą kartą.
- Sara trečią kartą pasirenka keturis brangakmenis ir užduoda tą patį klausimą paskutinį kartą.

Pastaba. Kai antrą ir trečią kartą Sara renkasi po keturis brangakmenius, ji gali panaudoti ir tuos brangakmenius, kuriuos jau buvo pasirinkusi pirmą ar antrą kartą.

Koks didžiausias brangakmenių skaičius gali būti kolekcijoje, kad Sara teisingai nustatytų Trojaus išsirinktą gražiausią brangakmenį?

- A. 8
- B. 10
- C. 11
- D. 12

## Paaiškinimas

Tarkime, kad Trojaus kolekcijoje yra 10 brangakmenių ir Sara naudoja strategiją kiekvieną kartą pasirinkdama po 4 brangakmenius. Po pirmo klausimo ji gali nustatyti gražiausią tik iš 4 brangakmenių. Lieka 6 brangakmeniai ir pasirinkus iš jų vėl 4, galima nustatyti gražiausią iš jų. Lieka 2 brangakmeniai, kurių kiekvienas taip pat gali būti pats gražiausias kolekcijoje. Todėl trečiuoju klausimu pasirinkdama gražiausius iš pirmų dviejų atsakymų ir likusius 2, Sara nustatys patį gražiausią Trojaus brangakmenį. Naudojantis šia strategija neįmanoma nustatyti gražiausio brangakmenio, jei jų skaičius viršija 10.

Pavyzdžiui, jei yra 11 brangakmenių, tai pasirenkant po 4 (4 nauji; 4 nauji; 2 nauji ir po vieną iš pirmų dviejų atrinktų), liks vienuoliktas brangakmenis, apie kurį Sara negaus jokios informacijos, nes kiekvieno brangakmenio galimybė būti pačiu gražiausiu Sarai yra vienoda.

Tai informatika!

Ši užduotis reikalauja strategijos, kuri leistų visada teisingai nustatyti patį gražiausią brangakmenį, kai brangakmenių skaičius maksimalus. Strategija – tai algoritmas, kuris nurodo užduoties sprendimo veiksmus.

Didžiausio brangakmenių skaičiaus radimo strategija turi tam tikrų apribojimų:

1. Nustatyti mėgstamiausią iš pasirinktų keturių brangakmenių.
2. Galima tik tris kartus nagrinėti pasirinktus brangakmenių rinkinius iš keturių elementų.

Yra daug algoritmų, kurie gerai veikia tik tam tikrais atvejais. Pavyzdžiui, rikiavimo algoritmas gali rikiuoti teigiamus sveikuosius skaičius, bet ne neigiamus sveikuosius skaičius. Kai kurie maršrutų paieškos algoritmai gali rasti trumpiausią maršrutą tarp dviejų objektų, tačiau tai daro per lėtai, kai naudojami žemėlapiai su tūkstančiais susijusių objektų. Tokiu atveju dažnai pasirenkami greitesni algoritmai, kurie randa vieną iš trumpiausių maršrutų, bet nebūtinai patį trumpiausią. Geras greitas sprendimas yra geriau nei joks sprendimas!

Apskritai yra daugybė būdų, kaip įvertinti algoritmus, kurie naudojami kompiuterinėse sistemose. Pavyzdžiui, paprastai vertinama programos (algoritmo) veikimo trukmė arba kiek kompiuterio atminties programa gali užimti, kol yra vykdoma užduotis.

## 15. Automobilių numeriai



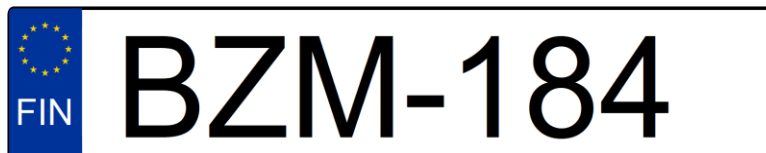
Skirtingose šalyse naudojami skirtingo dizaino ir užrašų valstybiniai automobilių numeriai. Dažniausiai naudojamos angliškos abėcėlės raidės (jų abėcėlėje yra 26) ir skaitmenys.

Bebras daro prielaidą, kad visų automobilių numeriai sudaryti pagal pateiktą schemą ir raidžių vietose gali būti bet kuri angliškos abėcėlės raidė (A–Z), o skaitmenų pozicijose – bet

kuris skaitmuo (0–9). Taip pat jis mano, kad visos kombinacijos yra leistinos. Bebras nekreipia dėmesio į mažas raides ir kitus simbolius.

Kuri iš valstybių galėtų užregistruoti daugiausiai automobilių?

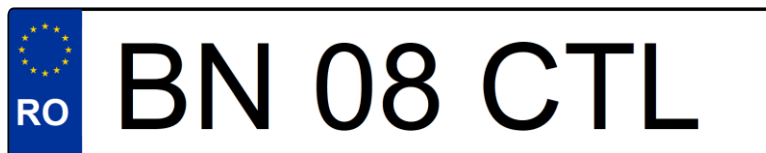
A. Suomija



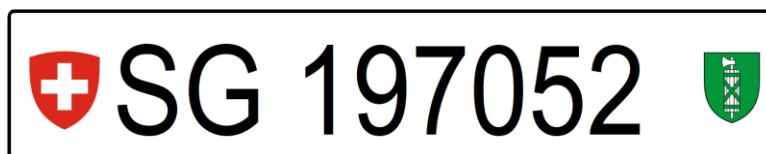
B. Slovakija



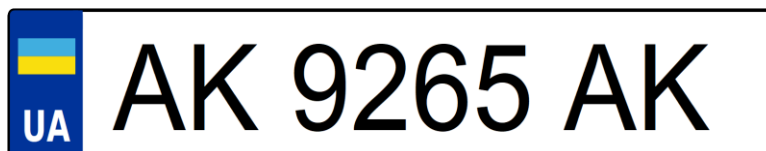
C. Rumunija



D. Šveicarija



E. Ukraina





## Paaiškinimas

Atsakymas: E (Ukraina).

Norėdami rasti teisingą atsakymą, turime palyginti galimų kombinacijų skaičių. Raidžių ir skaitmenų eiliškumas nėra svarbus.

- A: 3 raidės ir 3 skaitmenys →  $26^3 \times 10^3$ ;
- B: 4 raidės ir 3 skaitmenys →  $26^4 \times 10^3$ ;
- C: 5 raidės ir 2 skaitmenys →  $26^5 \times 10^2$ ;
- D: 2 raidės ir 6 skaitmenys →  $26^2 \times 10^6$ ;
- E: 4 raidės ir 4 skaitmenys →  $26^4 \times 10^4$ .

Skaičiai dideli, mintinai skaičiuoti nelengva. Todėl norėdami greitai surasti didžiausią skaičių palyginkime skaičius, kurių laipsnių rodikliai panašūs. Suradę mažesnę skaičių, iš galimų atsakymų jį pašalinsime. Norėdami palyginti skaičius galime juos dalinti vieną iš kito. Jei dalmuo didesnis už 1, dalinys didesnis, jei mažesnis – daliklis didesnis.

- A ir B → B didesnis, nes  $B/A = 26 > 1$  (A atmetamas)
- B ir C → C didesnis, nes  $C/B = 26/10 > 1$  (B atmetamas)
- C ir E → E didesnis, nes  $E/C = 100/26 > 1$  (C atmetamas)
- D ir E → E didesnis, nes  $E/D = 262/100 > 1$  (D atmetamas)

Tai informatika!

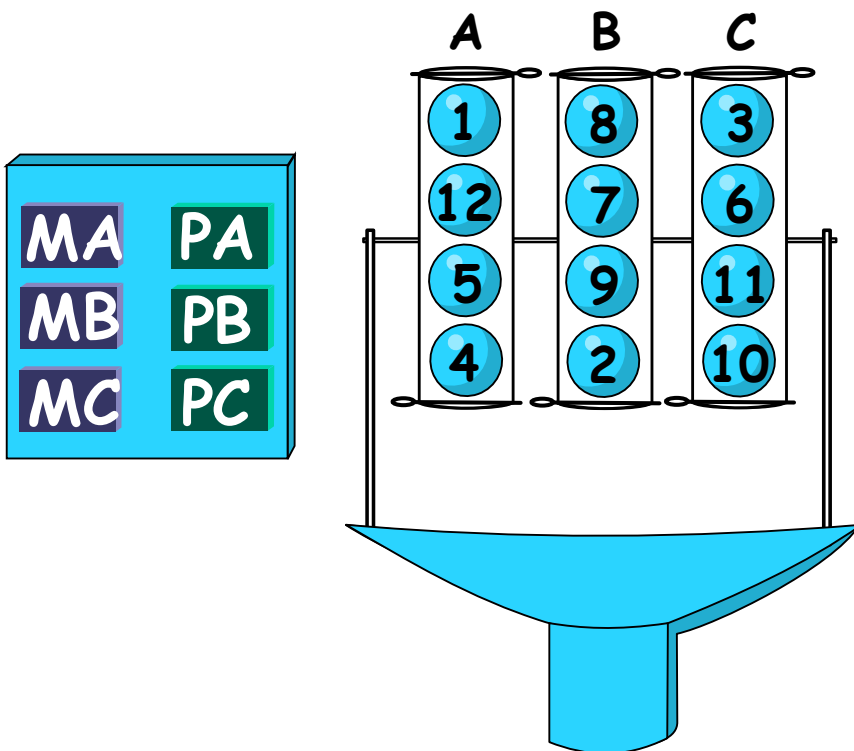
Užduotis reikalauja trupučio kombinatorikos ir matematinės nuojautos. Kombinatorikos elementas – galimų kombinacijų skaičius su tam tikru simbolių skaičiumi svarbus identifikatorių duomenų bazėse parinkimui ir pan. Taip pat tai svarbu įvertinant slaptažodžių saugumą.

