



ŠVIETIMO,
MOKSLO
IR SPORTO
MINISTERIJA



www.bebbras.lt

Informatikos
ir informatinio
mąstymo konkursas

·b·e·b·r·a·s[®]

© Bebras LT

Bebras
bebras.org

**Informatika:
Bebras 2019**

Informatika: Bebras 2019

XVI informatikos ir informatinio mąstymo konkurso 2019 metų
pirmojo etapo uždaviniai su sprendimais

Parengė: Valentina Dagienė, Vaidotas Kinčius, Vilniaus universiteto Duomenų mokslo ir skaitmeninių technologijų institutas

Verčiant ir adaptuojant uždavinius talkino: Daumilas Ardickas, Tolmantas Dagys, Viktoras Dagys, Eglė Jasutė, Tatjana Jevsikova, Alvida Lozdienė, Bronius Skūpas, Lina Vinikienė, Eimutis Karčiauskas, Rasa Mažutienė, Karolis Jasutis, Elena Sutkutė, Ieva Jonaitytė, Indra Sudeikienė, Kunigunda Zubytė, Tomas Šiaulys, Rimantas Žakauskas

Viršelį sukūrė Vaidotas Kinčius

Šią knygelę galima laisvai parsisiųsti ir ja naudotis.

Siūlome knygelėje teikiamą medžiagą naudoti klasėje ar užklasinėje veikloje.

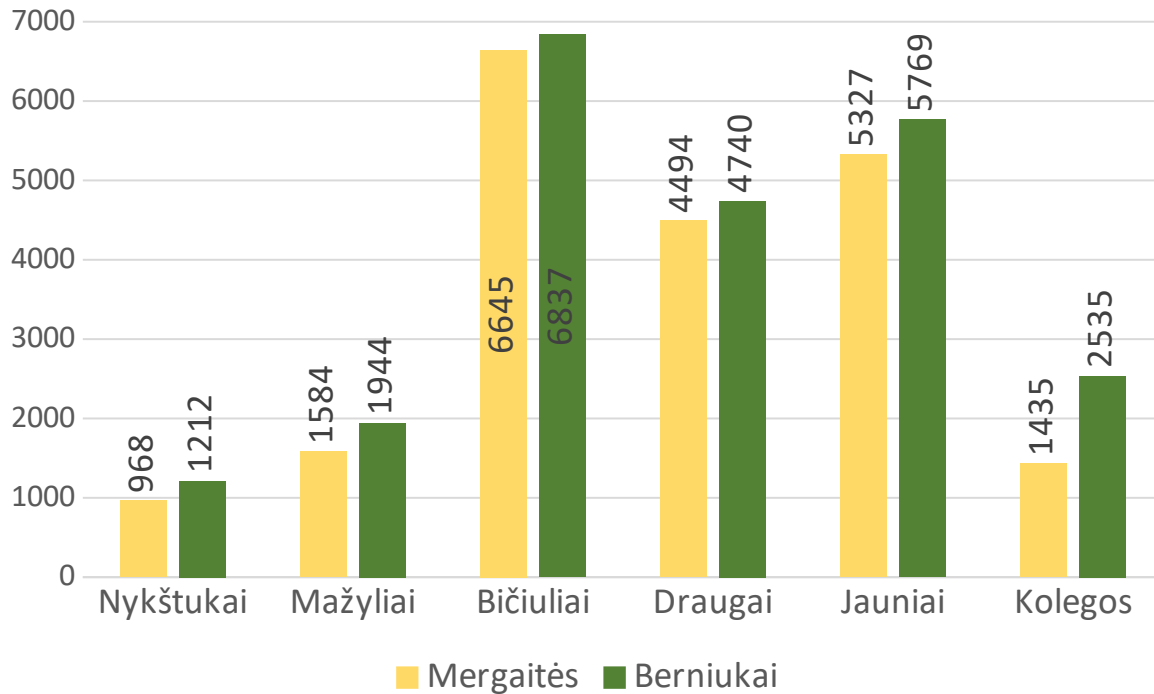
Knygelė platinama pagal kūrybinių bendrijų licenciją nekomerciniais tikslais (*Creative Commons Attribution–NonCommercial– ShareAlike*). Tai reiškia, kad knygelė galima laisvai dalytis: ją kopijuoti, platinti ar persiųsti. Daugiau informacijos apie šią licenciją galite rasti internete, ieškokite CC BY–NC–SA 3.0

Kūrybinių bendrijų licencija, 2019

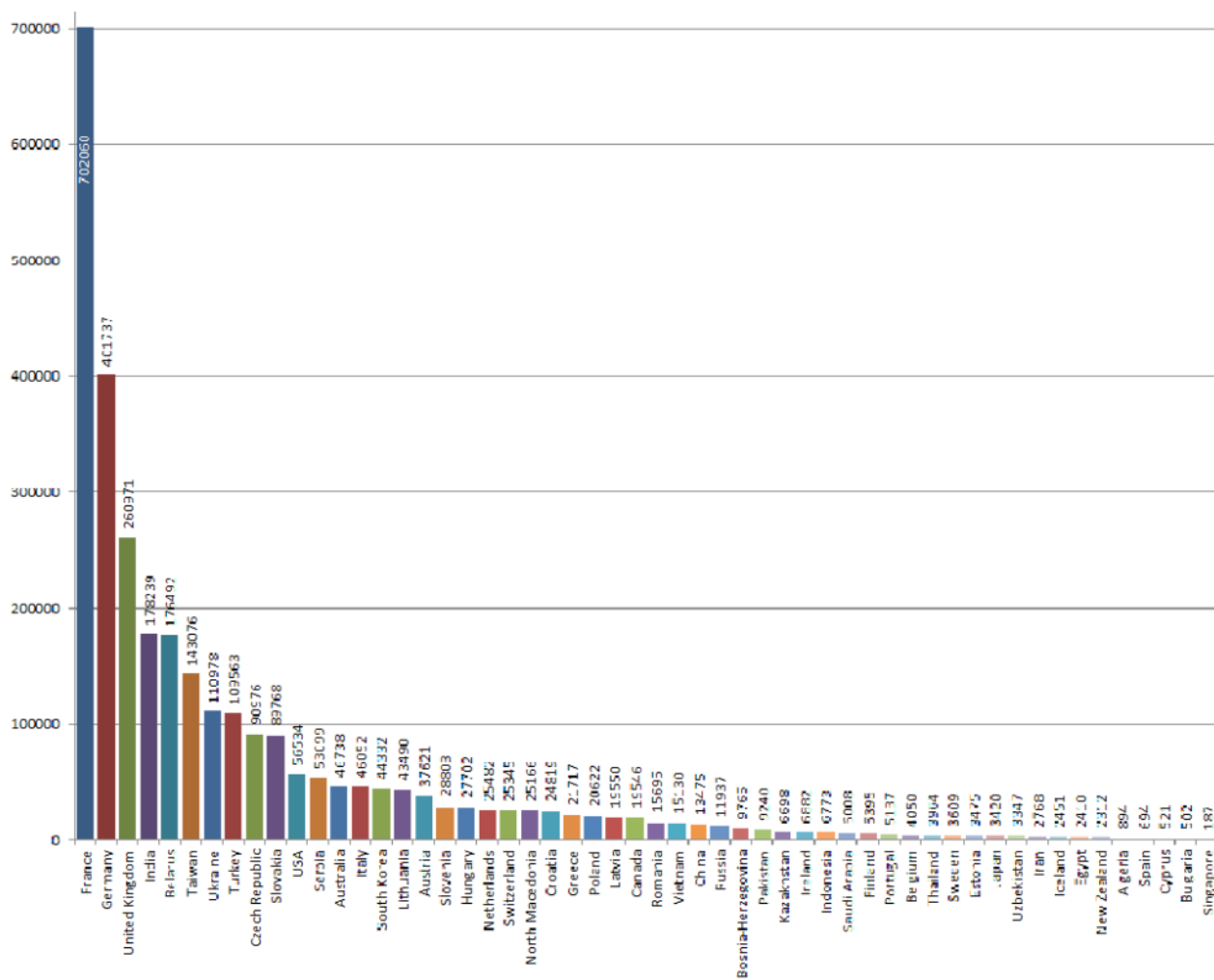
IVADAS

2019 lapkričio 4–15 d. įvyko 16-asis mokinių informatikos ir informatinio mąstymo konkursas „Bebras“. Kiekvienais metais Lietuvoje labai daug mokinių sprendžia šio konkurso uždavinius, tad šiais metais konkurse dalyvavo 43 555 mokiniai:

- 🕒 2 180 nykštukai (1–2 klasės),
- 🕒 3 538 mažyliai (3–4 klasės),
- 🕒 13 497 bičiuliai (5–6 klasės),
- 🕒 9 272 draugai (7–8 klasės),
- 🕒 11 098 jaunieji (9–10 klasės),
- 🕒 3 970 kolegos (11–12 klasės).



2019 metų „Bebro“ užduotis pasaulyje sprendė per 2 977 000 mokinių iš 54 valstybių.



Knygelės rengėjai dėkoja visiems kolegoms, talkinantiems rengiant ir aptariant „Bebro“ konkurso uždavinius, organizuojant susitikimus su mokytojais ir mokiniais.

Mieli skaitytojai, lauksime Jūsų siūlymų, samprotavimų, komentarų. Iš anksto dėkojame.

Rašykite info@bebras.lt

Prof. Valentina Dagienė
Vilniaus universitetas

I etapo uždaviniai:

Nr.	Uždavinio pavadinimas	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
1	Gimtadienio tortas	6					
2	Saldainiai	6	6				
3	Triušis miške	6	6				
4	Nuotrauka	6	6				
5	Bebrų monetos	9	6	6			
6	Meninio braižymo popierius	9	6				
7	Antspaudai	9	9	6			
8	Valymas	9	9	6			
9	Bendravimas debesimis	12	9	6			
10	Maudynės paplūdimyje	12	9	6			
11	Kojinės	12	6				
12	Kuris bokštas?	12	12	6			
13	Išskirtiniai bokštai		9		6		
14	Spalvotos bebrų vėliavos		12	9	6		
15	Piešiantis robotas		12	9	6		
16	Spalvotas kinų hieroglifas		12	12			
17	Besmegenių kepurės		12	9			
18	Kelionės kosmose			9	6		
19	Vyresniųjų bebrų pranešimas			12	9	6	
20	Automobilių parkavimas			12	9	6	
21	Garsenybė			12	6		
22	Rangoli menas			9	9	6	
23	Mėsainių ingredientai			12	9	6	
24	Lėkštės			9	6		
25	Alergiški bebrai			12	9	6	
26	Rutuliai				12	9	6
27	Saldainių maišelis				12	9	6
28	Kipu				12	9	6
29	Batų pirkimas				12	9	6
30	Skrudės pelkėje				12	9	6
31	Stebėjimas				12	9	6
32	Spąstų sala				9	6	
33	Lentpjūvė					12	12
34	Ekologiški skrydžių maršrutai					12	12
35	Traukinio vagonų rikiavimas					12	9
36	Gelbėjimo misija					12	9
37	Dvejetainės lemputės					12	9
38	Vaizdų glaudinimas					12	9
39	Bebrų keliai						12
40	Geležinkelio elektrifikavimas						9
41	Mokslininkų eksperimentai						9
42	Vandens pilstymas						12
43	Šifruotas žemėlapis						12
44	Dvejetainis skaitiklis						12

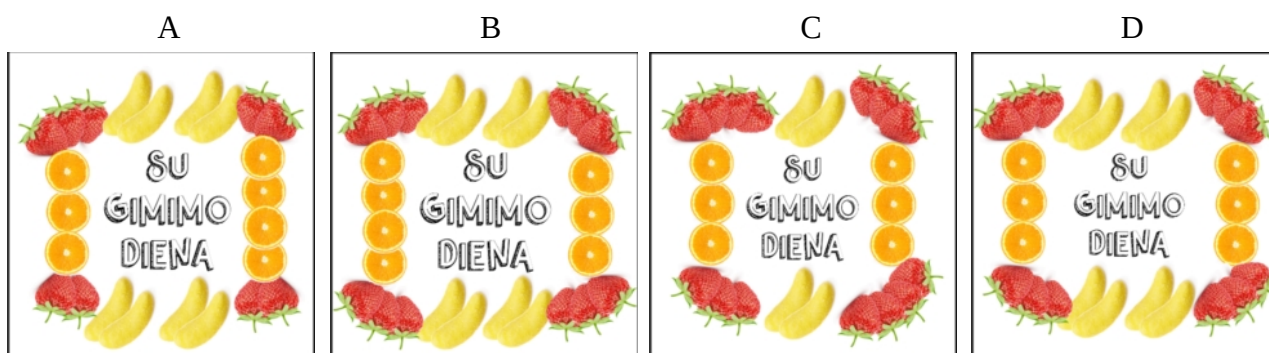
	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	6					

1. Gimtadienio tortas

Natalijos mama jos gimtadieniui perka tortą. Natalija nori, kad tortas būtų su

- trimis braškėmis  kiekviename torto kampe;
- ne daugiau negu trimis apelsino griežinėliais  kairėje ir dešinėje pusėje;
- dviem poromis bananinių ledinukų  viršuje ir apačioje.

Kurį iš šių tortų turėtų pirkti Natalijos mama?



Paiškinimas



Atsakymas: D.

- A tortas turi 4 apelsino griežinėlius dešinėje pusėje.
- B tortas turi 4 apelsino griežinėlius kairėje pusėje.
- C tortas turi tik vieną bananinių ledinukų porą.
- D tortas turi tris braškes visuose kampuose, 3 apelsino griežinėlius kairėje ir dešinėje pusėje ir po dvi poras bananinių ledinukų viršuje ir apačioje.

Tai informatika!

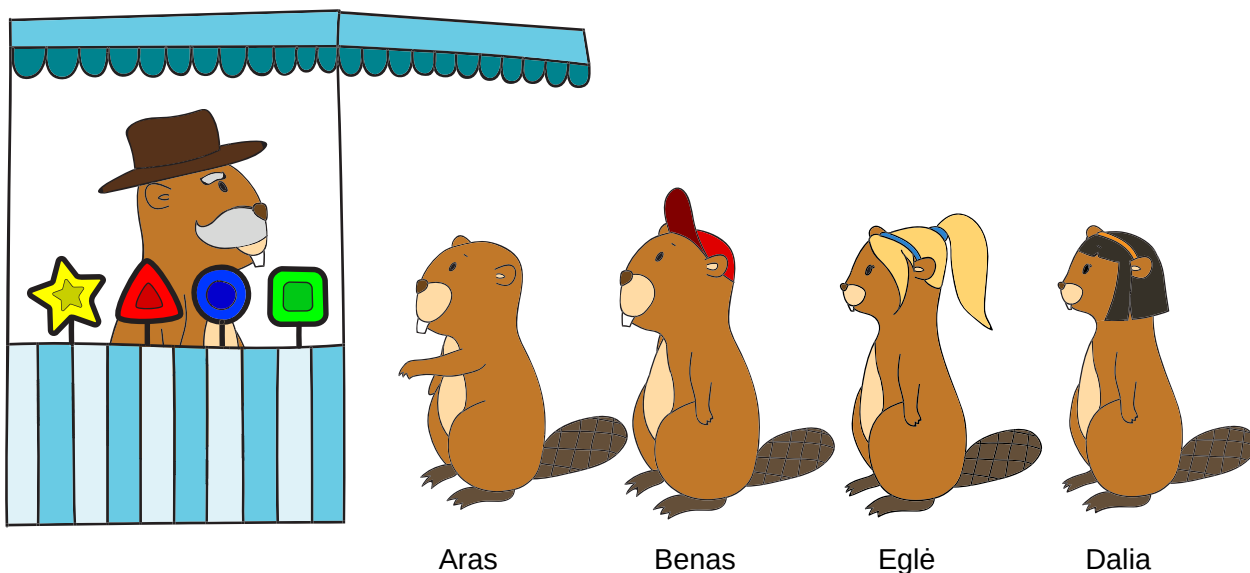
Dekoruojujant tortą vaisiai išdėstomi kaip šablonas. Identifikuojant šabloną turime patikrinti visus reikalavimus.

Natalijos mama turi pirkti tortą su nurodytais ingredientais tam tikroje torto vietoje. Šie reikalavimai yra problemos specifikavimas. Kai tik naudotojas gauna reikalavimus, jie virsta taisyklėmis, kurios turi būti vykdomos. Turi būti patenkinti trys sąlyginiai sakiniai, norint pasiekti pageidaujamą tikslą.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	6	6				

2. Saldainiai

Bebrai Aras, Benas, Eglė ir Dalia stovi eilėje prie saldainių kioskelio. Kiekvienas jų gaus po saldainį. Pardavėjas turi tik po vienos rūšies saldainį ir jis visuomet duoda tą saldainį, kuris yra arčiausiai stovinčiojo bebro. Pavyzdžiui, žalią kvadratinį saldainį  gaus Aras.



Kas gaus raudoną trikampį saldainį ?

Paaiškinimas

Atsakymas: C) Eglė

Aras gaus pirmąjį saldainį, kuris yra arčiausiai jo – žalią keturkampį .

Benas gaus apvalų mėlyną saldainį , nes jis yra arčiausiai žaliajo saldainio, kuris bus atiduotas Arui.

Eglė gaus raudoną trikampį saldainį .

Dalia gaus paskutinį saldainį – geltoną žvaigždę .

Tai informatika!

Šiame uždavinyje yra dvi eilės: saldainių ir bebrų. Žinome, kad bebrams gaus saldainius. Turime susitarti, kokia tvarka bus duodami saldainiai. Sąlygoje nurodoma, kad artimiausias saldainis bus įteiktas pirmajam bebrui eilėje ir taip toliau.

Informatikoje labai svarbu suformuluoti konkrečias instrukcijas programai. Saldainių dalinimo bebrams metodas yra toks pat, kaip kai kuriose informatikoje naudojamose populiariose duomenų struktūrose.

Pavyzdžiui, bebrai stovi eilėje, kurios duomenų struktūra ir vadinama eile (nauji bebrai pridedami eilės gale, o pirmasis bebras gauna pirmąjį saldainį). Jei saldainiai dedami į krūvą vienas ant kito, tai pirmas bebras gauna vėliausiai padėtą saldainį (tai dėklo arba steko duomenų struktūra). Gebėjimas suprasti duomenų gavimo taisyklės – tvarką yra labai svarbus informatikos įgūdis.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	6	6				

3. Triušis miške

Triušis nori pasivaikščioti savo miške.