Informatinis mąstymas

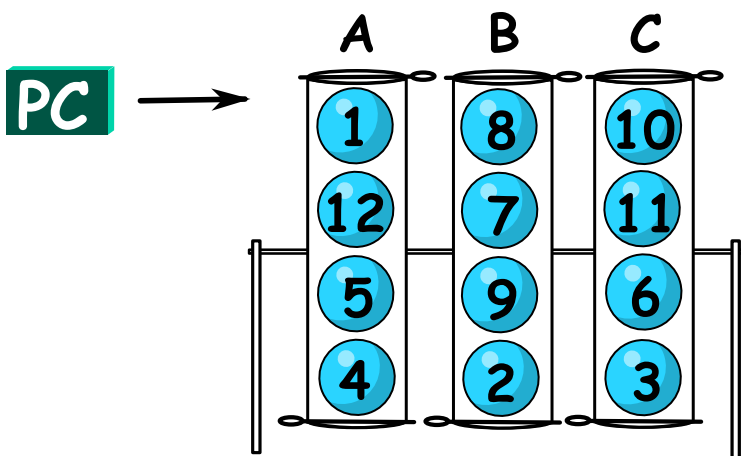
Užduoties paaiškiniame naudojamas truputį įdomesnis skaičių palyginimas juos dalijant vieną iš kito ir išsprastinant daliklius. Žymiai dažniau atliekamas skaičių palyginimas atimant. Tačiau efektyvinant skaičiavimus dažnai reikalingas greitesnis ir paprastesnis skaičiavimo būdas. Čia ši palyginimo idėja gali praversti. Taip pat svarbus sprendimas mažinti galimybes atmetimo būdu.

## 16. Naujas žaislas

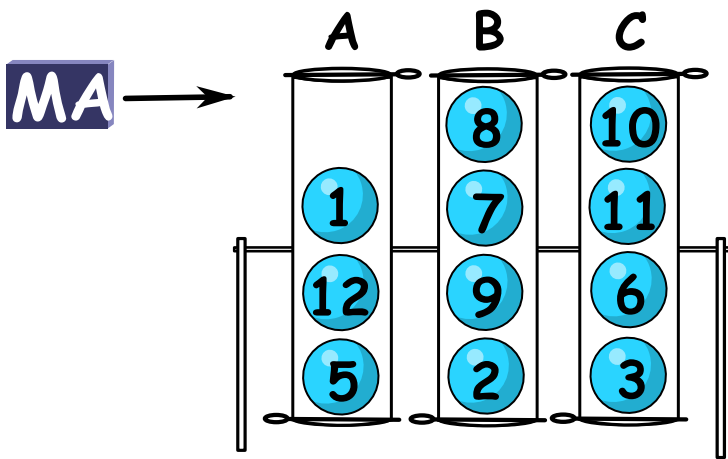
Jonas gimtadienio proga gavo dovaną – naują žaislą. Šis žaislas sudarytas iš trijų vamzdelių, kurių kiekviename yra po 4 kamuoliukus. Jonas gali paspausti vieną iš 6 mygtukų, esančių valdymo skydelyje. Mygtukai PA, PB ir PC pasuks atitinkamai A, B ir C vamzdelį 180 laipsnių (t. y., apvers žemyn). Mygtukai MA, MB ir MC mes vieną kamuoliuką iš atitinkamo vamzdelio į saugyklą.



Pavyzdžiui, paspaudus mygtuką PC, C vamzdelis apsivers:

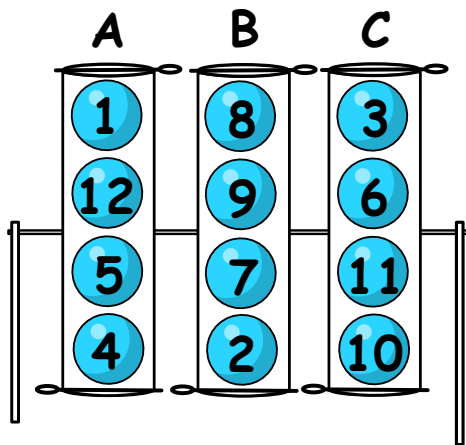


Tada paspaudus MA, kamuoliukas iškris iš vamzdelio A į saugyklą:

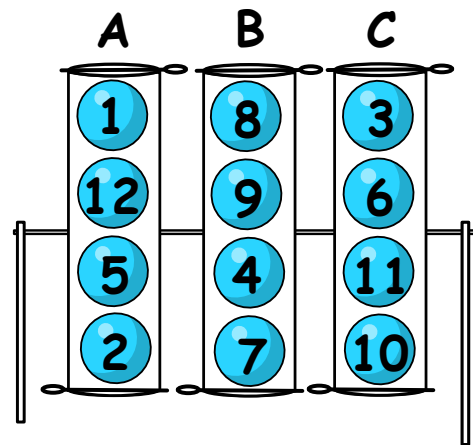


Jonas nori visus kamuoliukus perkelti iš vamzdelių į saugyklą kamuoliukų numerių didėjimo tvarka. Tačiau kartais kamuoliuko vieta vamzdelyje neleidžia to įgyvendinti.

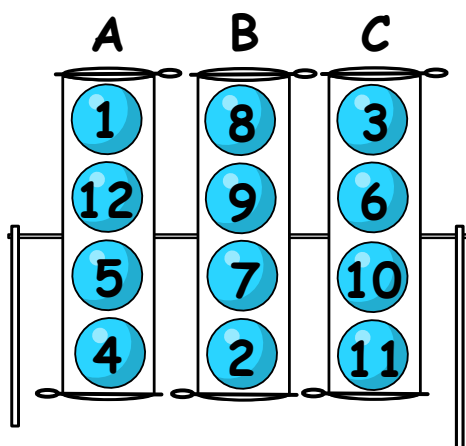
Kuriai iš šių situacijų esant Jonui to padaryti nepavyks?



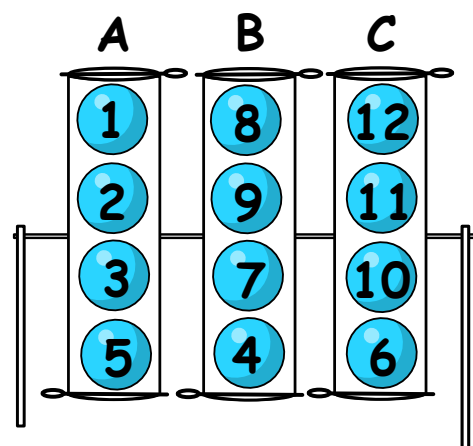
A.



B.



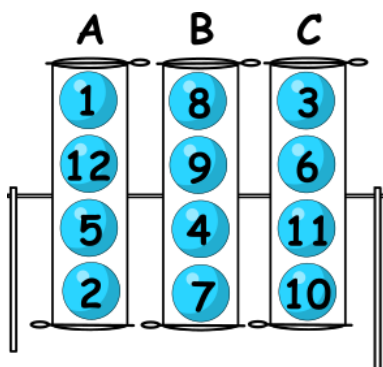
C.



D.

## Paaiškinimas

Teisingas atsakymas: B.



Žinome, kad teisinga kamuoliukų seka turėtų būti 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12. Tačiau nėra galimybės mesti į saugyklą 4-ą kamuoliuką prieš 7-ą ir 9-ą, kadangi 4-as kamuoliukas yra tarp kamuoliukų, pažymėtų numeriais 7 ir 9.

Visi kiti variantai tinka kamuoliukams surikiuoti.

Tam, kad Jonas kamuoliukus į saugyklą sumestų didėjimo tvarka, kamuoliukai kiekviename vamzdelyje nuo apačios iki viršaus turėtų būti išdėstyti vienu iš šių būdų:

- 1) didėjimo tvarka;
- 2) mažėjimo tvarka;
- 3) didėjimo tvarka, po kurios eina seka, surikiuota mažėjimo tvarka.

A variantas – neteisingas. Kamuoliukai A vamzdelyje išdėstyti didėjimo tvarka [4, 5, 12], po kurių eina seka mažėjimo tvarka [12, 1]. Kamuoliukai B vamzdelyje išdėstyti didėjimo tvarka [2, 7, 9], po jų – mažėjimo tvarka [9, 8]. C vamzdelyje kamuoliukai išdėstyti didėjimo tvarka [10, 11] ir mažėjimo tvarka [11, 6, 3].

C variantas – neteisingas. Kamuoliukai A vamzdelyje išdėstyti didėjimo tvarka [4, 5, 12], po kurių eina seka mažėjimo tvarka [12, 1]. Kamuoliukai B vamzdelyje išdėstyti didėjimo tvarka [2, 7, 9], po jų – mažėjimo tvarka [9, 8]. C vamzdelyje kamuoliukai išdėstyti didėjimo tvarka [11, 10, 6, 3].

D variantas – neteisingas. Kamuoliukai A vamzdelyje išdėstyti didėjimo tvarka [5, 3, 2, 1]. Kamuoliukai B vamzdelyje išdėstyti didėjimo tvarka [4, 7, 9], po jų – mažėjimo tvarka [9, 8]. C vamzdelyje kamuoliukai išdėstyti didėjimo tvarka [6, 10, 11, 12].

Tai informatika!

Ši užduotis priklauso rikiavimo uždavinių kategorijai. Rikiavimo algoritmai – vieni pagrindinių informatikos algoritmų. Rikiavimo algoritmų daug, vienas iš jų – sąlajinis rikiavimas. Sąlajinis rikiavimas yra efektyvus palyginimu grįstas rikiavimo algoritmas. Tai metodas, kai rikiuojamos sekos elementai skirstomi į grupes (rūšiuojami), o po to grupės paeiliui suliejamos į vieną seką. Procesas kartojamas keičiant skirstymo į grupes parametą, kol visa seka tampa surikiuota. Sekos elementai gali būti rekursyviai dalijami į dvi ar tris dalis – kaip šioje užduotyje, kai iš trijų vamzdelių išrenkamas mažiausias elementas ir perkeliamas į saugyklą. Tik skirtingai nuo sąlajinio rikiavimo, kamuoliukai vamzdeliuose yra nerikiuoti, todėl turime apversti vamzdelį.

Vamzdeliai yra abipusių eilių (angl. *double-ended queue*, *deque*) pavyzdžiai. Abipusė eilė yra abstraktusis duomenų tipas, leidžiantis įterpti ir šalinti elementus iš abiejų eilės galų.

Vamzdeliuose esančius kamuoliukus galime laikyti programos įvesties duomenimis. Pateiktos konfigūracijos atitinka tris bandymų pavyzdžius, kuriems mūsų programa pateikia sprendinius, ir vieną bandymų pavyzdį, kuriam sprendinio nėra.

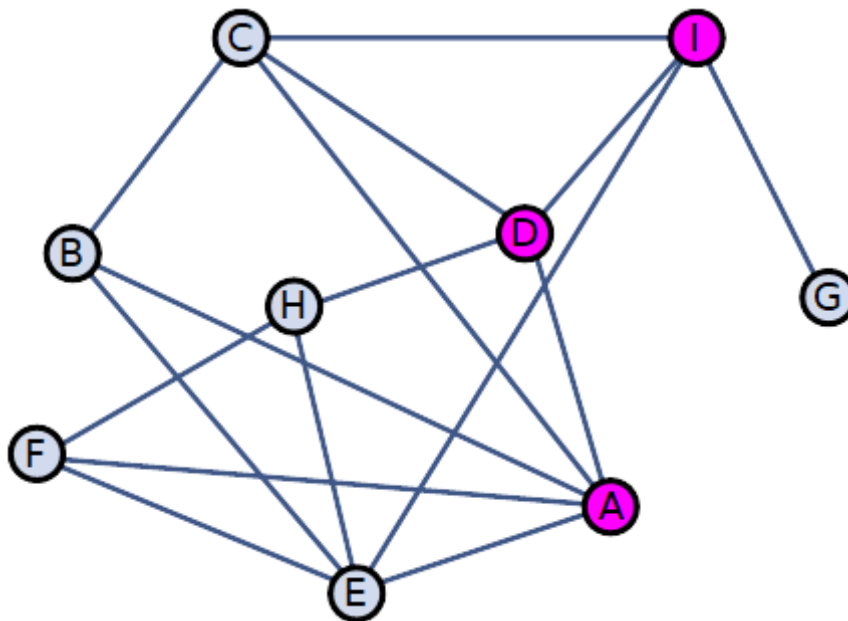
Informatinis mąstymas

Algoritmai. Sprendžiant šią užduotį reikia suprasti, kaip veikia algoritmas. Sudarome operacijų seką arba programą (šioje užduotyje – posūkis ir metimas), kad pasiektume tikslą (kamuoliukai saugykloje būtų surikiuoti).

Abstrakcija. Vamzdeliams vaizduoti galime naudoti masyvą arba abipusę eilę ir patikrinti, ar jie tinka pradiniais duomenimis mūsų programai pateikti. Sukuriamos bendrosios sąlygos, kad patikrintume, ar kamuoliukai vamzdeliuose gali būti išdėstyti didėjimo tvarka.

## 17. Ponia Begalybė

Mokiniai su bendraklasiais mokykloje kalbasi tokia tvarka, kaip parodyta schemeje. Pavyzdžiui, mokinys H per dieną kalbasi tik su D, E ir F. Pirmadienį į klasę atėjo nauja matematikos mokytoja. Mokiniai A, D ir I iš karto mokytoją pradėjo vadinti „Ponia Begalybė“ dėl jos šukuosenos. Pravardė tarp mokinių plinta tokiu būdu: jei daugiau nei pusė bendraklasių, su kuriais mokinys bendrauja, pasako šią pravardę, tai šis mokinys ją vartos ir kitą dieną.



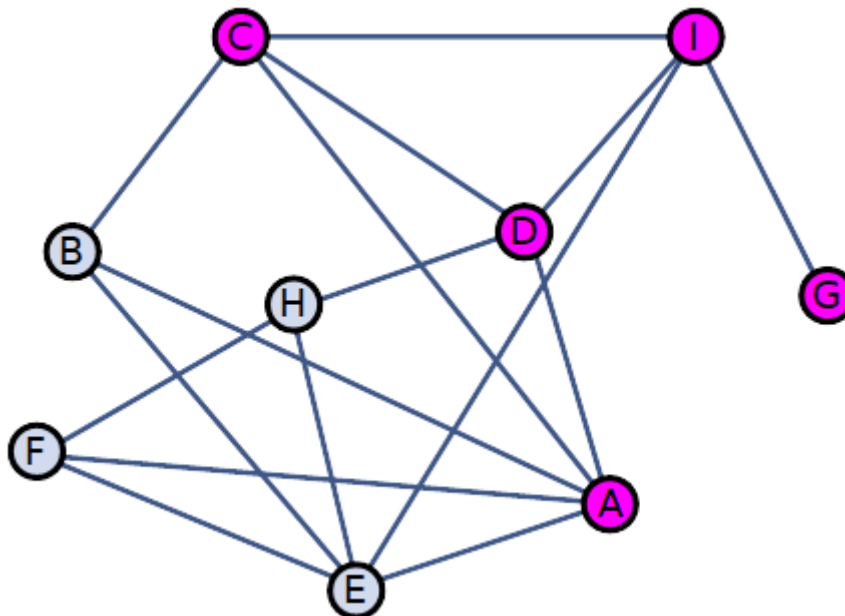
Kokia bus pirmoji savaitės diena, kai kiekvienas klasės mokinys vartos pravardę „Ponia Begalybė“?

- A. Antradienis
- B. Trečiadienis
- C. Ketvirtadienis
- D. Penktadienis

## Paaiškinimas

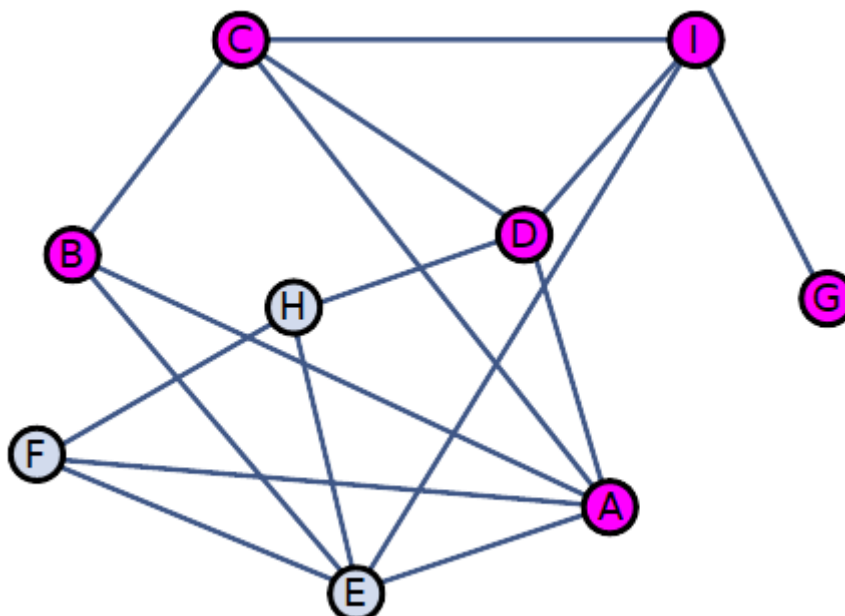
Teisingas atsakymas: penktadienis.

Galime pasekti, kaip pravarde plinta. Mokiniai G ir C pradeda naudoti pravarde antradienį.

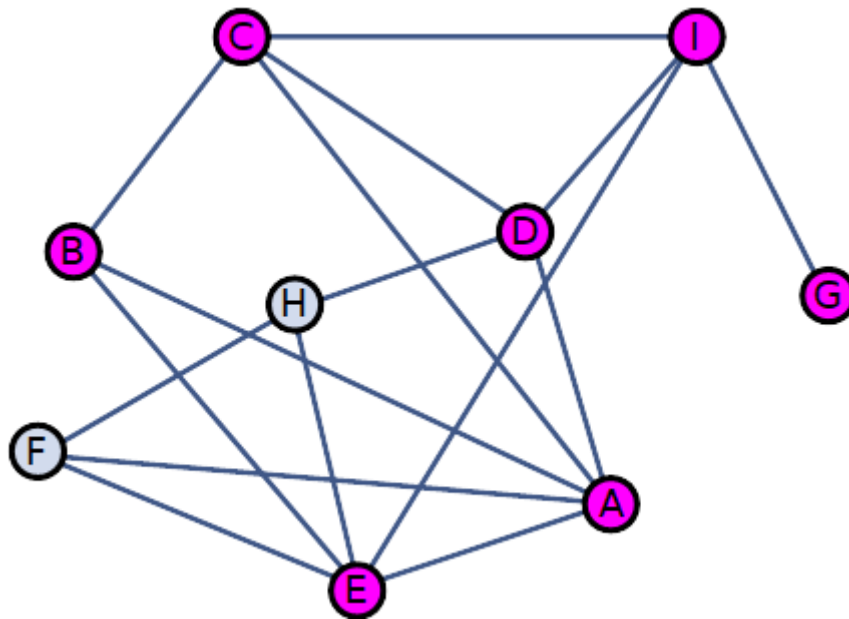


G bendrauja tik su vienu bendraklasiu (I) ir šis bendraklasis pravarde naudojo pirmadienį. C bendrauja su 4 bendraklasiais ir 3 iš jų pravarde naudojo pirmadienį. 3 iš 4 bendraklasių yra daugiau nei pusė. Taigi C taip pat pravarde naudojo antradienį.

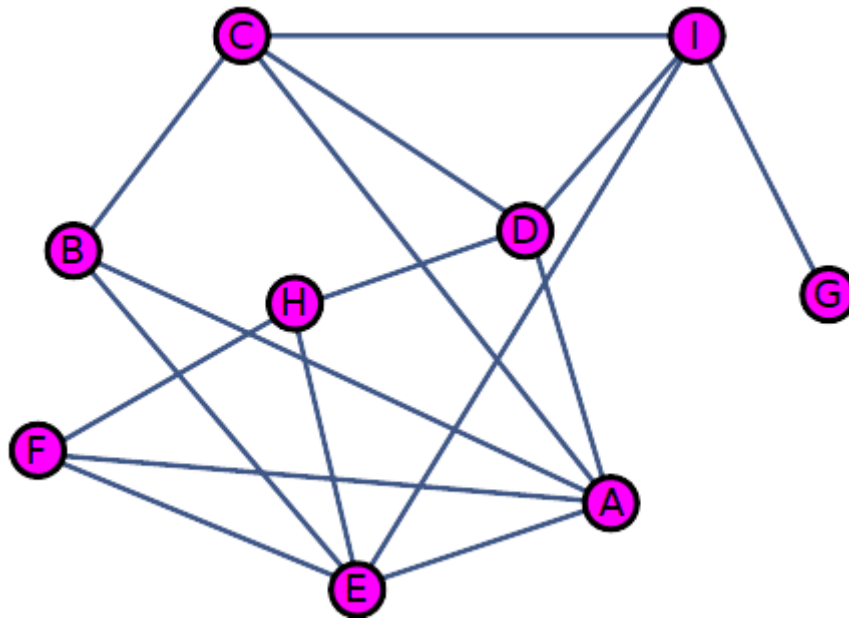
B pradeda naudoti pravarde trečiadienį.