Jis pradeda langelyje, kuris pažymėtas triušiu, ir toliau vykdo šias instrukcijas:

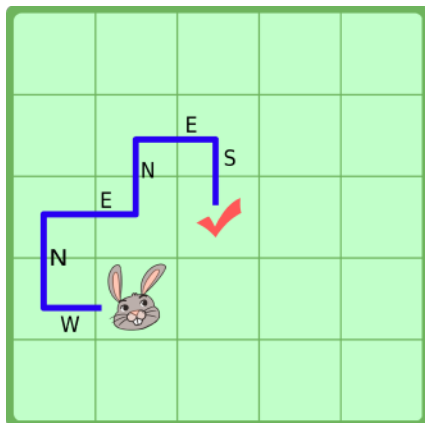
V Š R Š R P

Kiekviena raidė žymi kryptį pagal kompasą, kuria triušis turi judėti. **Pažymėk langelį, kuriame bus triušis, kai įvykdys nurodytas instrukcijas.**



Paaiškinimas

Atsakymas: Vykdydami nurodymus, jūs pateksite į šį langelį:



Tai informatika!

Užduotis susijusi su instrukcijų vykdymu. Šiuo atveju instrukcijos - tai eilės tvarka nurodytos judėjimo kryptys. Judėjimas vykdomas iš vieno langelio į kitą.

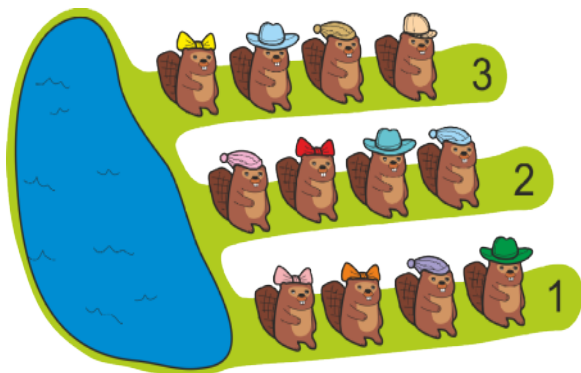
Miškas pavaizduotas labai abstrakčiai, kaip 25 langelių laukas. Judėjimas šiame miške vykdomas tik vertikaliai arba horizontaliai. Visos detalės, kurios nereikalingos šiai problemai išspręsti, buvo pašalintos.

Daugiau informacijos apie abstrakciją: <https://en.wikipedia.org/wiki/Abstraction>.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	6	6				

4. Nuotrauka

Bebriukai fotografuojasi. Jie sustoję trimis eilėmis po 4 kiekvienoje.



Mama bebrė nori, kad kiekvienoje eilėje bebbriukai su kaspinais būtų arčiau ežero negu bebbriukai su kepurėmis.

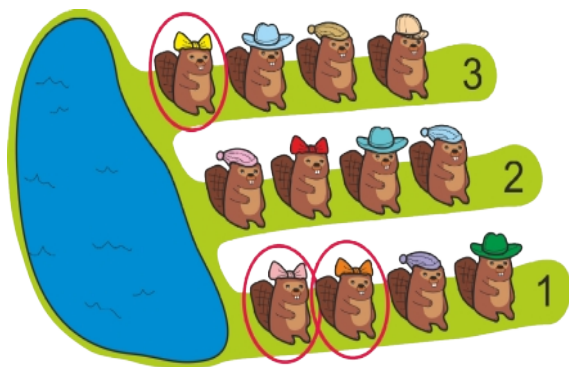
Kuri bebbriukų eilė sustojo pagal mamos pageidavimą?

- A) Tik 1
- B) 1 ir 2
- C) 1 ir 3
- D) Visos eilės

Paaikkinimas

Atsakymas: C)

Atidžiai nagrinėdami, matome, kad 1-oje ir 3-ioje eilėse bebrukai su kaspinais stovi arčiau ežero negu bebrukai su kepurėmis. Deja, antroje eilėje bebrukas su kepure stovi arčiau ežero negu bebrukas su kaspinu.



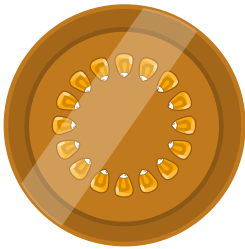
Tai informatika!

Šiuo uždavinys skirtas logikai: ar suprantamos taisyklės ir loginiai teiginiai. Tai labai reikalinga rašant programas: būtina aptarti ir surinkti visu reikalavimus būsimai programai. Pavyzdžiui, vartotojas gali pageidauti, kad programa išspausdintų duomenis tam tikra tvarka. Tokio pobūdžio reikalavimai vadinami uždavinio (problemos) specifikacija. Tik įvertinus specifikacijos duomenis imama rašyti programas. Šie reikalavimai, sudėjęs juos drauge, gali būti laikomi taisyklėmis.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	9	6	6			

5. Bebrų monetos

Bebrijos šalyje naudojamos 16, 8, 4, 2 ir 1 bebroeurų vetės monetos:

				
16	8	4	2	1

Bebrai nemėgsta nešiotis daug monetų, todėl turi taisyklę visada mokėti sumą kuo mažesniu monetų kiekiu.

Kurias monetas bebras turėtų pateikti mokėdamas 13 bebroeurų?

Paiškinimas

Atsakymas: Optimalus ir vienintelis teisingas atsakymas yra ,  ir  (viena 8 vertės bebroeurų moneta, viena 4 vertės moneta ir viena 1 vertės moneta), nes $8 + 4 + 1 = 13$. Mažesniu monetų kiekiu sumokėti neįmanoma, nes didesnės monetos nei 8 vertės imti negalima (16 vertės moneta viršytų norimą sumokėti monetų sumą), o 5 vertės monetos nėra. Kita mažesnė kaip 5 vertės moneta yra 4 vertės moneta. Mokinys galėtų pradėti pasirinkdamas kitas monetas, pavyzdžiui, 2×4 vertės, 1×2 vertės ir 3×1 vertės monetas. Tada mokinys galėtų mažinti bebroeurų monetų kiekį iškeisdamas kelias mažesnes vertės monetas į vieną didesnės vertės monetą.

Tai informatika!

Programuotojai yra įvairių tipų duomenų pateikimo ekspertai. Vienas iš tokių duomenų tipų yra skaičiai. Šioje užduotyje minima suma gali būti pateikta bet koku monetų skaičiumi, atsižvelgiant į sąlygą, kad surinktų monetų verčių suma atitinka nurodytą sumą. Tačiau tokie pateikimai yra nevienareikšmiški, todėl reikia pasirinkti monetų rinkinį, kuris turėtų mažiausią monetų skaičių. Tai galima pasiekti dvejetainiu būdu, keičiant monetas į tolesnes didesnes, kol didesnės vertės monetos nebėra. Garsusis abakas, šimtmečius naudotas įvairioms skaičiavimo sistemoms, veikia atsižvelgiant į tokius pačius principus.

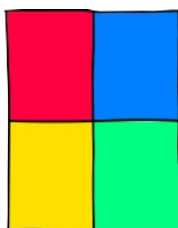
	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	6					

6. Meninio braižymo popierius

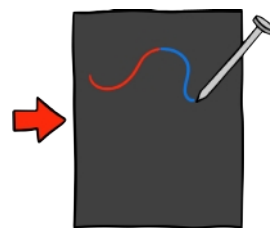
Ant meninio braižymo popieriaus su aštriu objektu galima nubraižyti paveikslėlį.



Iš pradžių matote juodos spalvos popierių.

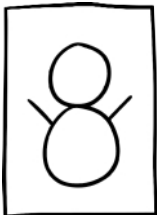
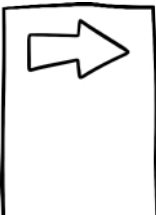
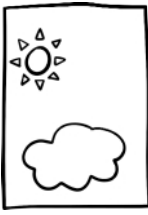


Šios keturios spalvos slepiasi po juoda spalva.



Kai braižote ant popieriaus su aštriu objektu, atsiranda spalvos.

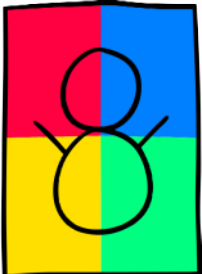
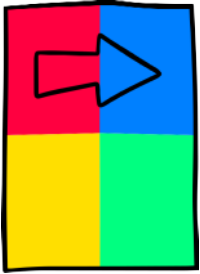
Kuris iš nurodytų paveikslėlių braižant su aštriu objektu užimtų tris spalvas?

A)	B)	C)	D)
			

Paaškinimas

Atsakymas: C.

Kiekvienas paveikslėlis patenka ant nurodyto spalvų kiekio:

A)	B)	C)	D)
			
4 spalvos	2 spalvos	3 spalvos	4 spalvos

Tai informatika!

Norėdami sužinoti, kiek spalvų užims paveikslėlis, turite pagalvoti, kur po juoda spalva yra tikrosios spalvos.

Dirbdami su kompiuterių programine įranga paveikslams piešti, dažnai tenka susidurti su sluoksniais, kurie atrodo kaip popieriaus lapai vienoje krūvoje. Sluoksniai gali maišytis vienas su kitu.

Piešimo programos turi dar daugiau galimybių: jos leidžia pakeisti sluoksnių tvarką, be to, jos leidžia pašalinti sluoksnio dalis. Tai vadinama skaidrumu.

Bet kuriame tinklapyje pamatysite daugybę dalykų, sudėtų vienas ant kito: fono paveikslėlį, spalvas, tekstą ir t.t. Kompiuteris turi žinoti, kokia tvarka dalykai išdėstomi vienas ant kito. Juose tai vadinama z indeksu - skaičiumi, nurodančiu, kiek viršuje yra sluoksnių.

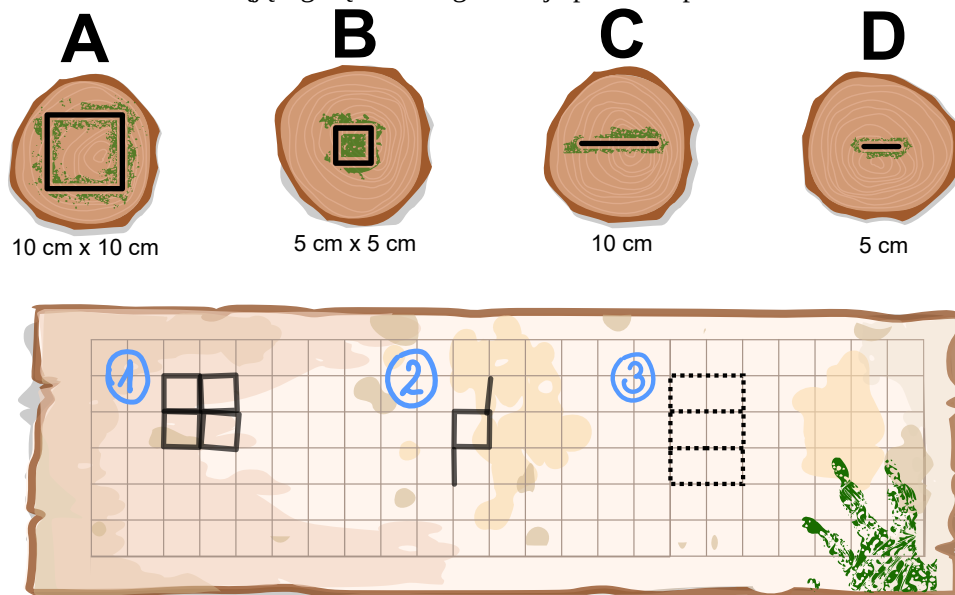
	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	9	9	6			

7. Antspaudai

Bebras Paulius turi 4 antspaudus: A, B, C ir D. Naudodamasis šiais antspaudais Paulius jau sukūrė 1-ąją ir 2-ąją figūras, pateiktas paveikslėlyje žemiau.

- Kurdamas pirmąją figūrą Paulius keturis kartus naudojo antspaudą B.
- Kurdamas antrąją figūrą Paulius vieną kartą panaudojo antspaudą B ir du kartus – antspaudą D.

Dabar Paulius nori sukurti 3-ąją figūrą. Jo draugė Marija pasiūlė padėti.



Marija teigia, kad ji gali sukurti 3-ąją figūrą padėdama vieną antspaudą du kartus!

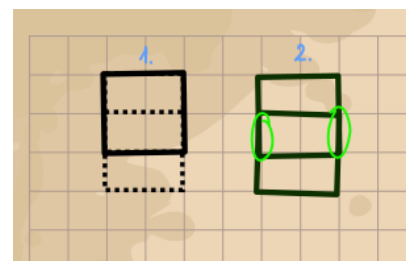
Pažymėkite, kurį antspaudą Marija naudos.

A) Didelis kvadratas	B) Mažas kvadratas	C) Ilga linija	D) Trumpa linija
			

Paiškinimas

Atsakymas: A.

Marija naudoja antspaudą A () , kurio dydis yra 10 cm x 10 cm. Pirmiausia ji padeda antspaudą, o antrą kartą padeda truputėlį žemiau. Tokiu būdu dvi figūros vietos yra pažymėtos du kartus (apibrėžtos žaliai).



Tai informatika!

Užduotis taip pat gali būti išspręsta daug kartų panaudojus antspaudą „Trumpa linija“. Daugelyje užduočių yra daugiau nei vienas atsakymas ir dalį tų atsakymų yra lengviau gauti, bet ne visais atvejais lengviausiai rasti atsakymai yra optimalūs. Pagal pasirinktą sprendimo būdą žingsnių skaičius, šiuo atveju antspaudų skaičius, gali keistis (kompiuteriuose kinta skaičiavimų kiekis). Viena iš pagrindinių informatikos problemų yra surasti optimalų sprendimą iš visų įmanomų.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	9	9	6			

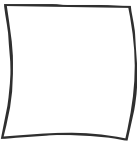
8. Valymas

Valymo robotas surenka šiukšles, kurias po koncerto dalyviai paliko ant žemės:



Robotas juda prie artimiausios šiukšlės ir ją pakelia. Po to, juda prie kitos artimiausios šiukšlės ir taip toliau, kol veja lieka švari ir visos šiukšlės surinktos.

Kuria šiukšlę robotas pakels paskutinę?

A	B	C	D
			

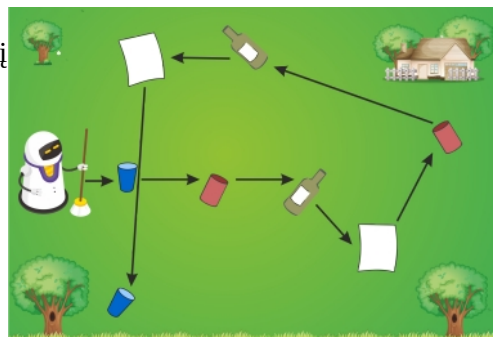
Paaškinimas

Atsakymas: B.

Paveikslėlyje parodyta, koku keliu robotas kelias atsižvelgdamas į

instrukcijas. Pirmiausia robotas pakels , tada judės link

 ir taip toliau, kol paskutinė pakelta šiukšlė bus popierinis puodelis .



Tai informatika!

Roboto veiksmai yra kontroliuojami kompiuteriu, kuris yra roboto viduje. Programa roboto kompiuteryje yra parašyta robotui suprantama kalba. Kameros ir jutikliai leidžia robotui surasti šiukšles ir nustatyti atstumą iki jų. Robote esančios programos kontroliuoja jį, taip leisdamos robotui judėti be žmogaus įsikišimo. Atliekų rinkimo robotai atlieka darbus, kurie gali būti pavojingi žmonėms. Todėl savarankiškai dirbantys robotai gali būti pagalbininkai pavojinguose darbuose ar nelaimių vietose.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	12	9	6			

9. Bendravimas debesimis

Už orus atsakingas bebras, nuo kalno viršaus į slėnį, siunčia orų prognozę. Jis sukuria mažus ir didelius dūmų debesis ir naudoja juos kaip kodą.

			
Audra	Silpnas lietus	Debesuota	Giedra

Vieną vėjuotą dieną, bebrai slėnyje galėjo matyti tik du didelius dūmų debesis:



Pasirinkite visus pranešimus, kuriuos galėjo nusiųsti bebras.

- A) Audra
- B) Silpnas lietus
- C) Debesuota
- D) Giedra

Paaškinimas

Atsakymas: A – Audra ir C – Debesuota.

Abiejuose koduose antras ir ketvirtas debesis yra dideli. Pasirinkimai B ir D negalimi, nes abu turi mažus debesis antroje arba ketvirtoje pozicijoje.

Tai informatika!

Jei yra kuriama simbolių seka, skirta komunikavimui (tiek žmonėms, tiek kompiuteriams), geriausia tai daryti tokiu būdu, kad net praradus dalį informacijos, seka galėtų būti atkurta. Tai atliekama siunčiant daugiau informacijos nei yra reikalinga žinutei perduoti, taip esminė informacija išliks. Jeigu originali simbolių seka gali būti atkurta, kai pasitaikė n klaidų, tai ši seka vadinama n klaidų taisymo kodu.

Programavime tai naudojama dažnai, pavyzdžiui, siunčiant muziką skaitmeniniu formatu. Muzika bus grojama be trukdžių, nors duomenys gali būti dalinai pažeisti.

Šioje užduotyje, užtektų dviejų debesų, kad būtų galima nuspėti žinutę:



. Jei naudosite penkis debesis keturių reikšmių kodavimui, tai, jei vienas, du ar trys debesis bus

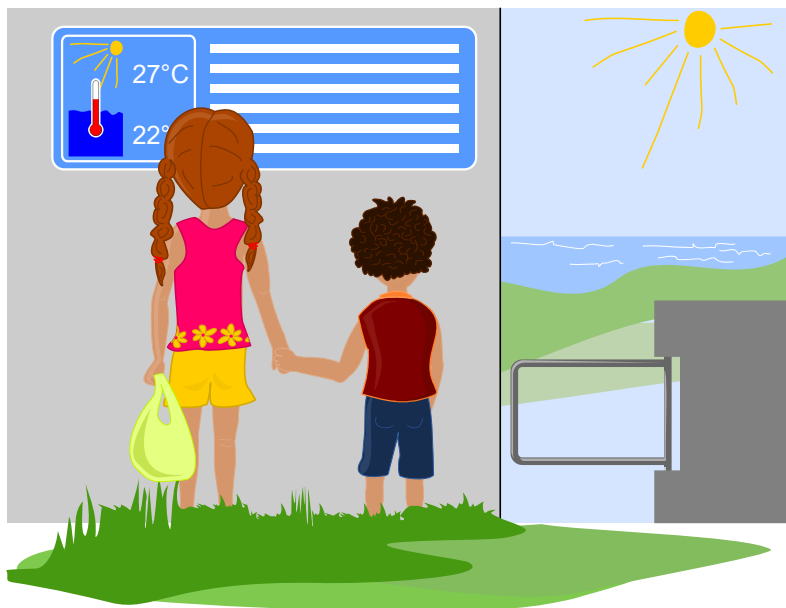


nematomi, žinutė vis tiek bus atpažinta. Pagrindinis bebrų kodavimo principas: visos kodų sekos skiriasi viena nuo kitos bent trimis debesų pozicijomis.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	12	9	6			

10. Maudynės paplūdimyje

Vasara kaime: karšta ir kasdien darosi vis karščiau. Dvylikametė Angelė nori išsimaudyti netoliese esančiame paplūdimyje ir kartu pasiimti šešiametį brolių Fredą.