E pradeda naudoti pravarde ketvirtadienį.



F ir G pradeda naudoti pravarde penktadienį.



Tai informatika!

Socialiniai tinklai atlieka svarbų vaidmenį skleidžiant informaciją apie infekcijas, idėjas, daro įtaką savo nariams. Socialiniame tinkle atsiradusi idėja ar naujovė greitai išplinta per socialines sąveikas arba išnyksta. Įtakos sklaidos socialiniuose tinkluose maksimizavimas aktualus informatikos, ekonomikos ir vadybos mokslams. Šioje užduotyje nagrinėjamas specialus sklaidos modelis, vadinamas slenkstiniu modeliu. Jame kiekvienam asmeniui yra nustatytas slenkstis, t. y., dalis jo ryšių, kurie turi būti aktyvūs, kad asmuo taptų aktyvus.

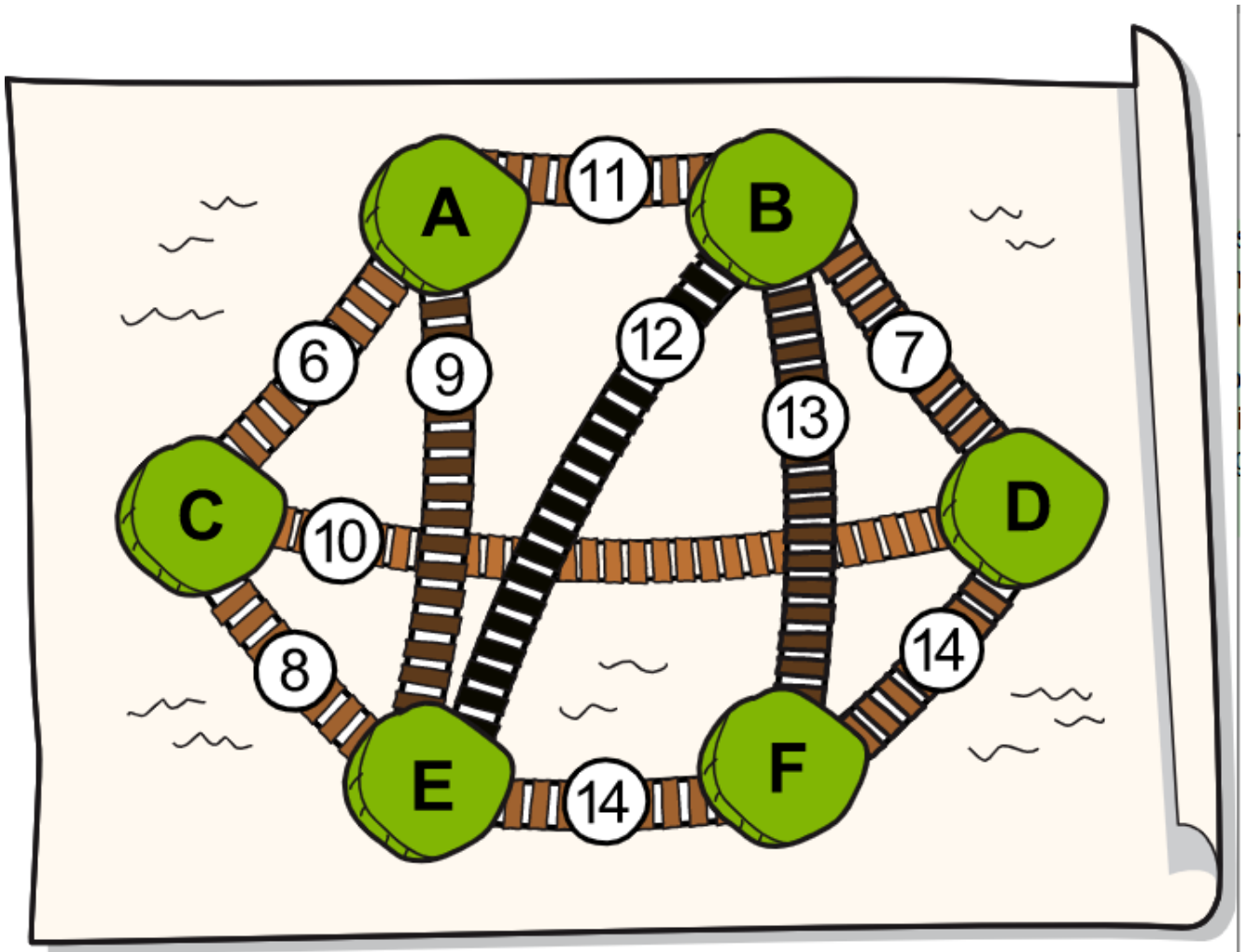


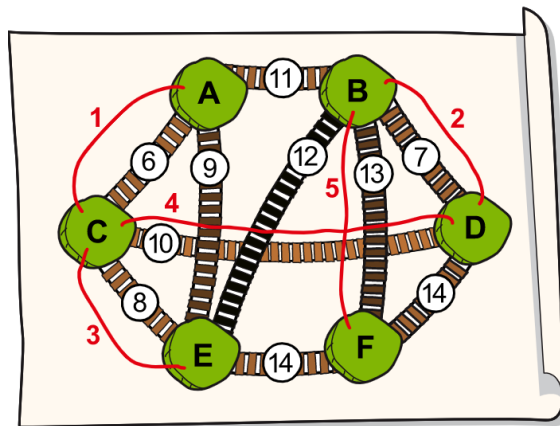
## 18. Salų jungimas

Šešiose salose gyvenanti džiunglių bendruomenė nori sujungti šias salas sukurdami skliautinių tiltų tinklą. Ji sudarė salų jungimo tiltais planą. Tiltai nesusikerta vienas su kitu. Skaičiai rodo tiltų statybos išlaidas.

Bendruomenė nori sujungti visas salas taip, kad iš bet kurios salos būtų galima keliauti į bet kurią kitą salą tiesiogiai arba netiesiogiai per vieną ar daugiau salų. Taip pat bendruomenė nori pastatyti tiltus kuo pigiau.

Koks yra pigiausias būdas sujungti visas šešias salas?





$$6 + 7 + 8 + 10 + 13 = 44$$

Algoritmas yra paprastas ir intuityvus (tai Kruskalo algoritmas):

1. Pradėkime nuo pigiausio tilto AC, kurio kaina 6.
2. Tada renkamės antrą pigiausią tiltą – BD (7).
3. Trečias pigiausias tiltas – CE (8).
4. Kito pigiausio tilto kaina 9. Tačiau jei jį pastatysime, gausime jungtį AECA. Vadinasi, šis tiltas mums visai nereikalingas – salas A, C ir E galime pasiekti nstatydami tilto AE.
5. Ketvirtas statomas tiltas yra CD (10).
6. Kitas tiltas AB, kurio kaina 11, vėl yra jungtis: ABDCA.
7. Panašiai ir tiltas BE (12) sudaro jungtį ACDBEA.
8. Penktas tiltas, kurį turime pastatyti, yra BF (13).

Dabar turime sujungtas visas šešias salas, o penkis jungiamuosius tiltus statome pigiausiu būdu: kiekviename žingsnyje pasirinkome pigiausią įmanomą tiltą.

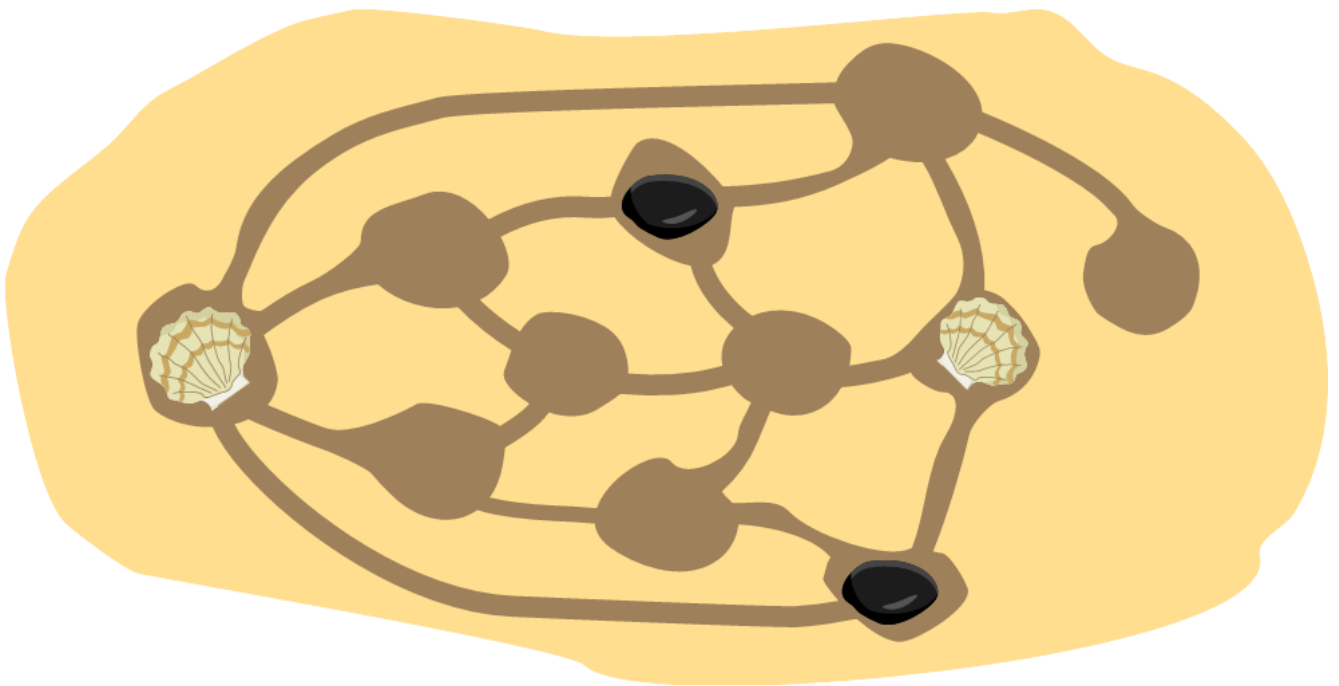
Tai informatika!

Sprendžiant šią užduotį reikia rasti grafo minimalų jungiamąjį medį. Minimalus jungiamasis medis – tai duoto grafo pografis, turintis lygiai tas pačias viršūnes ir kai kurias briaunas, taigi jis irgi yra medis. Kruskalo algoritmas visada randa minimalų jungiamąjį medį svertiniam grafiui. Algoritmas grindžiamas mažiausių svorį turinčių briaunų pasirinkimu po vieną, vengiant jungčių su jau pasirinktomis briaunomis.

## 19. Kriauklės ir akmenukai

Ana ir Bobas žaidžia paplūdimyje. Ana surinko daug šviesių kriauklių, Bobas – daug tamsių akmenukų. Vaikai smėlyje padaro duobutes ir sujungia jas grioveliais. Jie paeiliui į pasirinktą tuščią duobutę įdeda po vieną savo kriauklę ar akmenuką. Ana pradeda žaidimą.

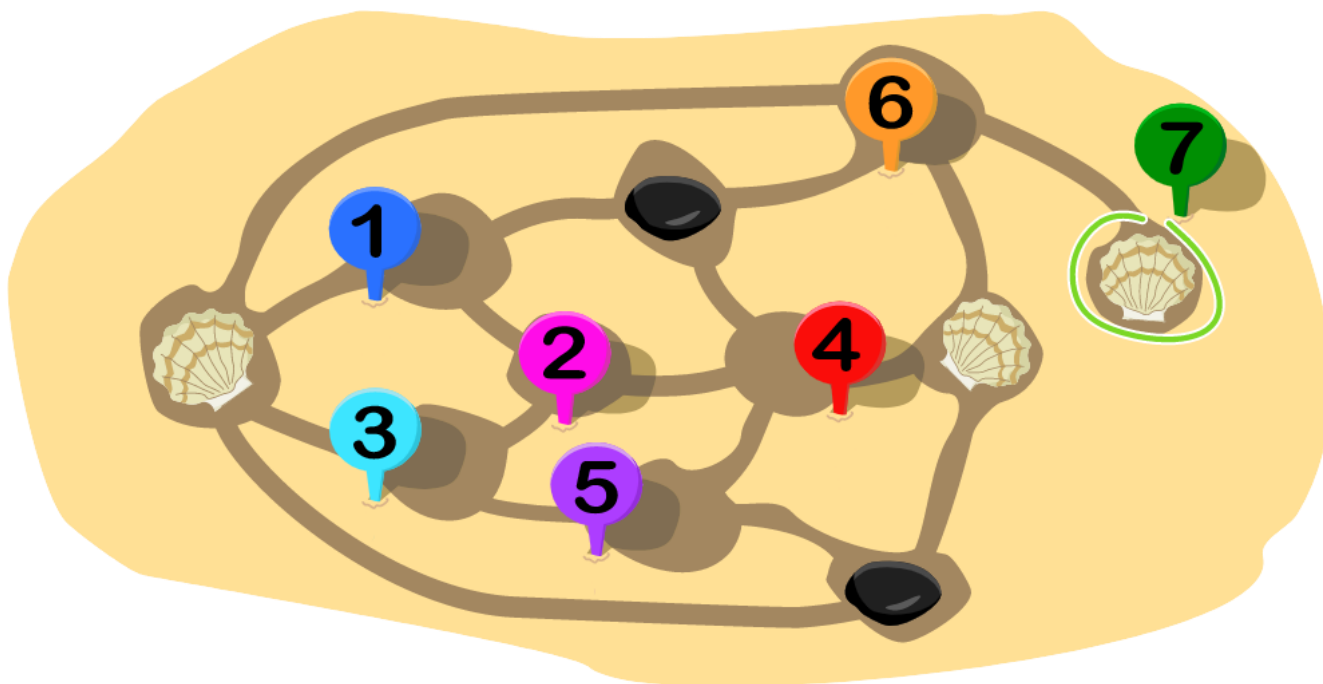
Pralaimi tas žaidėjas, kuris pirmas įdeda dvi savo figūreles į dvi duobutes, sujungtas vienu griovelium.



Dabartinė žaidimo situacija pavaizduota paveikslėlyje. Žaidimą tęsia Ana. Į kurią tuščią duobutę ji turi įdėti kriauklę, kad užsitikintų pergalę?

## Paaiškinimas

Teisingas atsakymas: 7-a duobutė.



1, 3, 4 ir 6 duobutės Anai yra netinkamos, o 1, 4, 5 ir 6 – Bobui. Į 7-ą duobutę Anai įdėjus kriauklę, Bobas gali pasirinkti 2 arba 3 duobutę. Abiem šiais atvejais Ana įdeda kriauklę į 5 duobutę ir Bobas pralaimi žaidimą.

Jei Ana pasirinktų 2-ą duobutę, tai Bobui liktų 7 duobutė. Toliau Anai tiktų tik 5 duobutė, o Bobui tiktų 3 duobutė ir Ana pralaimėtų.

Jei Ana pasirinktų 5-ą duobutę, o Bobas – 7, po to Ana – 2, Bobas – 3 ir Ana pralaimėtų.

Jei paveikslėlyje pavaizduotą situaciją tęstų Bobas, tai jis vis tiek pralaimėtų. Vienu atveju Bobas gali dėti akmenuką į 2-ą ar 3-ą duobutę, o Ana tuomet padėtų kriauklę į 7 duobutę. Antru atveju Bobas akmenuką padėtų į 7-ą duobutę, tada Ana dėtų į 2-ą, toliau Bobui liktų 3 duobutė, o Ana pasirinktų 5-ą duobutę ir Bobas pralaimėtų.

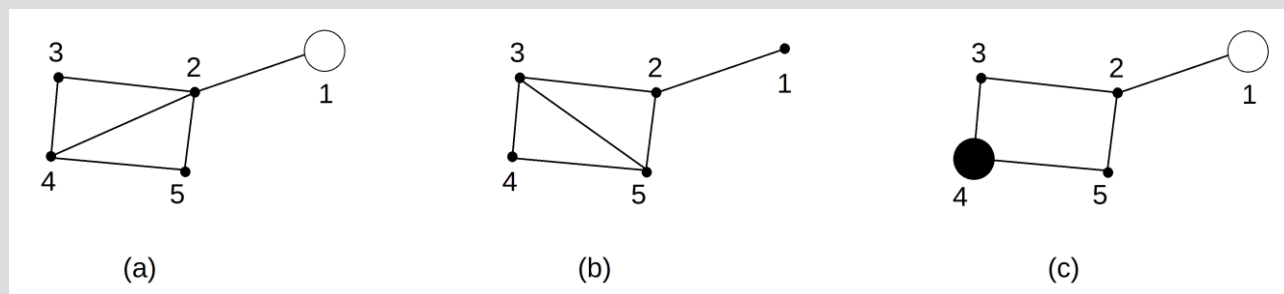
### Tai informatika!

1912 m. Ernsto Zermelo suformuluota teorema taikoma poziciniams baigtiniams žaidimams: jei abu žaidėjai daro viską, ką gali (jie yra tobuli žaidėjai), žaidimas baigiasi lygiosiomis arba vienas iš dviejų žaidėjų turi turėti laiminčią strategiją, t. y., gali iškovoti pergalę. Pavyzdžiui, pikarija (angl. *Picaria*) nėra baigtinis žaidimas (jei nėra atitinkamų apribojimų, žaidimas gali tęstis amžinai, jei nė vienas žaidėjas nesuklysta), o kryžiučių-nuliukų žaidimas (angl. *tic-tac-toe*) visada baigiasi lygiosiomis, jei abu žaidėjai žaidžia

geriausiai. Kalbant apie spalvinimo žaidimus *Col* ir *Snort* (neabejotinai baigtinius), rezultatas priklausys nuo žaidimo lentos, t. y., nuo pateikto grafo, kurio viršūnės yra duobutės, o briaunos – grioveliai (tarp skirtingų duobučių).

Visuose minėtuose žaidimuose lentą galima modeliuoti paprastu ir neorientuotu (taip pat baigtiniu, jungtiniu, plokščiu) grafu, kuriame figūrėlės užima viršūnes. Apskritai matematikoje ir informatikoje paprastas ir neorientuotas grafas yra struktūra, tinkama aibės elementų simetriniam santykiui vaizduoti.

Pateiktame paveikslėlyje pavaizduotos trys galimos baigtys, naudojant tris paprastas skirtingas lentas: a) pirmasis žaidėjas (baltieji) turi laiminčią strategiją ir privalo pradėti 1 (kitai pralaimės); b) antrasis žaidėjas (juodieji) turi laiminčią strategiją; c) partija baigsis lygiosiomis, jeigu baltieji pradeda 1 (kitai pralaimės), o juodieji atsako 4 (kitai pralaimės): pirmieji du ėjimai yra priverstiniai! (Beje, *Snort* naudojant šias tris lentas, baltieji laimėtų pradėdami 2.)



*Col* žaidimo tam tikrose situacijose, pavyzdžiui, simetrinėse pozicijose, yra laimėjimo strategijų.

## 20. Rikiavimas

Jorūnas turi 11 kortelių su skaičiais nuo 1 iki 9 ir raidėmis A, B. Jis naudoja taisyklę  $9 < A < B$ . Jorūnas nori korteles sudėti į vieną eilutę, vadovaudamasis  $a \rightarrow b$  tipo taisykle. Ši taisyklė reiškia, kad kortelė su numeriu  $a$  turi būti kažkur kairėje nuo kortelės su numeriu  $b$ . Kartu jis nori, kad kortelių išdėstymas sudarytų kuo mažesnę leksikografinę seką.

Leksikografiškai mažesnė iš dviejų sekų yra ta, kurios pirmoje (iš kairės) skirtingų kortų poroje yra mažesnis skaičius. Panagrinėkime šias dvi sekas:

1) 5 6 5 7 9 9 B

2) 5 6 5 8 0 0 0

Pirmoji seka leksikografiškai mažesnė, nes pirmosios trys atitinkamų kortelių poros yra vienodos, o ketvirtoje pozicijoje pirmoji seka turi 7, o antroji – 8 ( $7 < 8$ ).