Atvykusi Angelė perskaito informaciją, kurioje nurodyta, kad norint patekti į paplūdimį reikia būti bent 8 metų, kitu atveju reikalingas lydinčysis, vyresnis nei 10 metų.

Kas gali patekti į paplūdimį?

- A) Angelė ir Fredas gali patekti kartu.
- B) Angelė gali patekti, bet Fredas negali patekti kartu su Angele.
- C) Fredas gali patekti, bet Angelė negali patekti kartu su Fredu.
- D) Nei vienas negali patekti.

Paiškinimas

Atsakymas: A) Angelė ir Fredas gali patekti kartu.

Fredas gali patekti tik todėl, kad Angelė jį lydi. Fredas yra jaunesnis nei 8 metų IR Angelė yra vyresnė nei 10 metų.

Tai informatika!

Šis uždavinys pagrįstas sąlyginių sakinių (if... else) samprata: jeigu pateikta loginė sąlyga tenkinama, tai nurodytas įvykis įvyksta arba ne. Sąlyginiai sakiniai yra bet kurios programavimo kalbos viena iš pagrindinių konstrukcijų.

Loginė operacija IR skirta sujungti dvi sąlygas. Šiame uždavinyje pateiktos dvi sąlygos: Angelės amžius lyginamas su reikšme 10 ir Fredo amžius lyginamas su reikšme 8. Užduotis gali būti suformuluota taip: „jeigu Fredui < 8 metai, tada lydinčiam asmeniui privalo būti > 10 metų“. Programavimo kalbos turi sąlyginių operacijų, kuriomis galima aprašyti šio uždavinio sprendimą, remiantis loginiais teiginiais, kurių reikšmė yra teisinga arba klaidinga.

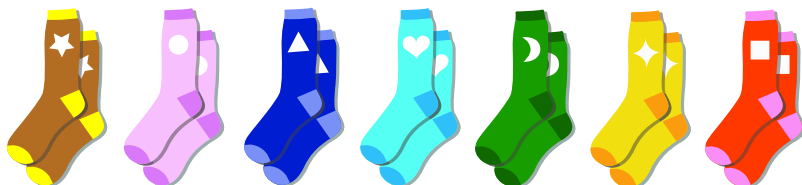
	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	12	6				

11. Kojinės

Ana mėgsta dėvėti vis kitos spalvos kojines kiekvieną dieną, jei tik išeina iš namų. Ana laikosi šių taisyklių:

- Kojinės skalbiamos kasdien. Išskalbus jos dedamos į kojinių eilę iš kairės.
- Kojinės, kurias dėvės, visada imamos iš dešinės kojinių eilės pusės.

Trečiadienį kojinių eilė atrodo taip:



Ana niekur neis šios savaitės sekmadienį bei kitos savaitės antradienį ir ketvirtadienį.

Kokios spalvos kojines dėvės Ana artimiausią šeštadienį?



A



B



C



D

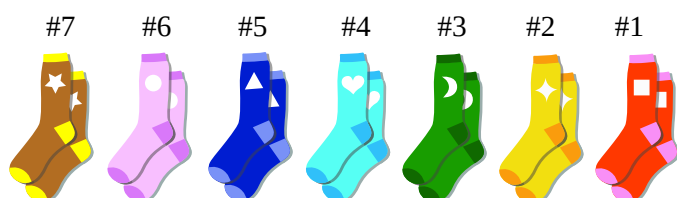
Paiškinimas

Atsakymas: A.

Kadangi Ana dėvi kojines, paimitas iš dešinės pusės, ir ji pradeda raudonomis kojineimis trečiadienį, kojines, kurias ji dėvės likusiomis savaitės dienomis taip pat ir kitą savaitę, nurodytos lentelėje (skaičius skliausteliuose pažymi poziciją paveikselyje):



Savaitės diena	Kojinių spalva [šią savaitę]	Kojinių spalva [kitą savaitę]
Pirmadienis		Mėlyna (#5)
Antradienis		- Neišeina -
Trečiadienis	Raudona(#1)	Purpurinė (#6)
Ketvirtadienis	Geltona(#2)	- Neišeina -
Penktadienis	Žalia (#3)	Tamsiai raudona (#7)
Šeštadienis	Žalsvai mėlyna (#4)	Raudona
Sekmadienis	- Neišeina -	



Tai informatika!

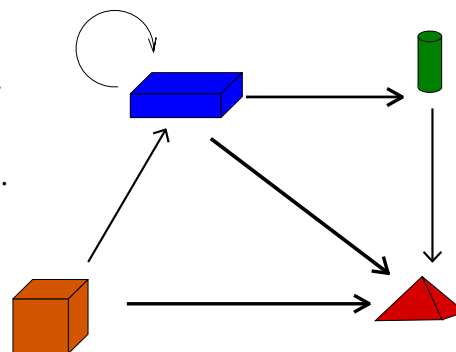
Informatikos moksle eilė yra duomenų struktūra, kur duomenys, pirmiau patekę į eilę, pirmiau iš jos ir paaimami (angl. First In First Out – FIFO). Šis principas yra panašus į stovėjimą eilėje norint patekti į mokyklinį autobusą: pirmas eilėje laukiantis vaikas į autobusą įlips pirmas, o vėliausiai atėjęs vaikas atsistos į eilę paskutinis. Eilė yra viena iš būtinų duomenų struktūrų, naudojamų programose ir kompiuterių procesuose. Eilės pavyzdžiai: spausdintuvo eilė (dokumentų, pasiūstų spausdinti, eilė), klaviatūros buferis (klavišų paspaudimai yra laikomi eilėje) ir kt.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
	12	12	6			

12. Kuris bokštas?

Ela metė iššūkį savo draugei Veronikai pastatyti bokštą iš medinių kaladėlių, laikantis tam tikrų taisyklių. Taisyklės parodytos paveiksle.

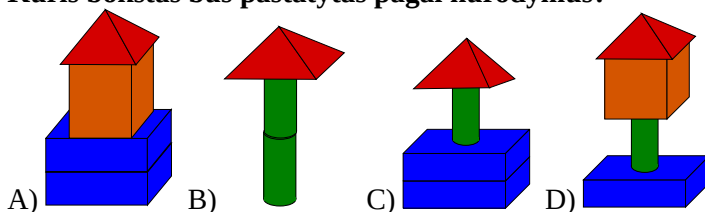
Kaladėlė gali būti uždėta ant kitos kaladėlės viršaus tik tada, jeigu rodyklės pradžia yra ties pirmą kaladėlę, o pabaiga ties antrą kaladėlę. Pavyzdžiui, piramidės formos kaladėlė gali būti uždėta ant stačiakampės formos kaladėlės viršaus, kai paveiksle parodyta rodyklė prasideda ties stačiakampės formos kaladėlę, o baigiasi ties piramidės formos kaladėlę.



Atkreipkite dėmesį, kad rodyklė vedanti nuo kaladėlės į tą pačią kaladėlę leidžia ant viršaus uždėti bet kokią skaičių kaladėlių.

Galima pradėti statyti bokštą su bet kokia kaladėle ir galite sustoti bet kuriuo momentu.

Kuris bokštas bus pastatytas pagal nurodymus?



Paaškinimas

Atsakymas: C.

Pradėkite nuo mėlynos plytos. Tada sekite rodyklę ir padėkite antrą mėlyną plytą. Rodyklė nukreipia į žalią cilindą. Padėkite cilindą. Nuo cilindro rodyklė veda iki raudonos piramidės. Uždėkite piramidę.

A – neteisingai pastatytas, nes rudas kubas uždėdamas ant mėlynos plytos. Taisyklėse nėra rodyklės nuo mėlynos plytos iki rudo kubo. Rodyklė eina kita kryptimi.

B – neteisingai pastatytas, nes taisyklės neleidžia dėti žaliųjų cilindų vienas ant kito.

D – neteisingai pastatytas, nes rudas kubas uždėdamas ant žalio cilindro, o taisyklėse tokios rodyklės nėra.

Tai informatika!

Paveikslas užduotyje yra blokinės schemos pavyzdys. Šiuo atveju rodyklė nurodo ryšį tarp blokų ir parodo, kokia tvarka galime dėti vieną bloką ant kito.

Tokios diagramos kaip ši, informatikoje naudojamos daugeliui tikslų: apibrėžti programavimo kalbos sintaksines taisykles, nurodyti ryšius tarp komponentų.

Blokinė schema gali būti naudojama atliekant bandymus, norint patikrinti, ar statinys pastatytas teisingai.

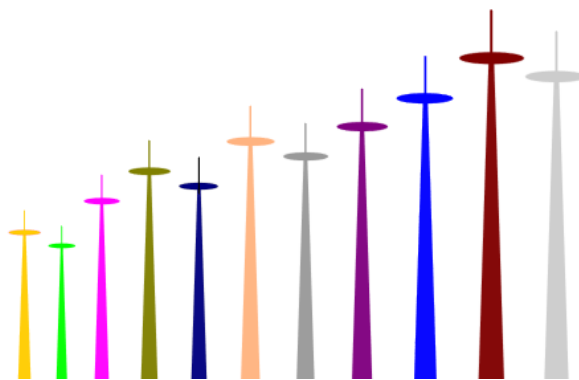
	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
		9		6		

13. Išskirtiniai bokštai

Panagrinėkite žemiau išdėstytus bokštus.

Bokštas vadinamas išskirtiniu, jeigu visi kairiau jo esantys bokštai yra žemesni už jį, o visi dešiniau jo esantys bokštai yra aukštesni už jį.

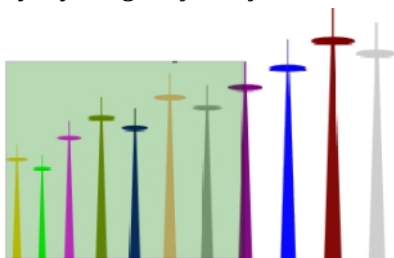
Kiek yra išskirtinių bokštų?



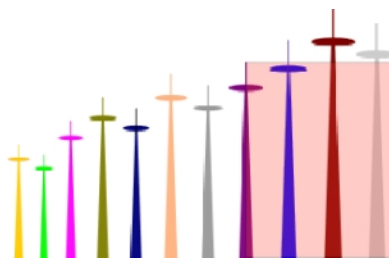
Paaiškinimas

Atsakymas: 3.

Bokštas yra išskirtinis, jeigu: visos kairėje esančios bokštų viršūnės yra spalvoto stačiakampio viduje (1 pav.) ir visos dešinėje esančios bokštų viršūnės nepatenka į spalvoto stačiakampio vidų (2 pav.). Pateiktas pavyzdys nagrinėja 8-ojo bokšto atvejį.

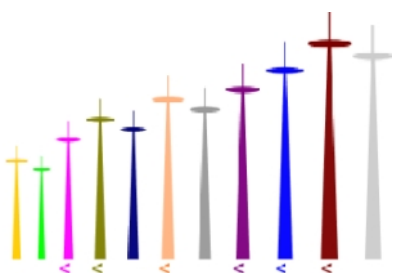


1 pav.

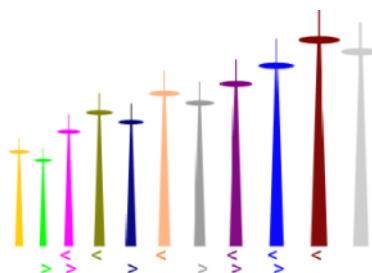


2 pav.

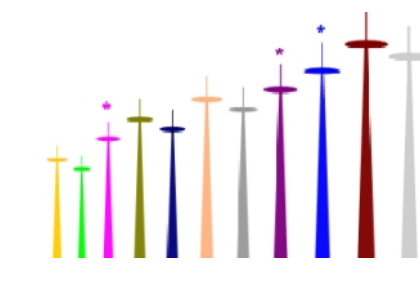
Kitas galimas sprendimo būdas: peržiūrėti kiekvieną bokštą ir pažymėti tuos bokštus, kurių kairėje pusėje visi bokštai yra žemesni (3 pav.). Tada peržiūrėti iš naujo bokštų eilę ir pažymėti tuos bokštus, kurių dešinėje pusėje visi bokštai yra aukštesni (4 pav.).



3 pav.



4 pav.



Visi du kartus pažymėti bokštai yra išskirtiniai bokštai (*).

Tai informatika!

Ši užduotis skirta susipažinti su rikiavimo konceptu. Objektų išdėliojimas pagal tam tikrą tvarką yra vadinamas rikiavimu, o tai yra viena iš labiausiai žinomų ir nagrinėjamų problemų informatikoje. Sukurta daug įvairių rikiavimo algoritmų. Greitojo rikiavimo algoritmas yra vienas žinomiausių ir dažniausiai naudojamų rikiavimo metodų, nes jis pasižymi spartumu, kaip ir nurodo pavadinimas!

Greitojo rikiavimo algoritmo veikimo principas: iš norimos rikiuoti duomenų sekos atsitiktinai imamas elementas. Šis elementas vadinamas slenksčiu. Visi elementai, mažesni už slenkstį, yra perkelti į kairę pusę nuo jo, o visi didesni elementai – į dešinę pusę. Taip gaunamos dvi duomenų sekos dalys – viena kairiau pasirinkto elemento, kita – dešiniau. Toliau tas pats kartojama rekursyviai. Po kiekvieno rikiavimo gaunamos duomenų sekos dalys mažėja. Rikiavimo procesas baigiasi, kai gautose duomenų sekos dalyse lieka tik vienas elementas, tada visa duomenų seka laikoma išrikiuota. Pateiktame uždavinyje išskirtiniai bokštai atlieka slenkščio funkciją: žemesni bokštai perkelti į kairę, o aukštesni – į dešinę. Daugiau galite paskaityti čia: https://inf-knyga.nmakademija.lt/lt/latest/06_rikiavimas.html

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
		12	9	6		

14. Spalvotos bebrų vėliavos

Laivų statykla „Bebras“ gamina vėlivis. Kiekvienas bebras nori turėti nuosavą vėliavą, tačiau kyla problema, kaip bebrui atpažinti savo vėliavą, jeigu jos visos vienodos! Kad galėtų atpažinti, bebrai nusprendžia kiekvieną vėliavą pažymėti vėliava. Vėliavos struktūra štai tokia:

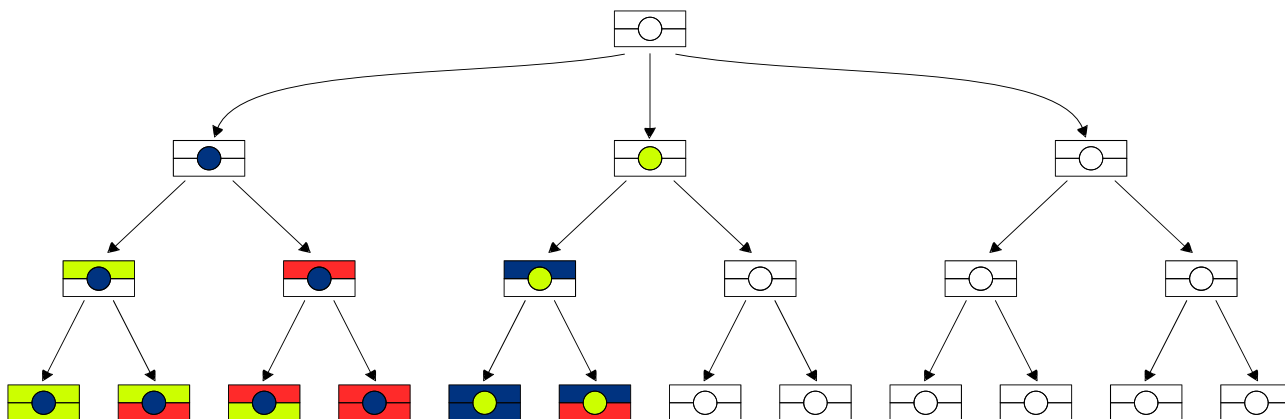


Bebrai susitaria dėl trijų skirtingų spalvų: raudonos, žalios ir mėlynos. Abi juostelės gali būti vienodos spalvos, tačiau centre esantis skritulys privalo būti kitos spalvos nei abi juostelės.



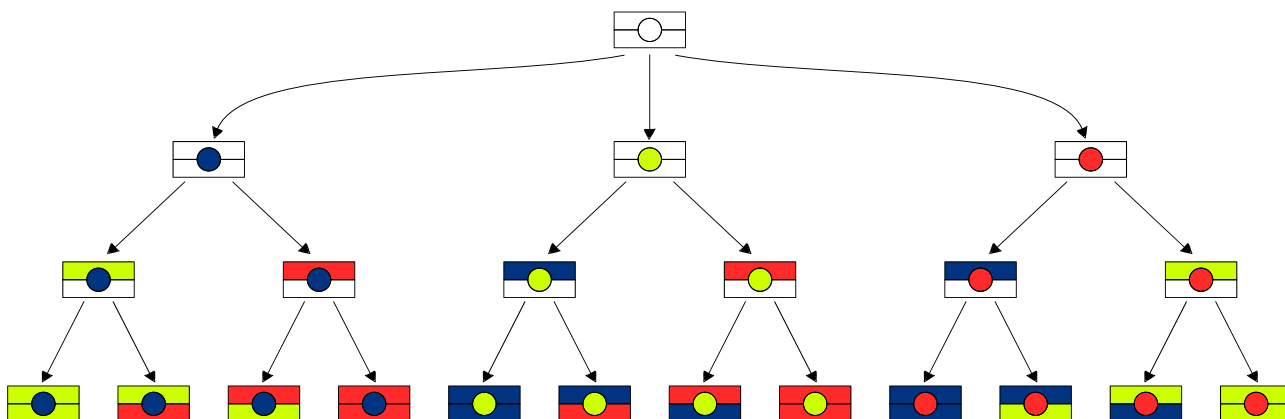
Bebrai pradėjo kurti schemą, pagal kurią būtų galima sudaryti visų spalvų vėliavų derinius. Deja, dėl audros jie nebaigė darbo ir kai kurios vėliavos liko nevisai nuspalyvintos.

Pabaikite užpildyti schemą: spauskite ant trūkstančių baltų laukų, kad galėtumėte juos nuspalyvinti. Atkreipiame dėmesį, kad galimi keli sprendimo variantai, bet jums pakanka rasti tik vieną.



Paiškinimas

Vienas iš galimų sprendimo variantų:



Bet kuris spalvų derinys yra teisingas, jei tenkinamos šios sąlygos:

- Antroje eilutėje dešiniausias skritulys turi būti raudonas.

- Trečioje eilutėje viršutinei juostelei nuspalvinti naudojamos likusios dvi spalvos iš trijų, kurios nėra panaudotos skritulio spalvinimui. Eiliškumas nėra svarbus.
- Ketvirtoje eilutėje apatinei juostelei nuspalvinti naudojama viršutinės juostelės spalva arba likusi trečia spalva, kuri nėra tokia pati kaip skritulio. Eiliškumas nėra svarbus.

Tai informatika!