Išdėstykite 11 kortelių leksikografiškai mažiausia seka pagal šias taisykles:

1)  $1 \rightarrow 2$

2)  $1 \rightarrow 6$

3)  $2 \rightarrow 3$

4)  $6 \rightarrow 5$

5)  $3 \rightarrow 5$

6)  $3 \rightarrow 4$

7)  $5 \rightarrow 7$

8)  $7 \rightarrow 8$

9)  $7 \rightarrow 9$

10)  $1 \rightarrow B$

11)  $B \rightarrow A$

12)  $A \rightarrow 7$

1	2	3	4	5	6	7	8	9	A	B
---	---	---	---	---	---	---	---	---	---	---

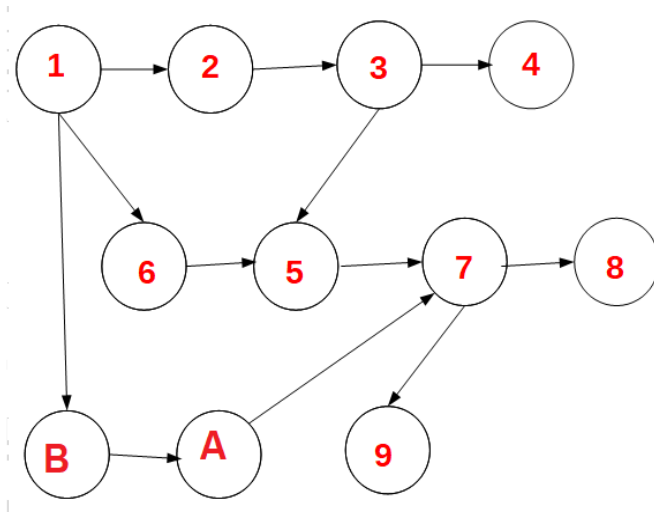
--	--	--	--	--	--	--	--	--	--	--

## Paaiškinimas

Teisingas atsakymas: 1 2 3 4 6 5 B A 7 8 9.

1 6 2 3 4 5 B A 7 8 9 taip pat tenkina visas nurodytas sąlygas, tačiau gaunama seka leksikografiškai didesnė.

Šią užduotį suformuluokime grafų teorijos terminais. Kiekvieną kortelę laikysime orientuoto grafo viršūne, o kiekvieną sąlygą – orientuota briauna, nukreipta iš kairės į dešinę kortelę (žr. paveikslą):



Sudarykite į kiekvieną viršūnę įeinančių briaunų sąrašus:

- 1: -
- 2: 1
- 3: 2
- 4: 3
- 5: 3, 6
- 6: 1
- 7: 5, A
- 8: 7
- 9: 7
- A: B
- B: 1

(1-a viršūnė neturi įeinančių briaunų, o į 5-ą viršūnę eina briaunos iš 3-os ir 6-os viršūnių.)

Leksikografiškai mažiausia seka bus sudaryta taip:

- Kiekviename žingsnyje viršūnėms, kurios nebeturi įeinančių briaunų, parinksime briauną su mažiausiu skaičiumi.

- Ši viršūnė ir iš jos išeinančios briaunos bus pašalintos iš įeinančių briaunų sąrašo.

Pirmiausia pasirinkite viršūnę 1.

Dabartinė paieškos seka: [1, ...]

Atnaujinti įvesties briaunų sąrašai:

1: -

2: -

3: 2

4: 3

5: 3, 6

6: -

7: 5, A

8: 7

9: 7

A: B

B: -

Raudonai pažymėkite viršūnes, kurios jau įtrauktos į norimą seką.

Dabar 2, 6, 11 viršūnės neturi įeinančių briaunų.

Pasirinkite 2 viršūnę (mažiausias skaičius).

Dabartinė paieškos seka: [1, 2, ...]

Atnaujinti įvesties briaunų sąrašai:

1: -

2: -

3: -

4: 3

5: 3, 6

6: -

7: 5, A

8: 7

9: 7

A: B

B: -

Atlikdami kitus tris veiksmus, nuosekliai pridėkite viršūnes 3, 4, 6 dabartine norima seka:

Dabartinė paieškos seka: [1, 2, 3, 4, 6,...]



Atnaujinti įvesties briaunų sąrašai:

1: -

2: -

3: -

4: -

5: -

6: -

7: 5, A

8: 7

9: 7

A: B

B: -

Atlikdami tolesnius tris žingsnius nuosekliai pridėkite viršūnes 5, B, A dabartine norima seka.

Dabartinė paieškos seka: [1, 2, 3, 4, 6, 5, B, A, ... ]

Atnaujinti įvesties briaunų sąrašai:

1: -

2: -

3: -

4: -

5: -

6: -

7: -

8: 7

9: 7

10: -

11: -

Paskutinės trys viršūnės: 7, 8, 9. Mes jas išdėstome didėjimo tvarka:

[1, 2, 3, 4, 6, 5, B, A, 7, 8, 9]

Tai informatika!

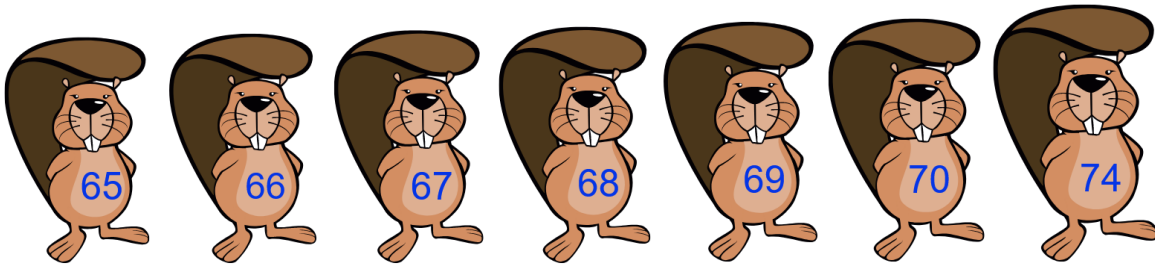
Algoritmai, skirti elementams išdėstyti tvarkinga seka, vadinami rikiavimo algoritmais. Tai yra viena iš klasikinių algoritmų teorijos sričių.

Paprasčiausias rikiavimo būdas yra visų galimų variantų paieška ir tikrinimas – vadinamasis jėgos metodas (angl. *brute force method*). Jo pakanka sekoms, turinčioms nedidelį elementų skaičių.

Kai esama daug kortelių ir nustatytų sąlygų, šis metodas yra labai sudėtingas ir reikalauja daug kartų tikrinti tas pačias sąlygas. Tuo pačiu reikia būti labai atsargiems, kad nepraleistume leksikografiškai mažesnio sprendinio.

Todėl toks uždavinys daug efektyviau sprendžiamas topologinio rikiavimo metodu.

## 21. Bebrų dirbtinis intelektas



Bebrai konstruoja dirbtinio intelekto sistemą. Ši sistema skirta matuoti gyvūno dydį ir tuo remiantis nuspręsti, ar tas gyvūnas yra bebras, ar ne. Dirbtinio intelekto sistema mokosi priimti sprendimus iš jai pateikiamų pavyzdžių. Pirmiausia sistema mokosi iš tokių dydžių gyvūnų pavyzdžių:

- 65, 66, 67, 68, 69 => bebras
- 11, 101, 110, 120, 130 => nėra bebras

Tada bebrai leidžia dirbtinio intelekto sistemai spręsti, ar tam tikri gyvūnai yra bebrai. Sistema pateikia tokį atsakymą:

- 70, 74 => bebras
- 86, 38 => nėra bebras
- 40, 80 => bebras

Dirbtinis intelektas padarė klaidą, kadangi 40 ir 80 dydžio gyvūnai nėra bebrai!

Dirbtinis intelektas aptiko, kad gyvūnas, kurio dydis 11, nėra bebras, o gyvūnas, kurio dydis 65, yra bebras. Atsižvelgusi į gyvūnų dydžių skirtumus, dirbtinio intelekto sistema padarė išvadą, kad bebrai yra tik tie gyvūnai, kurie didesni už 38 ir mažesni kaip 85.

Dirbtinio intelekto sistemai pagerinti bebrai pateikė jai dar vieną pavyzdį: gyvūnas, kurio dydis 42, nėra bebras.

Kaip dabar dirbtinio intelekto sistema klasifikuos du gyvūnus, kurių dydžiai 48 ir 84?

- A. bebras, bebras
- B. bebras, nėra bebras
- C. nėra bebras, bebras
- D. nėra bebras, nėra bebras

## Paaiškinimas

Teisingas atsakymas: C.

Kaip dirbtinio intelekto (DI) sistema pasirenka dydžių intervalą 38–85 bebrams identifikuoti? Iš tiesų, tiksliai nežinome. Kadangi tarp DI nagrinėtų pavyzdžių yra „11 – ne bebras“, ir „65 – bebras“, DI paskaičiavo ribas, kada gyvūnas yra bebras, kada – ne, ir nustatė jas tarp 11 ir 65. Tinkamas taškas ribai gali būti vidurkis  $(11+65)/2 = 38$ . Panaši taisyklė gali būti taikoma kitam intervalo galui rasti:  $(69+101)/2 = 85$ . Naudojant tą pačią procedūrą, pamačius pavyzdį „42 nėra bebras“, riba 85 neturėtų pasikeisti, bet nauja kairioji riba turėtų būti  $(42+65)/2 = 53,5$ . Taigi gyvūnas, kurio dydis 48, būtų identifikuotas, jog „nėra bebras“, o 84 dydžio gyvūnas būtų laikomas bebru.

Tai informatika!

Šios užduoties dirbtinio intelekto sistemoje taikomas mašininio mokymosi algoritmas. Mašininio mokymosi srityje informatikai tyrinėja kompiuterių algoritmus, kurie galėtų automatiškai tobulinti savo sprendimų priėmimo procesą. Mašininio mokymosi algoritmai formuoja modelį remdamiesi pavyzdžių duomenimis, vadinamais apmokymo duomenimis, tam, kad prognozuotų arba priimtų sprendimus be tiesioginio nurodymo programoje tai daryti. Tokių algoritmų kokybė labai priklauso nuo naudotų apmokymo duomenų kokybės, – tai galime matyti šioje užduotyje.

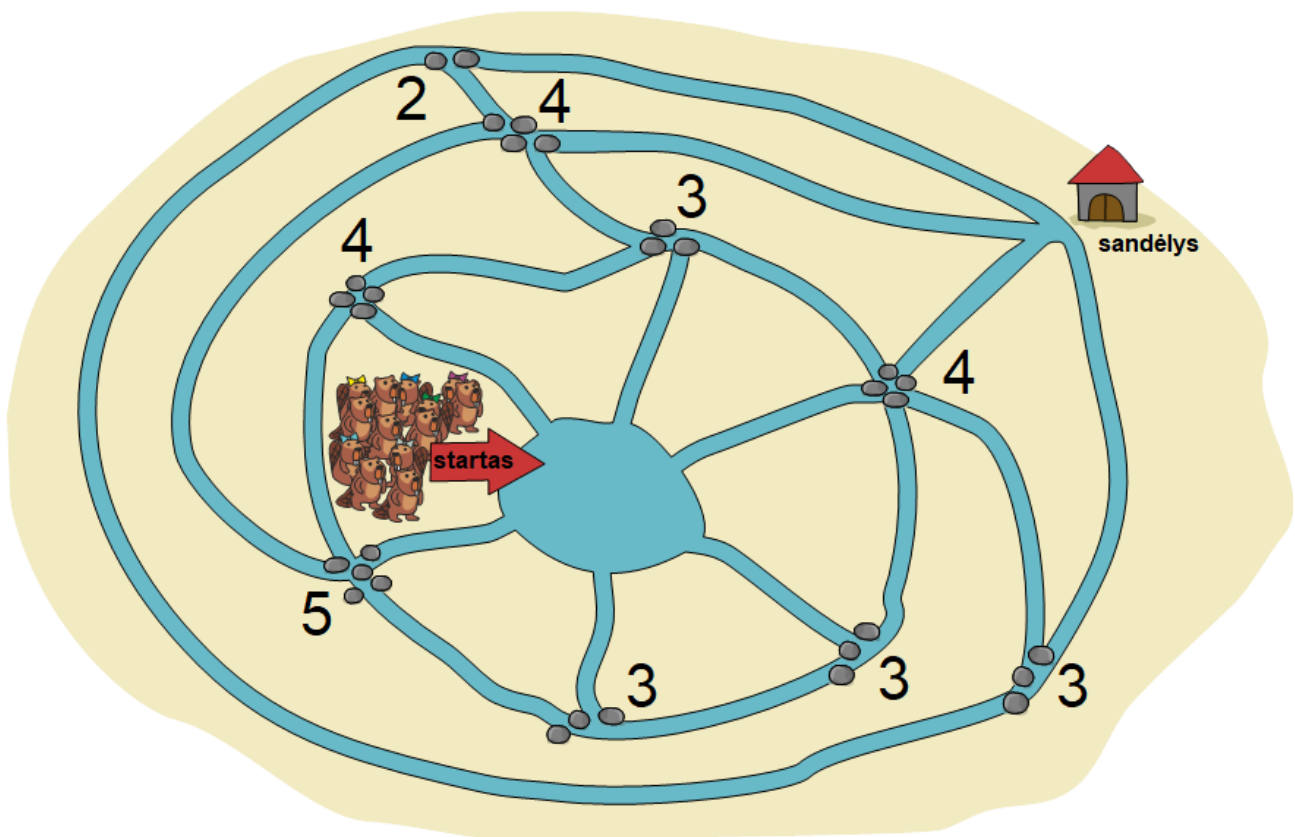
Mašininio mokymosi algoritmų taikymo spektras labai didelis. Jie naudojami medicinoje, el. pašto laiškams filtruoti, šnekos atpažinimui, kompiuterinei regai, – tada, kai sudėtinga arba netikslinga projektuoti tradicinius algoritmus numatytiems uždaviniams atlikti.

Mašininis mokymasis laikomas priklausančiu dirbtinio intelekto sričiai. Mašininio mokymosi veikimo principą galite išbandyti interneto svetainėse, pvz.,

<https://machinelearningforkids.co.uk> arba „Dirbtinis intelektas vandenynui“ ([AI for Oceans](#)).

## 22. Akmenys

Bebrų šeimos sodyboje yra iš 21 kanalo susidedanti vandens sistema. Kai kurių kanalų sankirtose yra akmenų. Iš viso yra 31 akmuo ir juos visus bebrai, plaukiodami kanalais, turi sunešti į sandėlį.



Akmenys sunkūs, todėl vienu metu bebras gali plukdyti tik vieną ar du akmenis, bet ne daugiau. Kad nukeliautų nuo vienos kanalų sankirtos iki kitos, bebrai visada plaukia lygiai vieną valandą. Pradėję vienu metu bebrai turi per keturias valandas sunešti į sandėlį visus akmenis.

Koks mažiausias bebrų skaičius gali šį darbą atlikti?

- 14
- 18
- 20
- 24

## Paaiškinimas

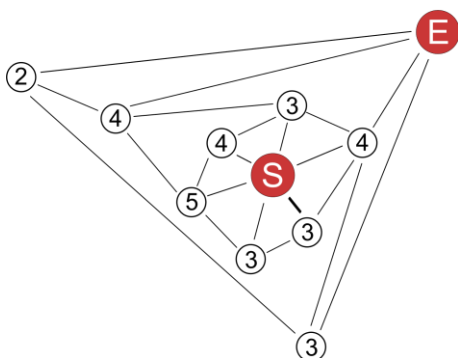
Teisingas atsakymas: 14 bebrų.

### Trumpas paaiškinimas

Yra tik vienas pasirinkimas, kai pradėję plaukti iš starto vietos bebrai pasiekia sandėlį per 2 valandas. Šiuo atveju kanalų sankirtoje yra 4 akmenys. Šiuos 4 akmenis iki sandėlio gali nunešti du bebrai, kurie gali grįžti, ir vėl nunešti dar 4 akmenis iki sandėlio. Lieka 23 (31–8) akmenys ir reikia dar mažiausiai 12 bebrų, nes likusiems akmenims nunešti reikia 3 arba 4 valandų, tad nebelieka laiko atnešti daugiau akmenų. Likusius 23 akmenis į sandėlį gali atnešti 12 bebrų, iš kurių 11 neša po 2 akmenis, o vienas – tik vieną akmenį.

### Detalus paaiškinimas

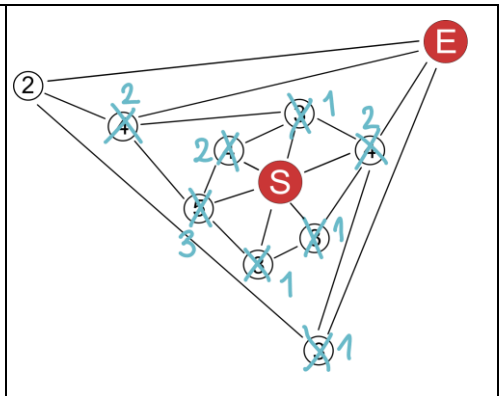
Iš pradžių visi akmenys yra kanalų sankirtose. Vienoje sankirtoje yra ir sandėlis (E). Paveikslėliuose kanalai vaizduojami tiesiomis linijomis, o skaičiai reiškia akmenų skaičių atitinkamoje sankirtoje.

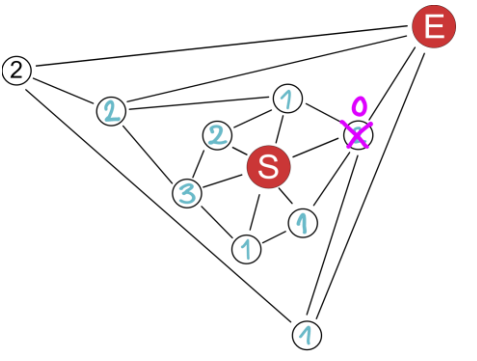
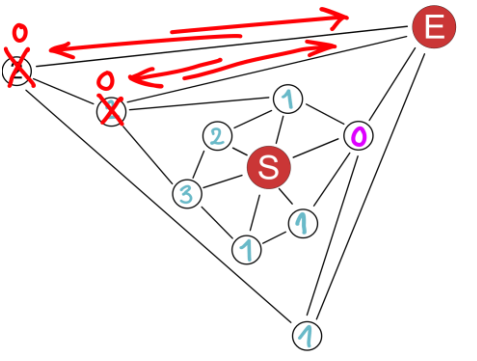
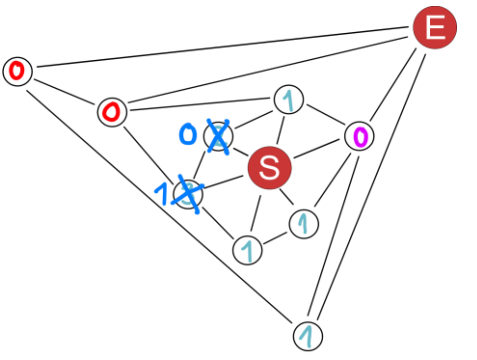
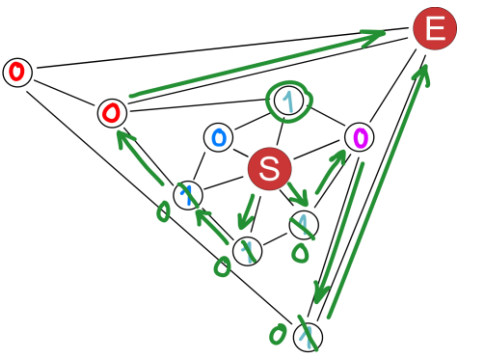


Kelio trukmę tarp pradinio taško ir sandėlio matuojame skaičiuodami kanalus, kuriais bebrai turi plaukti. Nėra atvejų, kai vienas kanalas jungia startą ir sandėlį (t. y., nėra 1 valandos trukmės kelių). Yra 2 valandų, 3 valandų ir 4 valandų trukmės kanalais keliai. Bebrai akmenis turi rinkti ne ilgiau kaip keturias valandas, tad ilgesnių nei keturių valandų trukmės kelių nagrinėti nereikia.