Moderniame pasaulyje kiekvienas turi skirtingas užduotis. Vienos iš jų yra lengvai išsprendžiamos, kitų sprendimui reikia daugiau laiko. Sunkioms, sudėtingoms užduotims reikia daug laiko. Tokias užduotis galima spręsti išrašant visus galimus sprendimo variantus ir išrenkant geriausią sprendimą. Dėl šios priežasties programuotojams svarbu žinoti, kaip greičiau, sugaištant mažiau laiko, išrašyti visus sprendimo variantus.

Išvardijimas turi būti sisteminis, kad nei vienas elementas nebūtų pamestas arba pakartotas kelis kartus, aiškiai išdėstytas, kad būtų paprasta ir kitiems programuotojams, ir mėgėjams suprasti, kur turi būti kiekviena parinktis. Tai yra svarbu ne tik dėl visų elementų ar variantų radimo, bet ir dėl galimybės automatizuoti procesą programuojant.

Norint atlikti tokias operacijas, reikia panaudoti tinkamas duomenų struktūras. Jei turimos tam tikros ypatybės (vėliavų dalys), kurios turi skirtingas reikšmes (spalvas), ypatybes galima sunumeruoti taip, kad ypatybių reikšmės būtų apribotos mažesniu skaičiumi (juostelės priklauso nuo skritulio spalvos, taigi skritulys yra pirmas, o juostelės atitinkamai žymimos 2 ir 3). Tam paprastai naudojama medžio struktūra. Medis – tai duomenų struktūra, vaizduojama iš viršaus į apačią, kur elementai vadinami viršūnėmis. Kai viena viršūnė išskiriama iš kitų, ji vadinama medžio šaknimi. Kiekvienoje viršūnėje judama skirtingais keliais, priklausomai nuo kitos ypatybės reikšmės. Toks procesas padeda sudaryti struktūruotą visų galimų variantų sąrašą.

Papildoma informacija apie medžius: https://inf-knyga.nmakademija.lt/lt/latest/11_medžiai.html



Nykštukai

Mažyliai

Bičiuliai

Draugai

Jauniai

Kolegos

12

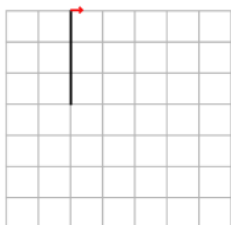
9

6

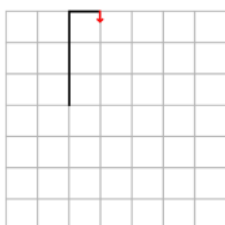
15. Piešiantis robotas

Robotas, judėdamas kvadratiname tinklelyje, piešia linijas, sukuriančias paveikslą. Kiekvienas paveikslas pavaizduotas trijų skaičių seka (tripletu). Pavyzdžiui, tripletas 3,1,5 reiškia:

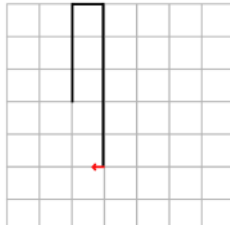
- judėti pirmyn per 3 kvadratus, tada pasukti dešinėn (1 pav.),
- judėti pirmyn per 1 kvadratą, tada pasukti dešinėn (2 pav.),
- judėti pirmyn per 5 kvadratus, tada pasukti dešinėn (3 pav.).



1 pav.



2 pav.



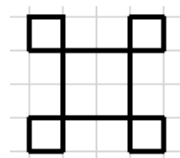
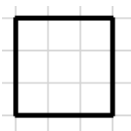
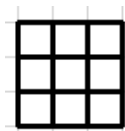
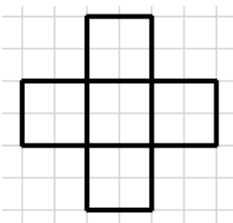
3 pav.

4

pav.

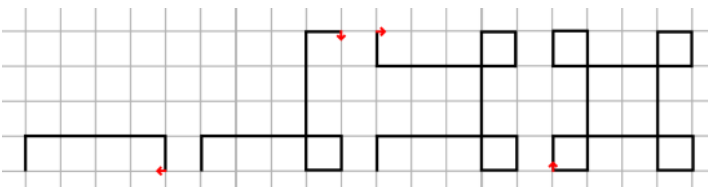
Šią tripletų seką pakartojus kelis kartus sukuriamas 4 pav.

Kurios tripletų sekos 1,4,1 2,2,3 4,2,4 3,3,3 atitinka nurodytus paveikslėlius?

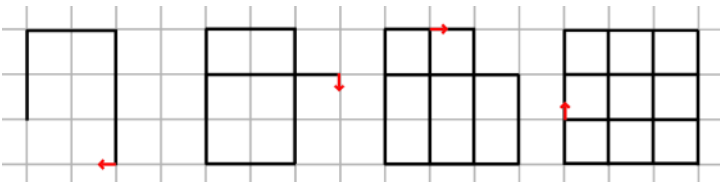


Paaškinimas

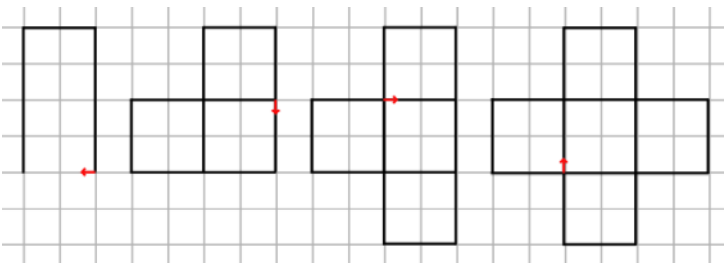
Pakartojus tripletą 1,4,1 kelis kartus gauname paveikslą:



Pakartojus tripletą 2,2,3 kelis kartus gauname paveikslą:



Pakartojus tripletą 4,2,4 kelis kartus gauname paveikslą:



Pakartojus tripletą 3,3,3 kelis kartus gaunamas kvadratas.

Tai informatika!

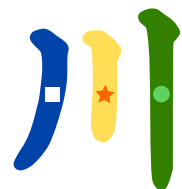
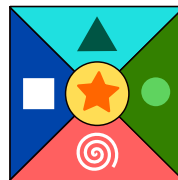
Kompiuterio programos vykdymas programų inžinerijoje yra procesas, kur kompiuteris vykdo programos instrukcijas. Kiekviena programos instrukcija – tai aprašymas specifinių veiksmų, atliekamų tam, kad būtų išspręsti uždaviniai. ([https://en.wikipedia.org/wiki/Execution_\(computing\)](https://en.wikipedia.org/wiki/Execution_(computing)))

Užduoties sprendimas apima nurodymų supratimą ir duotojo algoritmo vykdymą, kas ir yra esminė programavimo dalis. Turite sugebėti skaityti ir suprasti nurodymus bei atlikti juos pažingsniui, kad pamatytumėte, kas vyksta. Tai yra labai svarbus įgūdis kiekvienam programuotojui ir yra naudojamas kompiuterinės programos suderinime.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
		12	12			

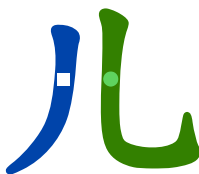
16. Spalvotas kinų hieroglifas

Kinų ženklai labai pritraukė mažojo bebro dėmesį. Jis nori suprasti šių ženklų struktūrą ir sukūrė spalvų bei simbolių paveikslą:



"川"

Struktūra: kairė – vidurys – dešinė



"儿"

Struktūra: kairė – dešinė



"吕"

Struktūra: viršus – apačia

Kuris pasirinkimas vaizduoja ženklus „三“, „二“, „八“?

A)



B)



C)



D)



Paaškinimas

Atsakymas: B.



Ženklas „三“ sudaro struktūrą: viršus – vidurys – apačia, taigi viršutinė dalis yra šviesiai mėlyna , vidurinė dalis – geltona  ir apatinė dalis – rožinė . Ženklas „二“ sudaro struktūrą: viršus – apačia, taigi kairioji dalis yra mėlyna , o dešinioji dalis – žalia .

Vadinasi, teisingas atsakymas yra B. Variantas A: ženklas „二“ yra teisingas, tačiau „三“ ir „八“ yra neteisingi. Variantas C: Visi ženklai yra neteisingi. Variantas D: ženklas „八“ yra teisingas, tačiau „二“ ir „三“ yra neteisingi.

Tai informatika!

Duomenys, jų struktūros ir vaizdavimas: duomenis galima pavaizduoti daugybe skirtingų pavidalų, pavyzdžiui, paveikslu, tekstu arba skaičiais. Šiame uždavinyje peržvelgiant duomenis, ieškoma paveikslų sekos, kuri padės išspręsti problemą. Atpažįstant šiuos paveikslus, galima sudaryti prognozes, sukurti taisykles ir išspręsti įvairias problemas.

Kinų ženklai turi struktūrą, kuri sudaryta iš "blokų statymo" dalies ir išdėstyta plokštumoje. Šie blokai paprastai pasitaiko formose: kairė – dešinė, viršus – apačia ir suformuoja dvimatį simbolį. Daugiau žiūrėkite: <http://www.littlechinesechannel.com/>

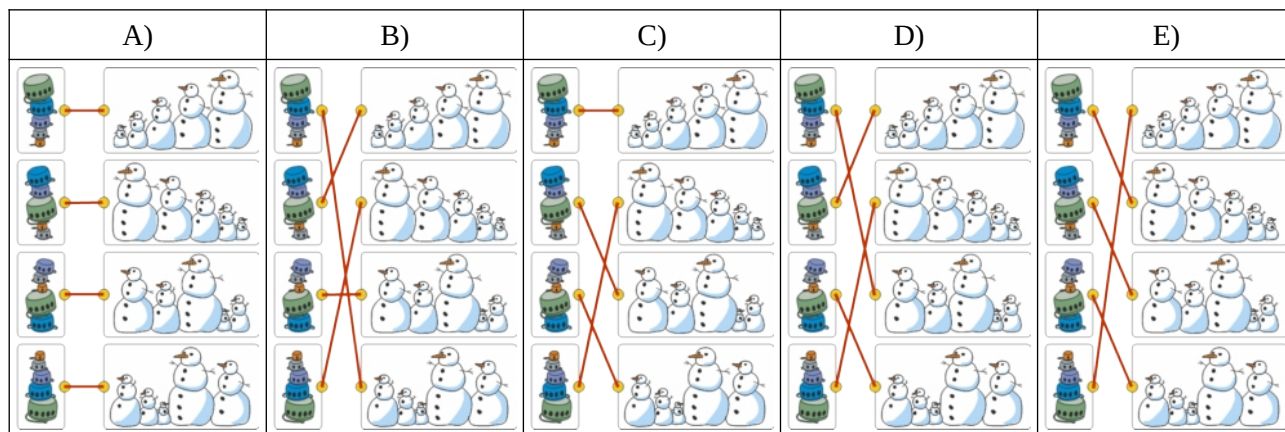
	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
		12	9			

17. Besmegenių kepurės

Penki besmegeniai stovi eilėje. Kiekvienas besmegenis pradedant iš kairės į dešinę turi gauti po skrybėlę pagal jo dydį. Skrybėlės imamos nuo viršaus po vieną.

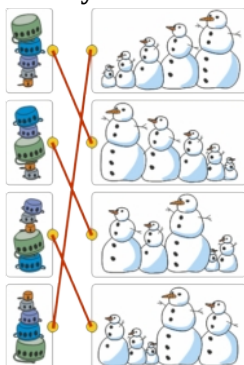


Kuri skrybėlių krūvelė kuriai besmegenių grupei priklauso?



Paaiškinimas

Atsakymas: E.



Pirmoji kepurė priklauso antrajai sniego senių sekai. Pirmasis sniego senis yra didžiausias, su 5 sagomis – atitinkamai pirmoji kepurė yra didžiausia. Antrajam sniego seniui tenka antroji kepurė ir taip toliau.

Antroji kepurė priklauso trečiajai sniego senių sekai. Pirmasis sniego senis su 4 sagomis yra antras pagal dydį ir jam tenka pirmoji kepurė (kuri taip pat yra antroji pagal dydį). Analogiškai visi kiti sniego seniai gauna tinkamas kepures.

Trečioji kepurė priklauso ketvirtajai sniego senių sekai. Čia pirmieji sekoje yra trečias pagal dydį sniego senis ir trečia pagal dydį kepurė. Analogiškai visi kiti sniego seniai gauna tinkamas kepures.

Ketvirtoji kepurė priklauso pirmajai sniego senių sekai. Kaip ir anksčiau matome, kad pirmi sekoje sniego senis ir kepurė yra mažiausi pagal dydį, antri eilėje yra antri pagal dydį. Analogiškai visi kiti sniego seniai gauna tinkamas kepures.

Atsakymas A neteisingas, nes pirmasis sekoje, mažiausias sniego senis gauna didžiausią kepurę.

Atsakymas B neteisingas, nes pirmasis sekoje, mažiausias sniego senis gauna antrąją pagal dydį kepurę.

Atsakymas C neteisingas, nes pirmasis sekoje, mažiausias sniego senis gauna didžiausią kepurę.

Atsakymas D neteisingas, nes pirmasis sekoje, mažiausias sniego senis gauna antrą pagal dydį kepurę.

Tai informatika!

Šiame uždavinyje kiekvieno sniego senio suporavimas su kepure, nekeičiant esamo išsidėstymo tvarkos, vadinamas „atvaizdavimu“. Kiekvienai sekai yra apibrėžtas pirmasis elementas (aukščiausiai esanti kepurė ir kairiausias sniego senis), jie turi sutapti. Toliau kiti elementai poruojami atitinkamai iš eilės. Kepurių krūvelė čia pateikta kaip stekas. Tai yra tam tikras būdas, kaip duomenys laikomi kompiuteryje (abstrakti duomenų struktūra stekas). Su stekais galima daryti keletą veiksmų (operacijų):

- Galima pašalinti aukščiausiai esantį elementą
- Galima įterpti naują elementą į steką, padedant viršuje

Nežinome, kaip buvo sudarytos sekos, jos galėjo būti sudarytos ir iš viršaus į apačią (t. y. Pirmoji kepurė krūvelėje buvo padėta pirmoji, antroje vietoje esanti, buvo padėta antroji ir taip toliau), tai būtų vadinama eile, jai galimos šios operacijos:

- Galima pašalinti priekinį elementą
- Galima įterpti naują elementą eilės gale.

Tą patį galima pasakyti ir apie sniego senius: jie formuoja eilę (kairysis buvo pirmas, dešinysis – paskutinis, ir eilę paliktų paskutinis. Gali būti ir kad kairysis sniego senis eilėje atsirado paskutinis, o jį paliktų pirmas.

Svetainės:

[https://en.wikipedia.org/wiki/Stack_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type))

[https://en.wikipedia.org/wiki/Queue_\(abstract_data_type\)](https://en.wikipedia.org/wiki/Queue_(abstract_data_type))



Nykštukai

Mažyliai

Bičiuliai

Draugai

Jauniai

Kolegos

9

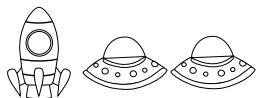
6

18. Kelionės kosmose

Astronautai gali keliauti tarp planetų raketa (🚀) arba erdvėlaiviu (🛸), kaip parodėta paveikslėlyje dešinėje:

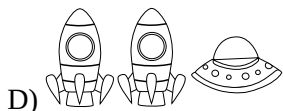
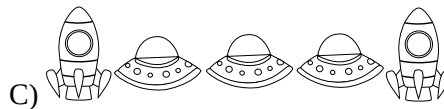
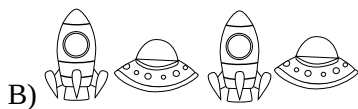
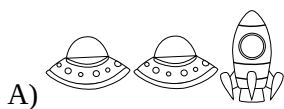
Pavyzdžiui, astronautas yra Veneroje (🌋) ir nori nuvykti į Saturną (♄). Jis gali pirmiausiai raketa nuvykti į Jupiterį (♃).

Tada erdvėlaiviu keliauti į Neptūną (♆) ir galiausiai erdvėlaiviu pasiekti savo kelionės tikslą. Astronautai šį kelionės maršrutą pavaizduoja taip:



Astronautas Tinas yra Neptūne (♆) ir nori grįžti namo į Žemę (🌍). Kosmoso kelionių agentūra pateikė šiuos kelionių maršruto pasiūlymus.

Kuriuo iš pateiktų maršrutų Tinui nepavyktų grįžti į Žemę?



Paaškinimas

Atsakymas: B.

Jei Tinas pasirinktų šį maršrutą, galiausiai jis vėl grįžtų į Neptūną. Pirmiausiai, nuskristų raketa į Marsą, tada erdvėlaiviu į Neptūną, tada dar kartą raketa į Marsą ir galiausiai atgal į Neptūną erdvėlaiviu.

Tai informatika!

Užduotis remiasi kompiuterinių sistemų principu, kuris paaiškina, kas slepiasi po bankomatais ir kavos aparatais, ir kaip kompiuteriai vykdo komandų sekas. Informatikoje mes tai vadiname Deterministiniais baigtiniais automatais (DBA). Kosmoso žemėlapis šioje užduotyje yra DBA pavyzdys.

Deterministiniai baigtiniai automatai turi:

- įvesties abėcėlę (mūsų atveju, raketa ir erdvėlaivis)
- baigtinį būsenų skaičių (planetas)
- pradinę būseną (kur astronautas pradeda kelionę)
- galutinių būsenų rinkinį (kur astronautas turėtų baigti kelionę; mūsų atveju – Žemė)
- perėjimus nuo vienos būsenos prie kitos (mūsų atveju galimi skrydžiai tarp dviejų planetų)

DBA priima abėcėlės ženklų seką tik tokiu atveju, jei ji veda iš pradinės būsenos į galutinę. Pavyzdžiui, DBA gali modeliuoti programinę įrangą, kuri nusprendžia, ar vartotojo įvesti duomenys (pvz., el. pašto adresas) yra tinkami.



Nykštukai

Mažyliai

Bičiuliai

Draugai

Jauniai

Kolegos

12

9

6

19. Vyresniųjų bebrų pranešimas

Bebrė Karolina užtvankoje rado labai seną medį. Atidžiai jį apžiūrėjusi, pastebėjo medyje išpjaustytus mistinius ženklus. Karolina samprotauja, kad tai panašu į kodų lentelę dar nuo tų laikų, kai vyresnysis bebras dar gyveno užtvankoje.

	I	II	III	III	○	○	○	○	○
	A	B	C	D	E	F	G	H	I
	J	K	L	M	N	O	P	Q	R
	S	T	U	V	W	X	Y	Z	

Po ilgo lentelės tyrinėjimo Karolina suprato, kaip naudotis lentele: iš atitinkamose eilutėse ir stulpeliuose esančių ženklų kombinacijos sudaromas naujas simbolis. Pavyzdžiui, raidė H užkoduota taip:

	I	II	III	III	○	○	○	○	○
	A	B	C	D	E	F	G	H	I
	J	K	L	M	N	O	P	Q	R
	S	T	U	V	W	X	Y	Z	

+ =

Karolina prisimena, kad matė tokius simbolius ir kitoje bebrų užtvankos vietoje. Nuėjusi ten, iš tikrųjų patvirtino savo mintį. Štai kas buvo parašyta ant kito medžio:



Kuris vyresniųjų bebrų pranešimas užkoduotas?