Pirmiausiai nagrinėjame visas kanalų sankirtas, kuriose yra bent trys akmenys. Kiekvienoje tokioje sankirtoje bebras paima du akmenis ir atneša juos į saugyklą. Tam reikia 8 bebrų. Jie surenka 16 akmenų. Dar lieka  $31 - 16 = 15$  akmenų. Visi bebrai, išskyrus vieną, užtrunka kelyje mažiausiai tris valandas. Jie negali grįžti prie kanalų ir paimti daugiau akmenų (t. y., padirbėti pakartotinai).

Vienas bebras per dvi valandas surenka du akmenis. Jis turi pakankamai laiko grįžti iki sankirtos, paimti dar



<p>du akmenis ir nunešti juos į sandėlj. Todėl šis bebras gali padirbėti pakartotinai.</p>	
<p>Kitas bebras per dvi valandas surenka taip pat du akmenis. Šiuo bebru taip pat galima pasinaudoti pakartotinai. Dar lieka <math>15-2 = 13</math> akmenų. Jau pasinaudojome 9 bebrais. Du iš jų dar gali padirbėti pakartotinai, nes jie sugaišo tik po dvi valandas.</p>	
<p>Jau dviem bebrais iš pasirinktų 9 bebrų pasinaudojome pakartotinai, kiekvienas iš jų atnešė po du akmenis. Lieka <math>13-4 = 9</math> akmenys. Tačiau jau atidirbusių bebrų skaičius toks pat – 9. Nė vienu iš jų jau nebegalime pasinaudoti.</p>	
<p>Toliau taikoma ta pati strategija ir nagrinėjamos visos kanalų sankirtos, kuriose yra bent du akmenys. Reikalingi du bebrai. Jie surenka 4 akmenis. Dar lieka <math>9-4 = 5</math> akmenys. Jau pasinaudota <math>9+2 = 11</math> bebrų.</p>	
<p>Galutinis žingsnis – rasti sprendimą (galbūt bandymų ir klaidų metodu), kuris leistų naudoti kuo mažiau bebrų. Dar du bebrai surenka po du akmenis. O paskutinis bebras paima paskutinį akmenį. Geriausias sprendimas yra A: <math>11+3 = 14</math> bebrų.</p>	

Yra 31 akmuo, tik du bebrai gali atnešti po 2 akmenis ir grįžę pasiimti daugiau, kiti bebrai gali atnešti ne daugiau kaip 2 akmenis, todėl reikia mažiausiai 13,5 bebro. Taigi geriausiam sprendimui reikia 14 bebrų.

Tai informatika!

Pirmasis veiksmas – atrinkti informaciją, kuri būtina uždaviniui išspręsti, ir atsisakyti perteklinės.

Reikšmingą informaciją sudaro:

- Kanalai ir kanalų sankirtos (akmenys, starto vieta ir sandėlis).
- Matematiniai apribojimai: bebras nuo vienos kanalų sankirtos iki kitos plaukia lygiai 1 valandą; bebrai turi sunešti akmenis į sandėlį per 4 valandas; jie turi surinkti 31 akmenį; jie gali vienu metu nešti vieną arba du akmenis.
- Reikia surinkti visus 31 akmenį naudojant kuo mažiau bebrų.

Tiksli kanalų sistema ir jų išsidėstymas, bebrų atsiradimas reikiamoje sankirtoje ir panaši informacija yra perteklinė, tad ją galima praleisti.

Informatikoje toks formalus uždavinio vaizdavimas vadinamas grafu. Grafa sudaro mazgai ir briaunos tarp mazgų. Atstumą tarp dviejų mazgų galima išreikšti keliais būdais, pavyzdžiui, suskaičiuojant, kiek briaunų turime pereiti, kad patektume nuo vieno mazgo iki kito. Grafai yra viena iš pagrindinių informatikos uždavinių sprendimo metodų. Navigacijos sistemos yra populiarus taikomųjų programų, kuriose naudojami grafai, pavyzdys.

Šioje užduotyje prašoma surinkti visus akmenis. Gali būti, kad yra daugiau nei vienas būdas šiam tikslui pasiekti. Galima įsivaizduoti, kad reikia pasinaudoti 14, 18, 20 ar net 31 skirtingais bebrais. Tačiau užduotis aiškiai reikalauja rasti sprendimą, kuriam būtų panaudota kuo mažiau bebrų.

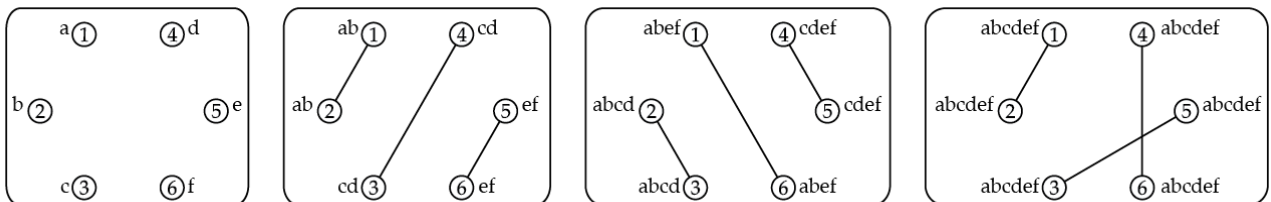
Uždaviniai, kurie reikalauja geriausio įmanomo sprendimo, labai dažni. Informatikoje jie vadinami optimizavimo uždaviniais. Pavyzdžiui, tikimasi, kad navigacinė sistema parinks geriausią būdą nuvykti iš taško A į tašką B. Žinoma, tikslią geriausio sprendimo reikšmę reikia apibrėžti a *priori*. Navigacinės sistemos atveju kriterijumi galėtų būti atstumo kilometrais arba laiko, kurio reikia automobiliui pasiekti tikslą, minimizavimas.



## 23. Šnipai

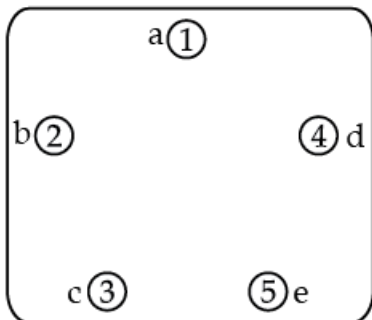
Kiekvieną penktadienį šeši šnipai keičiasi visa per savaitę surinkta informacija. Vienu metu šnipas negali susitikti su daugiau nei vienu šnipu. Todėl šnipai turi rengti kelis susitikimus, kuriuose susitiktų poromis ir apsikeistų visa tuo metu turima informacija.

Šešių šnipų grupei reikia tik trijų susitikimų apsikeisti informacija. Sutarta, kad prieš susitikimą kiekvienas šnipas turi po vieną informacijos vienetą (1-asis šnipas žino „a“, 2-asis šnipas – „b“ ir t. t.). Per pirmąjį susitikimą 1-asis ir 2-asis šnipai apsikeičia informacija, todėl abu žino „ab“. Schemoje linijomis parodyta, kurie šnipai dalyvauja kiekviename susitikime ir kokią informaciją jie turi. Po trijų susitikimų jie bus apsikeitę visa informacija.



Po tarptautinio incidento vienas šnipas nustojo dalyvauti susitikimuose.

Kiek mažiausiai susitikimų reikia, kad likę penki šnipai apsikeistų visa informacija?



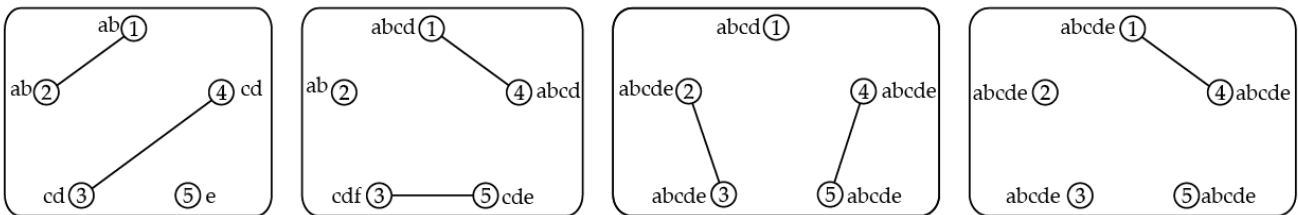
## Paaiškinimas

Teisingas atsakymas: 4 susitikimai.

Tai gali pasirodyti netikėta: manytume, kad atsakymas būtų trys (ar mažiau) susitikimų, nes vienu šnipu yra mažiau. Dar įdomiau, kad pasigilinę išsiaiškintume, jog keturi šnipai apsikeistų visa turima informacija per du susitikimus.

Tačiau nesėkmingi bandymai išspręsti uždavinį netrukus atskleidžia problemos esmę: kadangi šnipų skaičius yra nelyginis, kiekviename susitikime vienas iš jų yra „neaktyvus“. Sakykime, kad 5-asis šnipas nedalyvauja pirmame susitikime, bet dalyvauja antrame. Taigi antrame susitikime tik du šnipai žinos jo informacijos dalį (e). Trečiame susitikime šie du šnipai susitiks su kitais dviem šnipais, taigi po trijų susitikimų (tik) keturi šnipai žinos e. Ketvirtas susitikimas reikalingas tam, kad ši informacija būtų perduota penktajam šnipui.

Taigi įsitikinome, kad reikia bent keturių susitikimų. Įrodymui, kad pakanka keturių susitikimų, užtenka sukonstruoti tokią schemą:



Tai informatika!

Kompiuteriai, keisdami informaciją, dažnai ja keičiasi poromis. Neretai gali iškilti problema, kaip per kuo trumpesnę laiką apsikeisti informacija visame tinkle. Informatikai sprendžia daug panašių problemų.

Šis uždavinys dar vadinamas gandų (angl. *gossip*) problema (<http://mathworld.wolfram.com/Gossiping.html>).

Galite pabandyti išspręsti uždavinį esant skirtingam šnipų skaičiui ir galbūt pastebėsite įdomių dėsningumų.

Pirmą kartą šio uždavinio sprendimas buvo pateiktas ir surasta bendra taisyklė aprašyta 1975 metais. Šis ir daugelis panašių uždavinių yra susiję su keitimusi duomenimis, ryšių tinklais ir kriptografija.