- A) LOVEWATER
- B) SLEEPDAYS
- C) LOVEMYSUN
- D) CAREFORME

Paiškinimas

Atsakymas: A) LOVEWATER

Pirmiausia patikriname, ar pranešimo ilgis gali pateikti užuominą. Matome, kad visi atsakymai yra vienodo ilgio.

Iškodavę pirmąją raidę remdamiesi kodų lentele, išsiaiškinome, kad  yra „L“. Tai leidžia suprasti, kad atsakymas negali būti nei B), nei D), o atsakymai A) ir C) gali pretenduoti į teisingo atsakymo vaidmenį.

Ieškome skirtumų tarp A) ir C) atsakymų. Matome, kad nuo penktos raidės pranešimai skiriasi. Iškodavę

penktąjį ženklą, randame, kad  yra „W“. Dabar žinome, kad turėtų būti teisingas atsakymas A).

Norėdami tuo įsitikinti, iškoduojame dar vieną ženklą, pavyzdžiui paskutinį. Išsiaiškiname, kad  yra „R“. Tai patvirtina mūsų prielaidą. Dabar esame tikri, kad atsakymas A) yra teisingas.

Tai informatika!

Duomenų apsauga yra svarbi problema šių dienų visuomenėje. Vienas iš metodų apsaugoti duomenis nuo neįgaliotų asmenų yra slaptas šifravimas. Kriptografija prasidėjo prieš 3500 metų. Vienas iš pirmųjų metodų buvo raidžių sukeitimas vietomis. Šiame pavyzdyje abėcėlės raidės yra pakeičiamos naujais ženklais taip, kad būtų lengva prisiminti šifravimo būdą.

Jei šiame uždavinyje nebūtų jokios šifravimą aprašančios lentelės, o turėtume tik ženklus, tuomet piešinius iššifruoti būtų žymiai sudėtingiau. Šifrų analitikai, kurie bando atpažinti kodus, naudoja tokias technikas, kaip dažnių analizė ir šablonų atpažinimas.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
			12	9	6	

20. Automobilių parkavimas

Stovėjimo aikštelėje automobiliai gali būti statomi stovėjimo vietose arba priešais šias stovėjimo vietas, kaip parodyta žemiau pateiktame paveiksle. Automobiliai, kurie yra pastatyti priešais stovėjimo vietas, gali būti patrauktas priekin arba atgal tuo atveju, jeigu jie yra užstatę automobilį, kuriuo norima išvažiuoti iš stovėjimo aikštelės.

Pavyzdžiui, paveiksle pateiktame žemiau, automobilis A nėra užstatytas ir juo galima išvažiuoti iš stovėjimo aikštelės. Automobilis L yra užstatytas automobilio M, tad šis automobilis privalo būti perstatytas, kad automobiliu L būtų galima išvažiuoti iš stovėjimo aikštelės.



Kuriuo automobiliu norint išvykti, privalo būti perstatyti kiti du automobiliai?

Paaiškinimas

Atsakymas: Automobilis I.

Šis automobilis yra užstatytas automobilio N. Kadangi nėra pakankamai vietos perstatyti automobilį N, automobiliu O reikia pavažiuoti į kairę arba automobiliu M į dešinę. Tuomet atsiranda vietos automobilį N perstatyti taip, kad automobilis I galėtų išvažiuoti.

Paveiksle nėra jokio kito automobilio, kuriuo norint išvažiuoti reikėtų perstatyti kitus du automobilius: automobiliais A, D, E, J ir Q galima palikti stovėjimo vietas tuojau pat. Automobiliais B, C, F, G, H, K ir L galima palikti stovėjimo vietas, jeigu vienas iš automobilių P, O, N arba M bus perstatytas.

Tai informatika!

Ši užduotis apima du svarbius aspektus, kurie yra glaudžiai susiję su informatika:

- 1) Išsamios paieškos algoritmas, skirtas tikrinti visus galimus automobilius ir ieškoti, kurie iš jų turi reikiamas savybes, t.y. kuriuo automobiliu galima išvažiuoti iš stovėjimo vietos po to, kai kiti du automobiliai bus perstatyti.

https://en.wikipedia.org/wiki/Brute-force_search

- 2) Automatiniai parkavimo algoritmai, kurie tampa vis labiau ir labiau prieinami šių dienų automobiliuose:

https://en.wikipedia.org/wiki/Automatic_parking

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
			12	6		

21. Garsenybė

Socialinio tinklo „TeeniGram“ nariai gali būti vienas kito sekėjais. „TeeniGram“ grupės – tai narių rinkiniai. Grupės narys gali būti garsenybė, jeigu:

- Jis yra sekamas kiekvieno grupės nario;
- Grupėje jis neseka nieko kito.

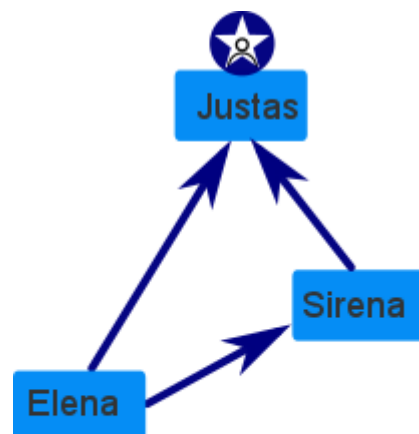
Vienos „TeeniGram“ grupės nariai yra Elena, Sirena ir Justas.

- Elena seka Justą ir Sireną
- Sirena seka Justą
- Justas nieko neseka

Justas yra grupės garsenybė.

Kitos „TeeniGram“ grupės nariai yra: Alanas, Dana, Fausta, Gražina ir Rokas.

- Alanas seka Daną ir Gražiną.
- Dana seka Gražiną ir Roką.
- Fausta seka Alaną, Gražiną ir Roką.
- Rokas seka Alaną ir Gražiną.



Pavyzdys

Ar šioje grupėje yra garsenybė?

- Taip, Alanas yra garsenybė.
- Taip, Fausta ir Rokas yra garsenybės.
- Taip, Gražina yra garsenybė.
- Ne, šioje grupėje garsenybių nėra.

Paiškinimas

Atsakymas: C.

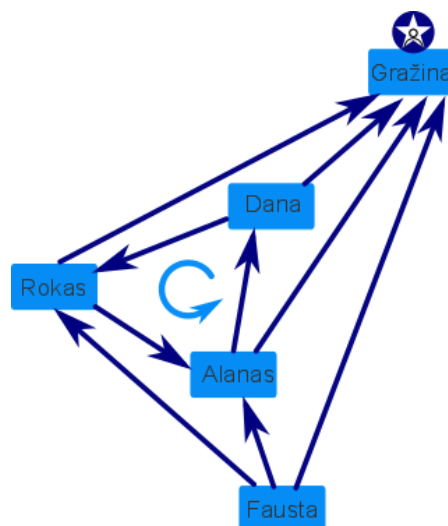
Gražina yra garsenybė šioje grupėje. Ji atitinka abi sąlygas: Alanas, Dana, Fausta ir Rokas ją seka, ji nieko kito neseka.

A: neteisingas, nes Alanas seka du grupės narius: Daną ir Gražiną.

B: neteisingas, nes Fausta ir Rokas seka kitus grupės narius.

Nenurodyta taisyklė, kad grupėje negali būti daugiau negu viena garsenybė.

D: neteisingas, nes Gražina yra garsenybė.



Tai informatika!

Socialiniai tinklai paremti ryšiais tarp narių. Vienas atvejis – vienas narys seka kitą. Narys A seka narį B, bet narys B neseka nario A. Tai vadinsime vienakrypčiu ryšiu. Kitas atvejis – du žmonės yra draugai. Toks ryšys yra simetrinis ir neturi krypties. Narys A yra nario B draugas ir narys B yra nario A draugas. Tai vadinsime bekrypčiu ryšiu.

Abu ryšiai gali būti pavaizduoti grafu. Grafų uždavinių sprendimui naudojami algoritmai.

Realūs socialiniai tinklai formuoja labai didelius grafus ir kompanijos, realizuojančios šiuos tinklus, yra suinteresuotos rasti šiuose grafuose kokias nors pasikartojančias struktūras, pavyzdžiui, surasti grupės garsenybę. Kai socialiniame tinkle nėra grupių, rasti narius, kurie turi daug sekėjų ir neseka kitų narių, yra įdomi užduotis, nes šie nariai gali būti laikomi lyderiais ar nuomonės formuotojais (angl. influencer).

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
			9	9	6	

22. Rangoli menas

Rangoli – tai meno forma, kai naudojant spalvotas medžiagas ant grindų kuriami ornamentai.

Rita turi trijų rūšių plyteles: 8 violetinius trikampius, 4 žalius kvadratus ir 6 juodus trikampius. Visos vienos rūšies plytelės yra vienodo dydžio.

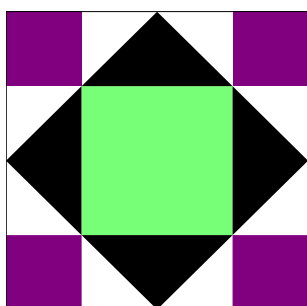
Figūra			
Kiekis	8	4	6

Rita nori sukurti Rangoli ornamentą naudodama tik šias plyteles.

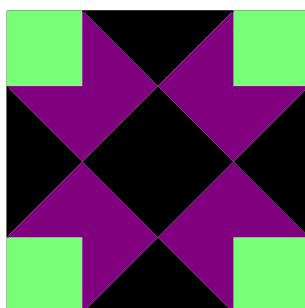
Ji neprivalo panaudoti visų turimų plytelių ar uždengti visų grindų.

Kuriuos Rangoli ornamentus gali sukurti Rita iš turimų plytelių?

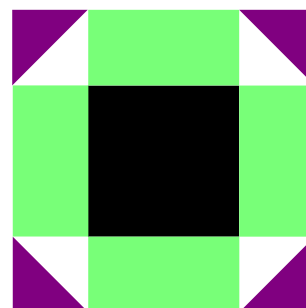
A



B



C

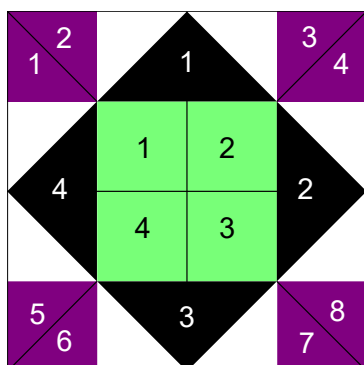


- A) A
- B) B
- C) C
- D) Visus 3 ornamentus

Paiškinimas

Atsakymas: A ornamentas.

Uždavinio sąlygoje duota, kiek Rita turi plytelių, tad reikia išsiaiškinti, kaip galima sudėlioti plyteles pagal Rangoli ornamentus ir suskaičiuoti jų skaičių. Pateiktas paveikslėlis iliustruoja vieną iš ornamento A plytelių sudėliojimo būdų. Kiekvienos rūšies plytelės yra sunumeruotos.



Lentelėje pateiktas kiekvienam ornamentui reikalingas plytelių skaičius. Paskutiniame stulpelyje nurodoma, ar Rita turi pakankamai plytelių kiekvienam ornamentui sukurti.

Ornamentas	Reikalingos plytelės	Ar Rita gali sukurti šį ornamentą?
A	8 violetiniai trikampiai 4 žali kvadratai 4 juodi trikampiai	Taip, ji turi užtektinai kiekvienos rūšies plytelių.
B	12 violetinių trikampių 4 žali kvadratai 6 juodi trikampiai	Ne, ji turi tik 8 violetinius trikampius.
C	4 violetiniai trikampiai 8 žali kvadratai 4 juodi trikampiai	Ne, ji turi tik 4 žalius kvadratus.

Taigi vienintelis pasirinkimas yra ornamentas A.

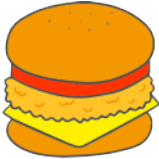
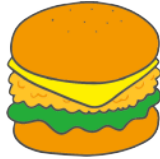
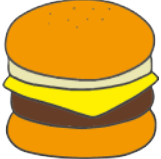
Tai informatika!

Siekiant išspręsti šią dėlionę, kiekviename ornamente reikia rasti visų rūšių plyteles ir suskaičiuoti jų skaičių. Šis uždavinys susideda iš skaidymo ir šablonų paieškos. Informatikoje šablonų paieška labai svarbus uždavinys. Kai ieškome žodžio dokumente, dokumento kompiuteryje ar kažko internete, tada yra vykdoma šablonų paieška. Pavyzdžiui, šablonas gali būti žodžio pradžia, tada paieškos sistema ar dokumentų naršyklė ieškos visų dokumentų, prasidedančių tų pačių raidžių junginiu.

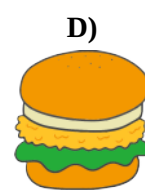
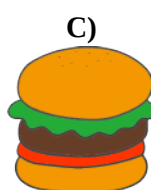
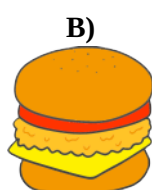
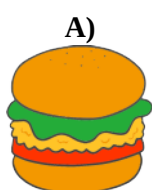
	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
			12	9	6	

23. Mėsainių ingredientai

Mėsainių kavinė “BurgerKingWay” naudoja šešias rūšis ingredientų: A, B, C, D, E ir F. Lentelėje pavaizduoti mėsainiai ir jų ingredientai išdėstant juos bet kaip.

Mėsainis				
Ingredientai	C, F	A, B, E	B, E, F	B, C, D

Kuris mėsainis turi A, E ir F ingredientus?



Paaškinimas



Atsakymas: A)

Lyginami mėsainiai		Bendra raidė	Bendras ingredientas
 C,F	 B,E,F	F	žalia „salota“
 C,F	 B,C,D	C	ruda „mėsa“
 A,B,E	 B,C,D	B	geltonas „sūris“
 B,E,F	 A,B,E	E	oranžinė „vištiena“

Taip sužinome kiekvieno mėsainio ingredientus, išskyrus vieną. Taigi lieka vienas nežinomas ingredientas.

Mėsainis	Nežinoma raidė	Naujas ingredientas
 A,B,E	A	raudonas „pomidoras“
 B,C,D	D	balta „mocarela“

Todėl mėsainis, turintis ingredientus A, E ir F, yra šis , tai yra A variantas.

Tai informatika!

Logika yra daugelio informatikos uždavinių pagrindas. Šis uždavinys gali būti išspręstas pasitelkiant loginį mąstymą. Šiuo atveju, nagrinėjant žinomą informaciją, identifikavus bendrus mėsainių ingredientus, galima padaryti išvadą apie nežinomus elementus.

Aibių teorijoje aibe vadinama grupė elementų (pavyzdžiui, skirtingi ingredientai). Su aibėmis atliekamos operacijos, pavyzdžiui, sankirta ir skirtumas:

- ingredientų aibių sankirtos rezultatas yra bendri elementai.
- visų ingredientų ir žinomų ingredientų aibių skirtumo rezultatas yra visi nežinomi mėsainio ingredientai (pavyzdžiui, atmetus geltoną „sūrį“ ir oranžinę „vištieną“ lieka raudonas „pomidoras“).

Jeigu sankirtos arba skirtumo operacijos rezultatas yra tik vienas elementas, tai galima nesunkiai nustatyti jo raidę.



Nykštukai

Mažyliai

Bičiuliai

Draugai

Jauniai

Kolegos

9

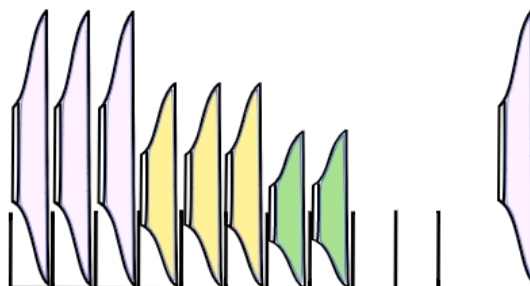
6

24. Lėkštės

Tvarkingoji Bebrė visuomet susideda lėkštės tokia tvarka, kaip matome pateiktame paveikslėlyje: pirmiausia didelės, tada vidutinio dydžio ir galiausiai mažos. Dabar jai reikia į indų džiovyklą padėti dar vieną didelę lėkštę.

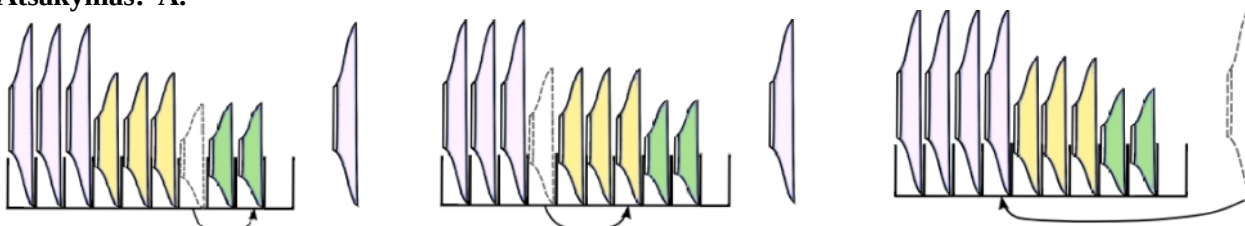
Kiek mažiausiai lėkščių (įskaitant ir naująją) jai reikės perkelti, norint surikiuoti lėkštes tvarkingai?

- A) 3
- B) 4
- C) 5
- D) 6

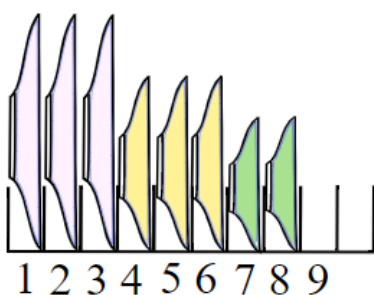


Paiškinimas

Atsakymas: A.



Užduotis gali būti atlikta perkeltiant tris lėkštes. Galimas sprendimas pavaizduotas paveikslėliais.



Norėdami rasti sprendimą, pirmiausiai turime pažiūrėti į pradinį lėkščių išsidėstymą ir surasti tinkamą vietą naujai lėkštei. Kad visos didelės lėkštės būtų šalia viena kitos, naujoji lėkštė turi būti įdėta į 4-tą vietą. Tačiau šią vietą jau užima vidutinio dydžio lėkštė. Turime rasti naują vietą šiai vidutinio dydžio lėkštei. Naujoji vieta turėtų būti 7-ta, kad visos vidutinio dydžio lėkštės būtų greta. Septintoje vietoje yra maža lėkštė, vadinasi, mums reikės perkelti ją iš 7-os į 9-tą vietą.

Tai informatika!

Labai tiesmukas šio uždavinio sprendimas yra perkelti visas vidutines ir mažas lėkštes per vieną vietą į dešinę. Tai sukurtų laisvą vietą būtent ten, kur mums reikia. Pagal užduoties sąlygą tai reikėtų, kad turime perkelti 6 lėkštes: pirmiausia perkeltume lėkštę iš 8-tos vietos į 9-tą, tuomet tolesnę ir taip toliau. Taip perkėlinėtume lėkštes, kol rastume didesnę ar tokio pat dydžio lėkštę kaip ta, kurią norime įdėti. Tik tuomet įdėtume naują lėkštę.

Šis tiesmukas metodas (algoritmas) gali būti naudojamas ir kompiuteriuose. Pagrindinis skirtumas yra, kad kompiuteriai gali apdoroti daug daugiau duomenų (iki milijardų elementų). Visgi tai yra pernelyg lėtas procesas net ir kompiuteriui, tad informatikos specialistai turi ieškoti geresnių sprendimų (algoritmų), panašių į čia mūsų pateiktą.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
			12	9	6	

25. Alergiški bebrai

Kai kurie bebrai negali valgyti tam tikros rūšies medžių, nes jiems yra alergiški. Bebrė Jonė ruošia vakarėliui patiekalus iš medžių ir nori pagaminti pakankamą įvairovę, kad visi bebrai rastų ko paskanauti. Kiekvienas patiekalas yra pagamintas iš vienos rūšies medžio ir bebrai noriai jais dalijasi. Jonė turi sąrašą, kuriame nurodoma, kokių rūšių medžius gali valgyti vakarėlio dalyviai.

Vardas	Medis
Jonė	Gluosnis, ąžuolas, uosis, klevas
Benas	Gluosnis, ąžuolas, drebulė
Cecilija	Ąžuolas
Domas	Uosis, beržas
Ema	Gluosnis, klevas, beržas
Fredas	Ąžuolas, uosis
Goda	Drebulė, klevas



Jonė nenorėtų gaminti patiekalų iš kiekvieno medžio.

Kiek mažiausiai patiekalų Jonė galėtų paruošti vakarėliui, kad visi liktų pavalgę?

A) 1 B) 2 C) 3 D) 4 E) 5 F) 6

Paaškinimas

Atsakymas: C.

Akivaizdu, Jonė būtinai turi pagaminti patiekalą iš ąžuolo Cecilijai. Jonė, Benas ir Fredas taip pat jį galės valgyti. Likę bebrai neturi bendrų tinkamų valgyti medžių, todėl reikės dar bent dviejų patiekalų. Tinkamų medžių poros: uosis ir klevas, drebulė ir beržas arba klevas ir beržas.

Tai informatika!

Tai yra klasikinis digrafo pavyzdys. (https://en.wikipedia.org/wiki/Bipartite_graph)

Čia pirmasis grafo perskyrimas - bebrai, antrasis – medžių rūšys.

Šis uždavinys algoritmų teorijoje žinomas kaip traukinių stoties optimizacijos problema. Pagal traukinių ir jų sustojimų sąrašą reikia surasti mažiausią galimą stočių rinkinį, kad kiekvienas traukinys bent kartą aplankytų bent vieną iš pasirinktų stočių.

Jis taip pat žinomas kaip dominuojančiosios aibės paieškos uždavinys. Šiuo atveju turime 6 aibes, atitinkančias kiekvieną patiekalą. Kiekvieną rinkinį sudaro bebrai, valgantys tokį patiekalą. Užduotis yra atrinkti mažiausią skaičių aibių (patiekalų), tokių, kad jų sąjunga (visi bebrai, valgantys tuos patiekalus) turėtų visus elementus (bebrus). Kitaip tariant, užduotis yra „surinkti“ visus elementus turint mažiausią skaičių aibių.

Šiame uždavinyje aibes ir elementus atvaizdavome kaip patiekalus ir bebrus, tačiau galime rasti daugybę panašių pavyzdžių ir kitokiomis temomis.

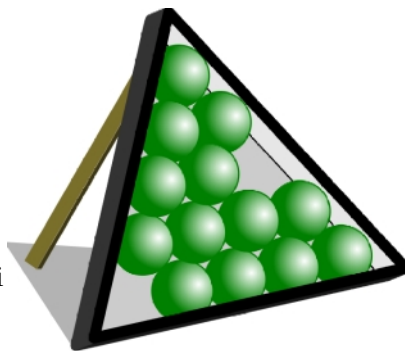
Ši problema laikoma viena sudėtingiausių duomenų moksle, vadinama NP-sudėtingumo uždaviniu. Tokiems uždaviniams spręsti nėra žinoma optimalių algoritmų. Vienintelis sprendimas - išbandyti įvairius variantus. Tačiau netgi esant nedideliui aibių skaičiui, galimų variantų skaičius gali būti labai didelis. Įsivaizduokite, jei turime 240 aibių, tai galimų variantų yra daugiau nei atomų visatoje!

Tad kaip galime išspręsti tokį uždavinį? Šiuo atveju, turėjome labai mažą aibių (patiekalų) skaičių. Jis buvo dar paprastesnis, nes Cecilijai buvo tinkama tik viena rūšis – tai dar sumažino variantų skaičių. Galiausiai rėmėms paprasta logika, kuri tiko, būtent, šiam konkrečiam uždaviniui. Tačiau vargu, ar tai tiks kitam uždaviniui. https://en.wikipedia.org/wiki/Set_cover_problem

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
				12	9	6

26. Rutuliai

Trikampėje dėžutėje 13 rutulių sudėti taip, kaip pavaizduota paveiksle. Jei pakeltume dėžutės viršutinį kampą, kai kurie rutuliai nuriudėtų žemyn, kadangi dėžutėje yra tuščių vietų.



Sakome, kad rutulys yra nestabilus – gali nuriudėti žemyn, – jei teisingas bet kuris iš šių teiginių:

- Žemiau rutulio kairėje arba dešinėje yra bent vienas tarpas.
- Žemiau rutulio kairėje arba dešinėje yra bent vienas rutulys, kuris yra nestabilus.

Kiek pavaizduotoje dėžutėje rutulių, kurie yra stabilūs?

Paaškinimas

Atsakymas: 8.

Vienas iš būdų rasti atsakymą yra toks:

- Pradėkime nuo apatinės trikampio eilės ir sužymėkime visus joje esančius rutulius.
- Pereikime viena eile aukščiau.
- Pažymėkime visus šios eilės rutulius, žemiau kurių yra du pažymėti rutuliai (žemiau kairėje ir žemiau dešinėje).
- Kartokime 2 ir 3 žingsnius, kol yra eilių dėžutėje.
- Visi pažymėti rutuliai ir yra tie rutuliai, kurie yra stabilūs.

Procesas pavaizduotas paveiksle. (Nuo ketvirtos eilės, skaičiuojant iš apačios, nei vienas rutulys negali būti žymimas.)



Tai informatika!

Yra dvi sąlygos, kurios apibrėžia nestabilių rutulį. Pirmoji gali būti patikrinta tiesiogiai. Antrajai sąlygai patikrinti reikia žinoti, ar žemesnėje eilėje yra nestabilių rutulių. Kadangi tai primena ciklo sąlygų apibrėžimą, galime pradėti iš apatinės eilės. Apatinės eilės rutuliai po savimi neturi rutulių, tad jie visi yra stabilūs. Kai sužymėjome rutulius, galime pereiti prie eilės virš jų. Taip sužymime visus rutulius nuo apatinės eilės iki pat viršutinės.

Dažnai rinkiniai komandų, kurias pateikiame kompiuteriui, yra labai panašūs į žmonėms skirtas instrukcijas. Programuotojai kartais naudoja trumpinius, kuriuos paprasčiau „suprasti“ kompiuteriui, nors tokie trumpiniai mums gali būti ir sunkiau suvokiami. Šioje užduotyje naudojamas rekursijos metodas. Rekursija paprastai apibrėžiamą sąvoką naudoja kaip paties apibrėžimo dalį: apibrėžiant rutulį, kuris yra nestabilus, naudojame stabilumo sąvoką, tarsi mes jau žinome, ką ji reiškia.

Dar vienas rekursyvos apibrėžties pavyzdys. Jei paprašytų paaškinti, kas yra Jūsų protėviai, atsakytumėte: „mano močiutės mama ir tėvas, mano senelio mama ir tėvas, kitos močiutės mama ir tėvas, kito senelio mama ir tėvas, jų tėvai ir taip toliau“. Kompiuteriams sunku apdoroti neaiškiai apibrėžtas sąvokas, tokias kaip „ir taip toliau“. Taigi programuotojas šiuo atveju naudotų rekursiją: „Jūsų protėviai yra Jūsų senelių tėvai, jų tėvai, jų seneliai ir jų protėviai“. Žmogui tokia apibrėžtis gali pasirodyti keista, bet kompiuteriui ji visiškai aiški ir vienareikšmiška.

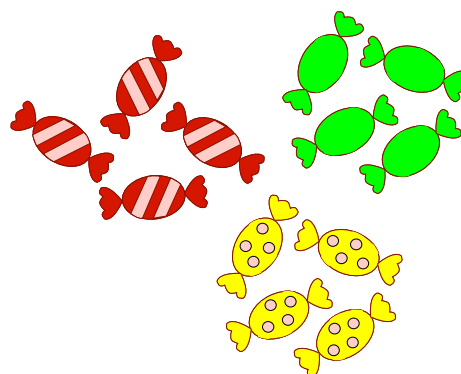
	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
				12	9	6

27. Saldainių maišelis

Paulina turi maišą saldainių. Maiše yra 4 žali, 4 geltoni ir 4 raudoni saldainiai.

Ji žaidžia su savo draugu ir turi keletą taisyklių: draugas turi tris bandymus ištraukti saldainį iš maišo.

- Kiekvieną kartą išimdamas žalią saldainį, jis turi jį įdėti į dubenį ir iš maišo ištraukti dar vieną saldainį (neišnaudodamas kito bandymo).
- Jei jis išima geltoną saldainį, jį iškart suvalgo, neįdėdamas į dubenį.
- Jei jis išima raudoną, jis turi jį nedelsdamas įdėti į dubenį.



Koks didžiausias saldainių skaičius gali būti dubenyje po trečiojo bandymo?

Paiškinimas

Atsakymas: 7.

Vienu bandymu negalime ištraukti daugiau nei vieną raudoną saldainį, taigi jų dubenyje bus ne daugiau nei 3, o geltonų saldainių dubenyje būti negali (jie iš karto suvalgomi), taigi daugiau nei 7 saldainių dubenyje būti negali.

Geriausias įmanomas rezultatas yra toks:

1-as bandymas: žalias + žalias + žalias + žalias + raudonas = 5

2-as bandymas: raudonas = 1

3-as bandymas: raudonas = 1

Iš viso: $5 + 1 + 1 = 7$

Tai informatika!

Vykdydamas programą, kompiuteris dažnai turi priimti sprendimus. Šie sprendimai gali priklausyti nuo sąlygų, į kurias atsižvelgiama programuojant. Vykdamas šią užduotį sprendimai priimami atsižvelgiant į saldainių spalvą.

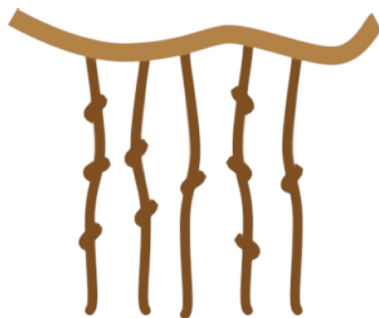
Šioje užduotyje naudojamos sąlygos: (saldainio spalva yra žalia), (saldainio spalva yra geltona) ir (saldainio spalva yra raudona). Kiekviena iš šių sąlygų gali būti teisinga arba klaidinga. Atsižvelgiant į tai, kuri vertė yra teisinga, konkretus veiksmas gali būti atliktas arba ne. Kompiuterių programos naudoja sąlygomis, kad sektų skirtingais programų keliais, atsižvelgiant į sąlygų tikrąją vertę.

Gerai žinomos valdymo struktūros, kurios naudojamos sąlygoms, yra šakojimo komanda ir ciklas.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
				12	9	6

28. Kipu

Karalienė riša mazgus ant kabančių virvių (vadinamų kipu), norėdama pranešti naujienas savo karalystei. Pavyzdžiui, žemiau pateiktos kipu reiškia „Švęskime“.



Pranešimo reikšmė priklauso nuo virvių išsidėstymo ir mazgų skaičiaus ant kiekvienos virvės.

Ant kiekvienos virvės gali būti 0, 1, 2 ar 3 mazgai.

Karalienė turi 50 skirtingų pranešimų, kuriuos gali perduoti virvėmis.

Kiek mažiausiai virvių reikia karalienei?

- A) 2
- B) 3
- C) 4
- D) 5

Paaiškinimas

Atsakymas: B.

Jei būtų tik viena virvė, tuomet būtų galima perduoti 4 skirtingus pranešimus, nes ant virvės gali būti 0, 1, 2 arba 3 mazgai. Dvejomis virvėmis, kurių kiekviena gali turėti 4 mazgų variantus, būtų galima perduoti $4 \times 4 = 16$ skirtingų pranešimų. Tačiau to dar nepakanka perduoti visus karalienės pranešimus. Jei pridėdama trečia virvė, galima perduoti $4 \times 4 \times 4 = 64$ skirtingus pranešimus. Kadangi 64 yra daugiau nei 50, trijų virvių pakanka, kad perduoti visus karalienės pranešimus.

Tai informatika!

Ši užduotis yra pozicinės sistemos pavyzdys. Sudarant kipu, svarbi yra virvės padėtis ir mazgų skaičius ant kiekvienos virvės. Kadangi kiekvieną virvę galima surišti keturiais būdais, kipu yra ketvirtainės sistemos pavyzdys.

Kai žmonės atlieka skaičiavimus, dažniausiai naudojama dešimtainė sistema. Dešimtainiai skaitmenys (nuo 0 iki 9) yra kaip mazgai ant kipu, o skaitmenų pozicija (atitinkanti 10 keliant laipsniais) yra kaip virvių padėtis kipu. Pavyzdžiui, $427 = 4 \times 100 + 2 \times 10 + 7$, su trimis dešimtainiais skaitmenimis galime sudaryti $10 \times 10 \times 10 = 1000$ skirtingų natūraliųjų skaičių (0 – 999).

Kompiuteriai naudoja dvejetainę sistemą. Tokioje sistemoje kiekvienas skaitmuo (kompiuterinėse sistemose vadinama bitais) gali turėti vieną iš dviejų reikšmių: 0 arba 1. Visi kompiuterio duomenys ir nurodymai yra saugomi remiantis šiuo principu.



Nykštukai

Mažyliai

Bičiuliai

Draugai

Jauniai

Kolegos

12

9

6

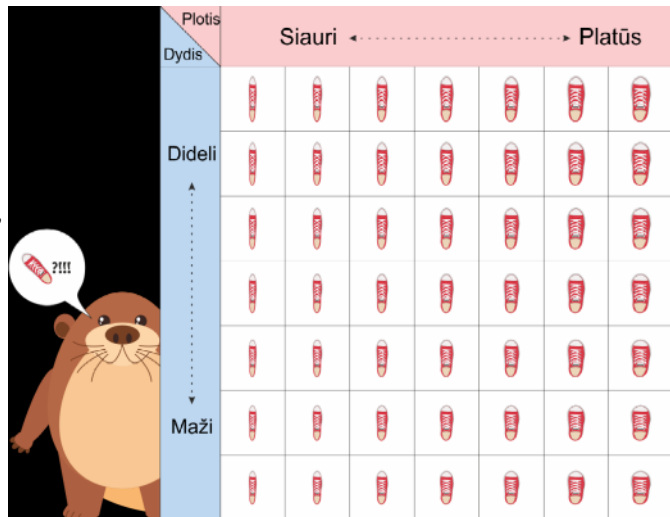
29. Batų pirkimas

Bebras nuėjo į parduotuvę, kad nusipirktų batų porą. Vitrinoje jis pamatė keletą batų, išdėliotų taip, kaip parodyta paveikslėlyje. Batai buvo išdėlioti didėjimo tvarka pagal dydį ir plotį. Siauriausi ir mažiausi buvo padėti kairiajame apatiniame kampe, o didžiausi ir plačiausi – dešiniajame viršutiniame kampe. Visi batai buvo skirtingo dydžio ir pločio.

Bebras buvo užmaršus, ir neprisiminė savo dydžio, todėl turėjo matuoti batus tol, kol rado tinkamus. Tinkami batai – tai tinkamo dydžio ir pločio batai.

Bebras naudojo metodą, užtikrinančią, kad tinkančius batus jis randa per „n“ matavimų.

Kokia yra mažiausia galima matavimo kiekio „n“ reikšmė?

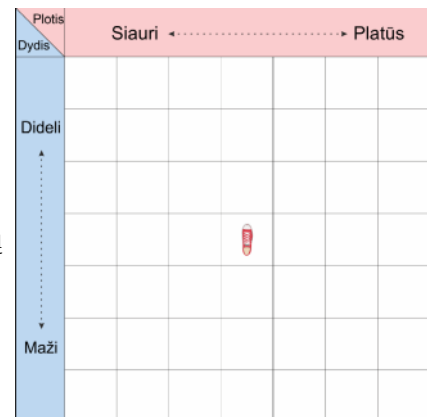


Paaškinimas

Atsakymas: 2.

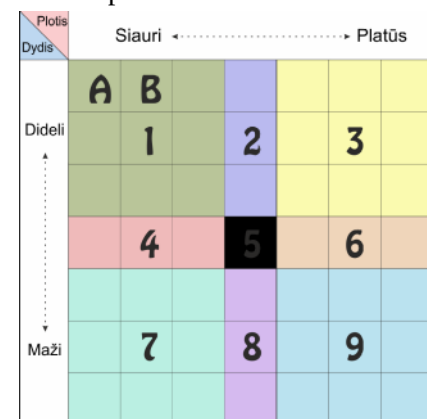
Bebrui gali pasisekti, ir tinkamus batus jis rastų jau pirmojo bandymo metu. Kitu atveju, tinkančius batus jis gali atrasti antruoju bandymu.

- Bebras gali pradėti matuoti nuo batų, esančių centre, kaip parodyta paveikslėlyje.
- Batas bus tinkamo dydžio, mažesnis, didesnis ir tinkamo pločio, per siauras arba per platus. Priklausomai nuo rezultato, bebras žinos, kad tinkamas batas bus vienoje iš devynių skirtingomis spalvomis pažymėtų zonų.
- Jei batas tiko, jis rado reikiamą apavą.
- Jei batas per mažas ir per platus, jis pasimatuos 1-oje zonoje esančius batus.
- Jei batas yra per mažas, tačiau tinkamo pločio, jis pasimatuos 2-oje zonoje esančius batus.
- Jei batas per didelis ir per siauras, jis matuos 9-oje zonoje esančius batus ir taip toliau.



Sakykime, kad bebras pasimatavo per mažus ir per plačius batus. Dabar jis turėtų matuoti didesnius ir siauresnius, kurie padėti 1-oje zonoje. Jis pasimatuoja batą, kuris padėtas 1-osios zonos centre (lentynėlė pažymėta skaičiumi 1).

- Jei batas jam tinka, jis rado tinkamą apavą.
 - Jei batas per mažas ir per platus, jam tiks batai A lentynėlėje.
 - Jei batas jam per mažas, bet gero pločio, jam tiks batai A lentynėlėje.
- Kaip matome, bebras turi pasimatuoti daugiausiai du skirtingus batus, kad rastų tinkamus.
- Jei jis pradėtų matuoti iš bet kurios kitos pozicijos, tektų pasimatuoti daugiau batų.



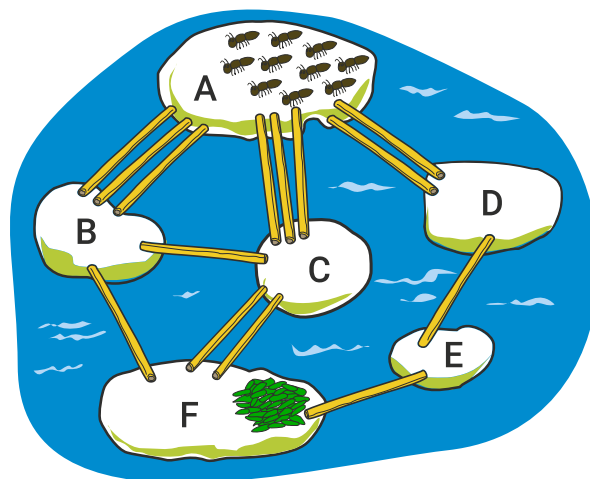
Tai informatika!

Užduotyje batai vitrinoje išdėlioti didėjančio dydžio ir pločio tvarka pagal dvi ašis. Toks išdėliojimas vadinamas rikiavimu. Dvejetainės paieškos algoritmai, naudojami surikiuotiems duomenims, reikiamą reikšmę randa labai greitai. Dvejetainė paieška paieškos erdvę kaskart mažina per pusę tam, kad teisingą reikšmę rastų naudojant kuo mažiau bandymų. Dvejetainę paiešką galima naudoti ir žaidžiant, pavyzdžiui, bandant atspėti skaičių nuo 1 iki 100 ir panaudojant kuo mažiau bandymų. Šis uždavinys iliustruoja dvejetainės paieškos algoritmą, ieškant reikiamos reikšmės dviejų dimensijų erdvėje.



30. Skruzdės pelkėje

Dešimt skruzdių yra ant A akmens ir stengiasi pasiekti maistą, esantį ant F akmens. Tuo pačiu metu per šiaudėlį gali eiti tik viena skruzde, ir užtrunka 1 minutę, kol skruzde pereina nuo vieno akmens ant kito.



Kiek daugiausiai skruzdių po 3 minučių gali pasiekti maistą, esantį ant F akmens?

Paiškinimas

Atsakymas: 7.

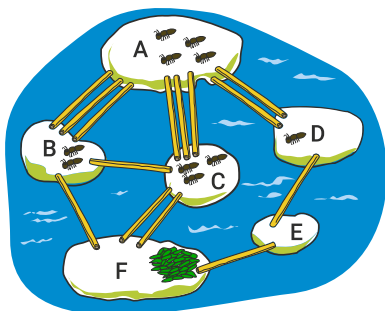
Galime pastebėti, kad:

- Nėra jokios prasmės siųsti daugiau nei vienos skruzdės per D-E akmenis
- Nėra jokios prasmės siųsti daugiau nei dvi skruzdes per A-B akmenis
- Šiaudelis tarp B-C akmenų visiškai nepadaeda, tad jį galime ignoruoti
- Srautą apriboja šiaudeliai, sujungiantys B-F ir C-F akmenis

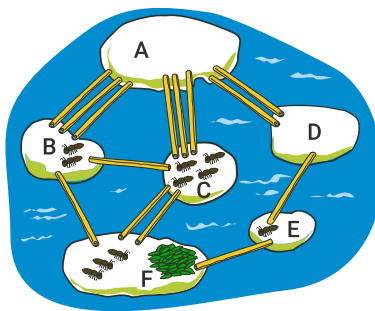
Pasinaudoję šiais teiginiais galime nesunkiai rasti optimalų sprendimą.

Žemiau pateiktose nuotraukose yra parodyta galima situacija po kiekvienos minutės.

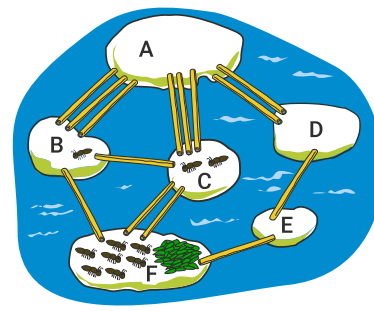
Po vienos minutės:



Po dviejų minučių:



Po trijų minučių:



Tai informatika!

Tikslas yra optimizuoti skruzdių srautą, kad kuo daugiau jų pasiektų maistą per 3 minutes. Tai vadinama optimizavimo uždaviniu.

Tos skruzdės, kurios nesupranta akmenų, per kuriuos keliauja, išdėstymo, nebus pajėgios išsiaiškinti geriausio kelio. Tačiau stebėtojas, kuris mato visą akmenų išdėstymą, gali surasti optimalią strategiją. Šiame uždavinyje mes tariame, kad skruzdės nusimano apie akmenų išdėstymą ir tai lemia jų pasirinkimą judėti konkrečiu keliu.

Grafai yra abstrakčios duomenų struktūros, naudojamos modeliuoti tinklus, ir yra daugybė algoritmų, padedančių optimizuoti srautą esant tam tikroms aplinkybėms.

https://en.wikipedia.org/wiki/Flow_network

https://en.wikipedia.org/wiki/Maximum_flow_problem

https://en.wikipedia.org/wiki/Optimization_problem

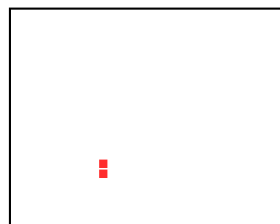
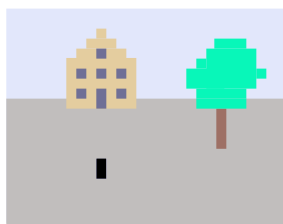
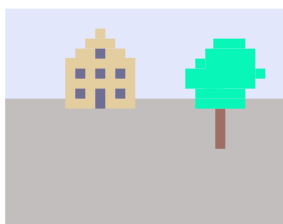
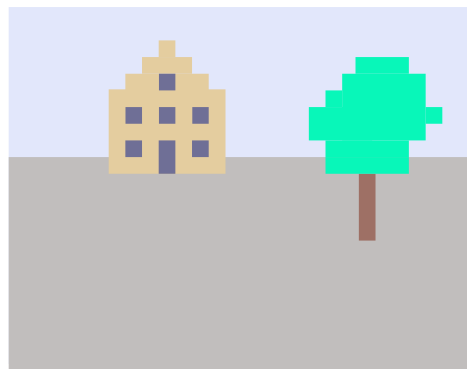
	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
				12	9	6

31. Stebėjimas

Skaitmeninė kamera fotografuoja rotušės vaizdą kas 10 sekundžių (žr. paveikslą). Kompiuterio programa lygina kiekvieną nuotrauką su ankstesne ir formuoja pokyčių paveikslą. Mažas raudonas stačiakampis pokyčių paveiksle žymi kiekvieną nuotraukos vietą, besiskiriančią nuo prieš tai darytos nuotraukos.

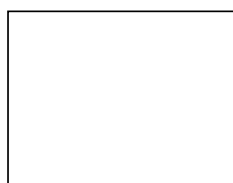
Jei pokyčių paveikslas baltas, tai abi nuotraukos yra vienodos.

Pavyzdžiui, dviem toliau pateiktoms nuotraukoms pokyčių paveikslas (dešinėje) rodo skirtumus tarp pirmos ir antros nuotraukos.

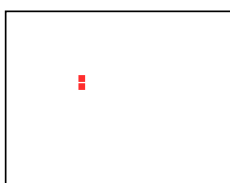


Pateikiama 50 sekundžių intervalo pokyčių paveikslų seka. Per šį laiką įvyko 5 įvykiai.

Perkelkite kiekvieną įvykį į jam tinkamą vietą.



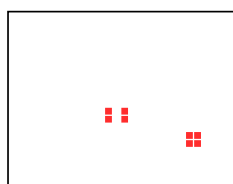
00



10



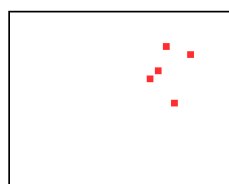
20



30



40



50

A: Tomas sutinka Laurą.
B: Kažkas atidaro rotušės duris.
C: Tomas ir Laura eina kartu susikibę rankomis.
D: Vėjas stiprėja.
E: Kažkas uždaro rotušės duris.

Paaiškinimas

Atsakymas: Yra du teisingi atsakymai: **BACED** ir **EACBD**.

Tarp 0 ir 10 sekundės: Kažkas atidaro rotušės duris.

10-s sekundės pokyčių paveikslas vaizduoja raudonus stačiakampius toje vietoje, kur yra durys, – pokytis galėjo įvykti atidarius (uždarius) duris.

Tarp 10 ir 20 sekundės: Tomas sutinka Laurą.

Pokyčių paveikslo viduje esantys raudoni stačiakampiai, atsiradę 20-ąją sekundę, rodo vietą, kur susitiko Tomas su Laura. Vienas iš jų atėjo iš kairės, kitas iš dešinės. Jų dar nebuvo nuotraukoje, darytoje prieš 10 sekundžių.

Tarp 20 ir 30 sekundės: Tomas ir Laura eina kartu susikibę rankomis.

30-ąją sekundę darytos nuotraukos pokyčių paveikslas vaizduoja raudonus stačiakampius, vaizduojančius

ankstesnę ir dabartinę Tomo ir Lauros buvimo vietą. Eidami jie buvo arti vienas kito, todėl fotoaparatas fiksuoja juos kaip vieną dėmę.

Tarp 30 ir 40 sekundės: Kažkas uždaro (arba atidaro) rotušės duris, kol Tomas su Laura eina iš rotušės aikštės. Pokyčių paveikslas po 40 sekundės rodo raudonus stačiakampius vietoje, kur yra durys (įvyko pokytis dėl durų uždarymo ar atidarymo) ir Tomo ir Lauros ankstesnėje buvimo vietoje, kurių čia nebeliko (jie spėjo išeiti per 10 sekundžių).

Tarp 40 ir 50 sekundės: Vėjas stiprėja.

50 sekundės pokyčių paveikslas rodo raudonus stačiakampius medžio lapų vietoje. Tai sako, kad įvyko pokyčiai (pvz., medžio lapų judėjimas), kurie galėjo atsirasti dėl stipraus vėjo.

Pastaba. B ir E įvykiai gali būti sukeisti vietomis – vien iš pokyčių paveikslo neįmanoma atspėti, kuris įvykis vyko pirmas: durų atidarymas ar uždarymas.

Tai informatika!

Vaizdų apdorojimo automatizavimas ir analizė yra svarbi saugumo sistemų viešosiose vietose (pvz., oro uostose, traukinių stotyse) funkcija. Ji gali būti naudojama įsibrovėliams identifikuoti sunkiai pasiekiamose vietose arba tirti įtariamųjų asmenų elgesį. Tačiau tokia analizė nepalieka privatumo viešosiose vietose, kurios nuolat stebimos.

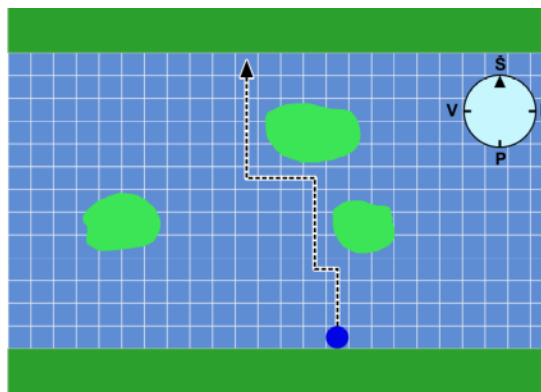
Pavyzdys iliustruoja, kad programa iš interneto kameros gali gauti daug informacijos. Paveikslų analizės užduotis gali būti paprasta (pvz., QR ar brūkšninių kodų nuskaitymas prekybos centruose) arba sudėtinga (pvz., asmens amžiaus grupės ir lyties automatinis nustatymas iš veido nuotraukos).

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
				9	6	

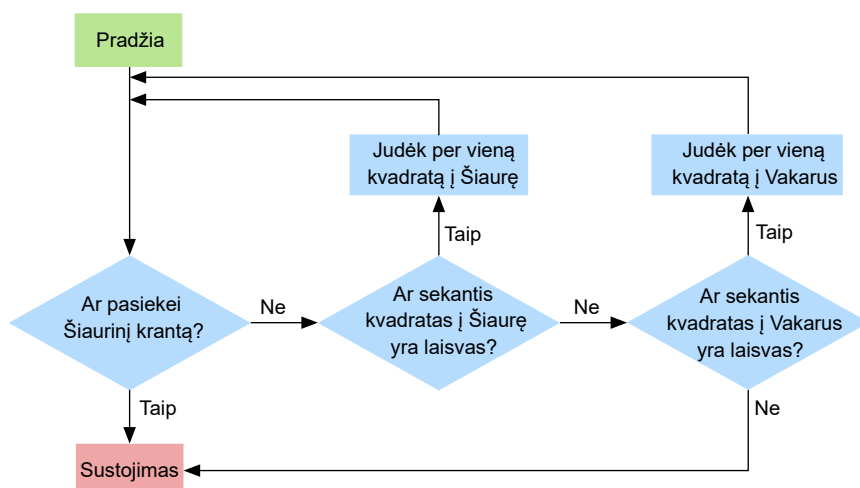
32. Spąstų sala

Autonominis laivas plaukia iš pietinio upės kranto į Šiaurinį krantą. Jis turi aplaukti aplink salas upėje.

Laivas keliauja skaitmeninio žemėlapio kvadratais (iš vieno kvadrato į kitą). Laivas gali plaukti į kvadratą tik tada, kai jis yra laisvas ir nėra iš dalies užblokuotas salos.

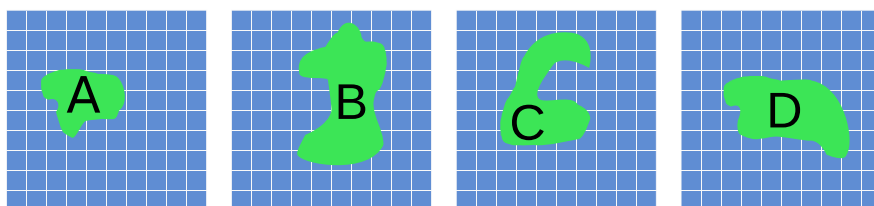


Autonominis laivas plaukioja pagal šias instrukcijas:



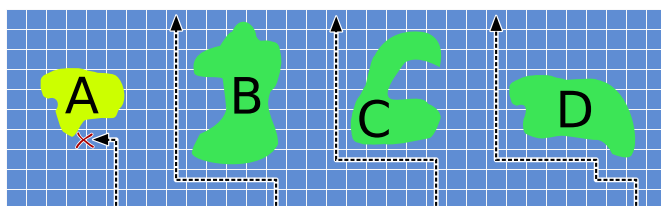
Piratai įdiegė dirbtinę salą, kuri gali būti spąstai autonominei valčiai. Tuomet, jei autonominis laivas priartės prie šios salos, gali atsitikti taip, kad laivas sustos prieš pasiekdamas Šiaurinį krantą.

Kuri iš šių salų gali būti spąstai autonominei valčiai?



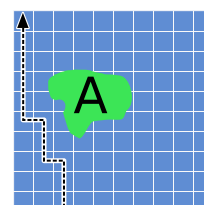
Paiškinimas

Atsakymas: Kairėje esanti sala yra dirbtinė sala (A).



Jei autonominė valtis artėja prie salos, kaip parodyta paveikslėlyje, ji sustoja nepasiekusi Šiaurinio kranto. Priežastis yra ta, kad ji negali judėti nei į Šiaurę, nei į Vakarus.

Nors į salą galima patekti iš kito pradžios taško, kuriame valtis nebus įstrigusi, tačiau kitose salose (B, C, D) neįmanoma įstrigti. Pietinėse šių salų pakrantėse nėra tokios kvadrato, kur šiauriniai ir vakariniai kvadratai nėra laisvi. Taigi, autonominis laivas, artėjantis prie vienos iš šių salų iš Pietų, visada gali pasitraukti į Vakarus.



Tai informatika!

Blokinė schema yra diagramos rūšis, vaizduojanti algoritmą, darbo eigą ar procesą. Blokinėje schemoje žingsniai vaizduojami įvairių rūšių dėžutėmis, o jų tvarka pavaizduojama sujungiant dėžutes rodyklėmis. Tiksliai apibrėžtoje schemoje būtų aiškiai atskirtos šakos nuo bet kurio sprendimo taško ir apibrėžtas bet kokio scenarijaus procesas.

Autonominę valtį valdanti schema puikiai tinka esamoms apvalios formos saloms. Tačiau blokinėje schemoje neatsižvelgta į tam tikras sąlygas, kurios leistų nepriartėti prie spąstais esančios salų ir leistų pasiekti Šiaurinį upės krantą. Blokinėje schemoje yra trūkumų, saugumo pažeidimų, kuriais gali pasinaudoti piktybiniai žmonės.

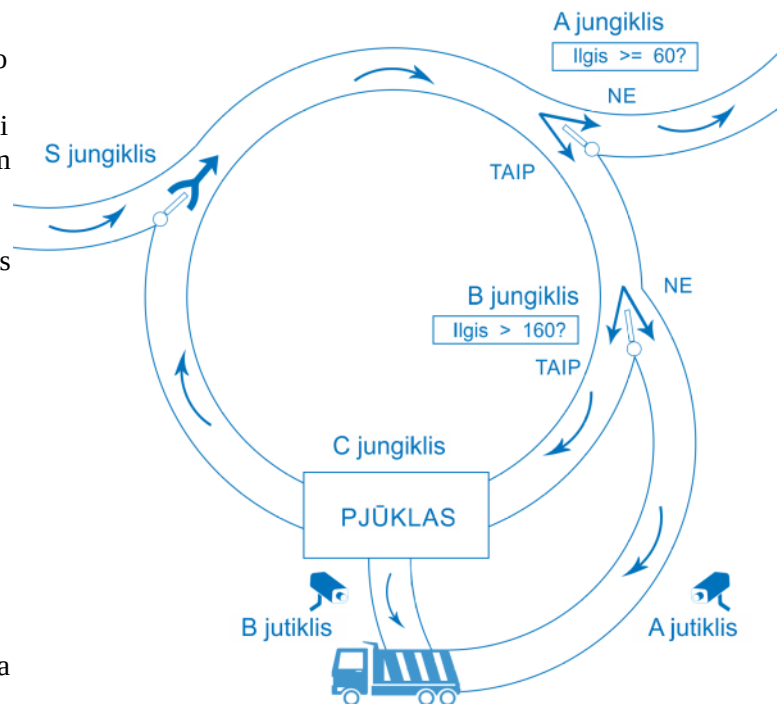
Kompiuterių programuotojai bando sukurti programinę įrangą, kuri puikiai veikia bet kokiomis sąlygomis. Pavyzdžiui, interaktyvi svetainė su įvesties laukais atidžiai tikrina įvesties duomenis ir atsisako bet kokių įvestų duomenų, kurie gali būti pavojingi programinei įrangai ar kompiuteriui, pavyzdžiui, kenksmingas scenarijaus kodas.



33. Lentpjūvė

Norint pastatyti rąstinį namą, reikia tinkamo ilgio rąstų. Į lentpjūvę atvežami įvairaus ilgio rąstai, iš kurių atrenkami ir supjaunami statybai tinkami rąstai nuo 60 cm iki 160 cm ilgio. Visi jie kraunami į sunkvežimį. Toliau pateikta lentpjūvės darbo schema. Rąstų atvežimo vieta pavaizduota lentpjūvės schemos kairėje dalyje. Lentpjūvė turi tokias dalis:

- S jungiklis jungia pradinį ir pakartotinio pjovimo srautus į vieną srautą.
- A jungiklis praleidžia 60 cm ir ilgesnius rąstus bei siunčia kitur trumpesnius rąstus.
- B jungiklis praleidžia ilgesnius kaip 160 cm rąstus, o likusius perduoda krauti į sunkvežimį.
- C jungiklis suaktyvina pjūklą vienam rąstui padalinti į du. Vienos padalinto rąsto dalies ilgis bus lygiai 160 cm, ir ji kraunama į sunkvežimį. Likusi rąsto dalis siunčiama tolesniam apdorojimui.
- A ir B jutikliai naudojami į sunkvežimį kraunamiems rąstams suskaičiuoti.



Į lentpjūvę siunčiami trys skirtingo ilgio rąstai (60 cm, 140 cm ir 360 cm).

Kiek rąstų suskaičiuos A ir B jutikliai po visiško šių rąstų apdorojimo?

- A jutiklis: 1 rąstas, B jutiklis: 3 rąstai
- A jutiklis: 3 rąstai, B jutiklis: 1 rąstas
- A jutiklis: 2 rąstai, B jutiklis: 2 rąstai
- A jutiklis: 0 rąstai, B jutiklis: 4 rąstai

Paaškinimas

Atsakymas: C.

Kadangi rąstai nesujungti fiziškai, jų apdorojimo tvarka nesvarbi.

- Jei rąstas yra 60 cm ilgio, jis praeina per A jungiklį. B jungiklis siunčia jį į sunkvežimį. A jutiklis suskaičiuoja vieną rąstą.
- Jei rąsto ilgis yra 140 cm, jis praeina per A jungiklį. B jungiklis siunčia jį į sunkvežimį. A jutiklis suskaičiuoja vieną rąstą.
- Jei rąstas yra 360 cm ilgio, jis praeina per A ir B jungiklius. C jungiklis įjungia pjūklą ir supjausto jį du rąstus, kurių ilgiai yra 160 cm ir 200 cm. 160 cm rąstas siunčiamas į sunkvežimį, o B jutiklis suskaičiuoja vieną rąstą. 200 cm rąstas vėl eina per S, A ir B jungiklius. C jungiklis įjungia pjūklą ir supjausto rąstą į dvi dalis, kurių ilgiai yra 160 cm ir 40 cm. 160 cm rąstas siunčiamas į sunkvežimį, o B jutiklis suskaičiuoja dar vieną rąstą. 40 cm rąstas vėl eina per S jungiklį, tačiau A jungiklis jo nepraleidžia, kadangi yra per trumpas. Iš viso A ir B jutikliai suskaičiuos po 2 rąstus.

Tai informatika!

Rąstų srautas gali būti nagrinėjamas kaip sveikųjų skaičių seka. Ši užduotis yra tiesioginis reaktyviojo programavimo taikymas, kai sveikųjų skaičių srautas (rąstų ilgiai) apdorojamas keliomis reaktyviosiomis operacijomis (jungti, filtruoti, transformuoti), o reikalingi matavimai atliekami skenuojant.

S jungiklis atlieka jungimo operaciją, sujungiančią du srautus. A ir B jungikliai atlieka filtrų operacijas – kiekviena srautą padalija į dvi dalis remdamiesi ribine reikšme. C jungiklis atlieka transformacijos operaciją, paversdamas sveikuosius skaičius sveikųjų skaičių poromis. Jutikliai atlieka skenavimo operacijas, jei įsivaizduotume, kad jie įsimena pro juos praeinančių rąstų skaičių.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
					12	12

34. Ekologiški skrydžių maršrutai

Tarptautinės oro linijų kompanijos „Bebras“ skrydžių maršrutai, jungiantys didžiausius pasaulio miestus, pavaizduoti paveiksle.



Skrydžių metu išmetamos į aplinką anglies dvideginio (CO₂) dujos didina klimato atšilimą. Siekdama sumažinti išmetamų dujų kiekį skrydžių kompanija ketina atsisakyti tam tikrų tiesioginių maršrutų nesukeldama didelių nepatogumų keleiviams, skrendantiems į bet kurį miestą. Pavyzdžiui, panaikinus tiesioginį skrydį San Fransiskas–Vašingtonas, keleiviai skristų iš San Fransisko į Niujorką, o iš Niujorko į Vašingtoną.

Kiek daugiausiai tiesioginių skrydžių maršrutų oro kompanija „Bebras“ gali atsisakyti?

A) 6 B) 7 C) 8 D) 9

Paaikškinimas

Atsakymas: C.

Panaikinus 8 tiesioginių skrydžių maršrutus, paveiksle pavaizduotus punktyrinėmis linijomis, dar įmanoma nuskristi iš bet kurio miesto į bet kurį kitą.

Iš viso buvo 15 maršrutų, o miestų – 8. Todėl likusių 7 maršrutų visiškai pakanka. Jei būtų tik 2 miestai, tai pakaktų vieno maršruto, jei 3 – dviejų. Tęsiant tokią seką iki 8, lieka 7 maršrutai.

Kad įsitikintume tokios sekos teisingumu, įrodykime, kad panaikinus 9 ar daugiau maršrutų, nebebus galimybės nuskristi į bet kurį miestą. Kadangi tokiu atveju lieka 6 maršrutai (ar mažiau), o miestų yra 8, lieka vienas miestas (ar daugiau) į kurį (kuriuos) skrydžių nėra.

Tai informatika!

Skrydžių maršrutai gali būti vaizduojami grafu. Miestai atitinka grafo viršūnes, o skrydžių maršrutai – briaunas. Uždavinys – atmesti visas briaunas (skrydžių maršrutus) taip, kad visos viršūnės (miestai) būtų sujungti. Tai minimalios aprėpties medžio (Minimum Spanning Tree) uždavinys programavime.

Programuotojai minimalios aprėpties medžio algoritmą naudoja sprenddami telekomunikacijų tinklų, transporto maršrutų, vandens tiekimo uždavinius.

Daugiau informacijos

https://en.wikipedia.org/wiki/Minimum_spanning_tree

<https://klevas.mif.vu.lt/~ragaisis/ADS/Grafai.htm>

[http://klevas.mif.vu.lt/~ragaisis/ADS/Grafai\(2\).htm](http://klevas.mif.vu.lt/~ragaisis/ADS/Grafai(2).htm)



Nykštukai

Mažyliai

Bičiuliai

Draugai

Jauniai

Kolegos

12

9

35. Traukinio vagonų rikiavimas

Traukinio vagonai atkabinami pagrindinėje geležinkelio atšakoje. Formuojant sąstatą vagonai turi būti surikiuoti taip, kad prie lokomotyvo kabinamo vagono numeris būtų pirmas: .

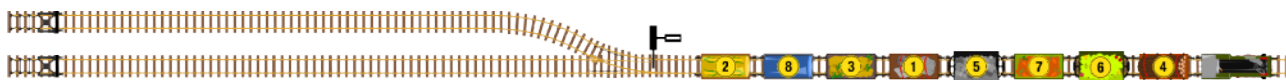
Vagonų rikiavimo zonoje vagonai stumdomi kairėn ar dešinėn ir gali būti perstumiami į gretutines bėgių atšakas naudojant rampą. Per rampą perstumti vagonai prikabinami prie lokomotyvo pasirinktu eiliškumu. Ši procedūra laikoma viena operacija.

Pavyzdžiui, jei yra 4 vagonai, tai norint juos išrikiuoti, tereikia atlikti dvi operacijas (①+ ② žingsniai):



Šių vagonų neįmanoma surikiuoti naudojant vieną operaciją.

Traukinio vagonai stovi tokia tvarka: 2 – 8 – 3 – 1 – 5 – 7 – 6 – 4. Kiek mažiausiai reikės atlikti operacijų, kad jie būtų surikiuoti 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8?



A) 3 B) 4 C) 5 D) 6

Paaškinimas

Atsakymas: A (tris kartus).

Vagonus galima surikiuoti įvairiais būdais. Vienas geriausių: vagonus, pažymėtus numeriais 1, 3, 5, 7, perstumti į viršutinę bėgių atšaką, o vagonus, pažymėtus numeriais 2, 4, 6, 8, perstumti į žemesnę atšaką. Vagonai dar nebus išrikiuoti reikiama tvarka, tačiau, jei pirmiausia bus prikabinas apatinės atšakos sąstatas, o tik po to viršutinės atšakos sąstatas, tai gausime štai tokį sąstatą:



Toliau vagonus 1, 5, 2 ir 6 reikėtų perstumti į viršutinę atšaką, o vagonus 3, 7, 8 ir 4 – į apatinę. Tokiu būdu vagonai bus surikiuoti štai taip:



Galiausiai numeriais nuo 1 iki 4 pažymėti vagonai išstumiami į viršutinę atšaką, o nuo 5 iki 8 – į apatinę. Pagaliau sąstatas surikiuotas:



Vagonai negali būti surikiuoti greičiau, nes 4-tas vagonas turi būti pirmesnis nei 8-tas (vienas pervažiavimas per rampą), 6-tas vagonas turi būti prieš 8-tą, bet už 4-to (antras kartas per rampą, nes tai negali būti atlikta vienu kartu), o 7-as vagonas turi būti tarp 6-to ir 8-to (trys kartai per rampą, nes pirmuoju kartu jis gali būti perstumtas prieš 8-tą, tačiau negali būti pastatytas prieš 7-tą; jei 6-tas vagonas antruoju kartu perstumiamas prieš 7-tą, tai 4-tas vagonas negali būti perstumtas prieš 7).

Tai informatika!

Geležinkelininkai visame pasaulyje su šiuo uždaviniu susiduria kasdien. Stumdyti ir rikiuoti vagonus yra varginantis ir daug ištvermės reikalaujantis darbas: žmogus kaskart rankiniu būdu turi sukabinti ir atkabinti vagonus. Šis procesas užtrunka ir blokuoja pagrindinę geležinkelio liniją, ypač jei kai kurie vagonai turi būti palikti pagrindiniame kelyje, kol lokomotyvas nustums kitus vagonus į gretutinę atšaką. Dėl šios priežasties didesnės geležinkelio stotys turi milžiniškas vagonų rikiavimo zonas su daug atšakų. Šioje užduotyje minimos tik dvi atšakos, todėl ilgesnių sąstatų rikiavimo užduotis būtų labai sudėtinga. Tačiau mažesnėse stotyse tai visiškai įprasta situacija.

Informatika pagelbėja efektyvinant vagonų rikiavimą. Šiuo atveju, uždavinys sprendžiamas dalimis per keletą kartų. Šis būdas vadinamas „skaldyk ir valdyk“ (https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm) metodu (algoritmu). Spręsdami šią 8 vagonų rikiavimo užduotį, pirmiausia padalijame į dvi dalis ir išrikiuojame 4 vagonus.

Gretutinės bėgių atšakos veikia kaip abstraktusis duomenų tipas - dėklas (angl. stack), kuris naudojamas tik informatikoje. Vienintelės galimos operacijos – įdėti, pristumti (angl. push) ir ištraukti, atkabinti (angl. pull).



Nykštukai

Mažyliai

Bičiuliai

Draugai

Jauniai

Kolegos

12

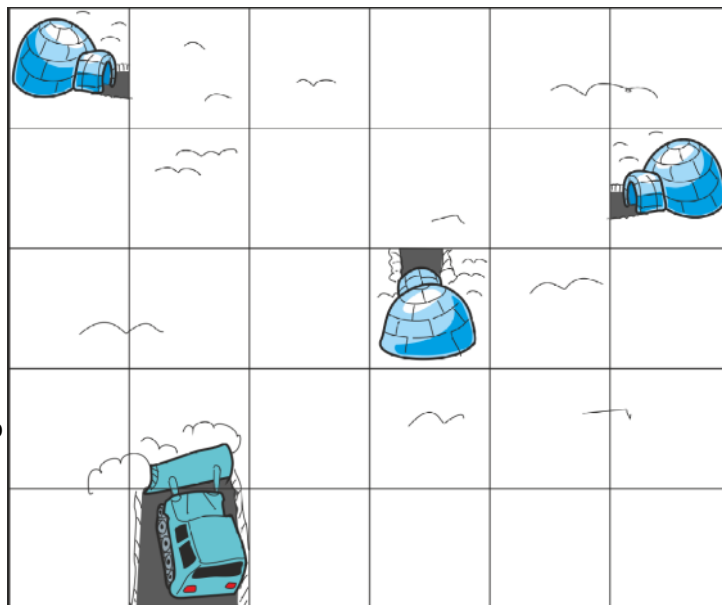
9

36. Gelbėjimo misija

Po sniego audros trys šeimos nori, kad jų ledro trobelės būtų sujungtos su pagrindine kelių sistema. Sniego valymo robotui duodama užduotis nuvalyti kelią iki visų trobelių. Atlikęs užduotį, robotas turi grįžti į pradinę padėtį.

Jeigu langelyje nėra sniego, robotas pereina iš vieno langelio į kitą per 1 valandą.
Jeigu langelyje reikia nuvalyti sniegą, pereiti iš vieno langelio į kitą užtrunka 2 valandas.
Roboto apsisukimui išvalytame langelyje laiko nereikia.
Robotas neina į trobelę, tik išvalo langelį priešais įėjimą į trobelę.

Koks trumpiausias roboto užduoties atlikimo laikas?



Paaškinimas

Atsakymas: 21 valanda.

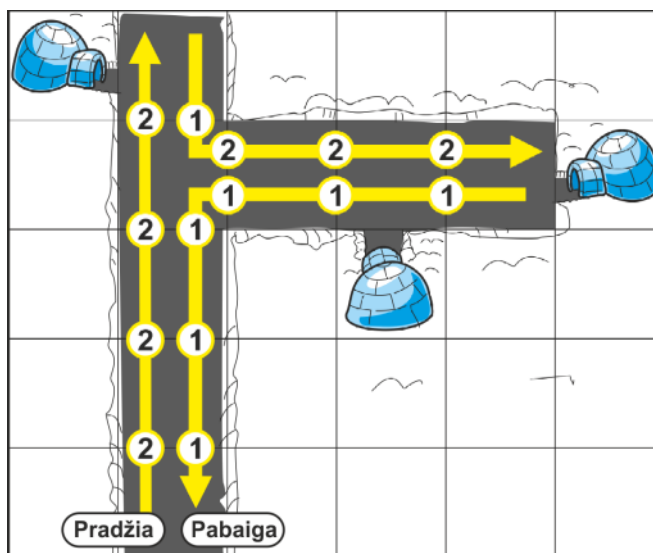
Kadangi pakartotinis išvalyto langelio praėjimas užtrunka mažiau laiko nei sniego valymas, būtų optimalu judėti išvalytais keliais. Galime valyti kelią, pateiktą dešinėje esančiame paveikslėlyje.

Kokia bus viso kelio valymo ir grįžimo trukmė?

Sudėjus paveiksle pavaizduoto kelio nuvalymui skirtą laiką žyminčius skaičius, gaunamas teisingas atsakymas.

Ar galime gelbėjimo misiją atlikti greičiau nei per 21 valandą?

Kadangi sniego valymas užima daugiau laiko nei paprasčiausias langelio praėjimas, norėtume kuo labiau sumažinti sniego valymą. Aukščiau esančiame paveikslėlyje visi išvalyti langeliai naudojami pakartotinai, kad būtų sumažintas laikas. Kitas būdas sumažinti laiką būtų trumpinti grįžimo kelią per nevalytus langelius, tačiau šioje schemoje to pasiekti negalima. Kadangi šiame uždavinyje pateikto kelio praeinamų langelių skaičiaus sumažinti neįmanoma, tai gautas rezultatas yra geriausias.



Tai informatika!

Šis optimizavimo uždavinys yra ypatingas keliaujančio pirklio problemos atvejis – ieškomas trumpiausias kelias su reikalavimu aplankyti visus mazgus. Hamiltono kelias – tai kelias, kuriame visi mazgai aplankomi tik po vieną kartą. Gelbėjimo misijos uždavinys yra šiek tiek ypatingas, nes pirmasis įėjimas į langelį su sniego valymu „kainuoja“ daugiau laiko nei kiti to langelio perėjimai.

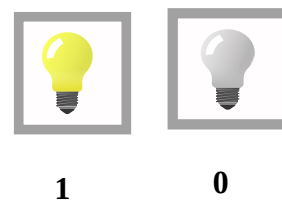
Šiuo metu yra nežinomas veiksmingas tokių problemų sprendimo algoritmas. Optimalių sprendimų suradimas dideliame plotui su tūkstančiais langelių užtruktų labai ilgai, net naudojant galingus kompiuterius.

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
					12	9

37. Dvejetainės lemputės

Pranešimui nuotoliniu būdu siųsti Bebrai sugalvojo naudoti dvejetaines lemputes. Jie naudoja tokią lentelę ir taisyklę pranešimui siųsti:

A	01000001	J	01001010	S	01010011
B	01000010	K	01001011	T	01010100
C	01000011	L	01001100	U	01010101
D	01000100	M	01001101	V	01010110
E	01000101	N	01001110	W	01010111
F	01000110	O	01001111	X	01011000
G	01000111	P	01010000	Y	01011001
H	01001000	Q	01010001	Z	01011010
I	01001001	R	01010010		



Naudodami lentelę ir nurodytą taisyklę Bebrai išsiuntė tokį pranešimą:



A) NAMAS B) MAŽAS C) SODAS D) VĖJAS

Paiškinimas

Atsakymas: A) NAMAS.

Naudodami pateiktą lentelę ir taisyklę, konstatuojame:



Tai informatika!

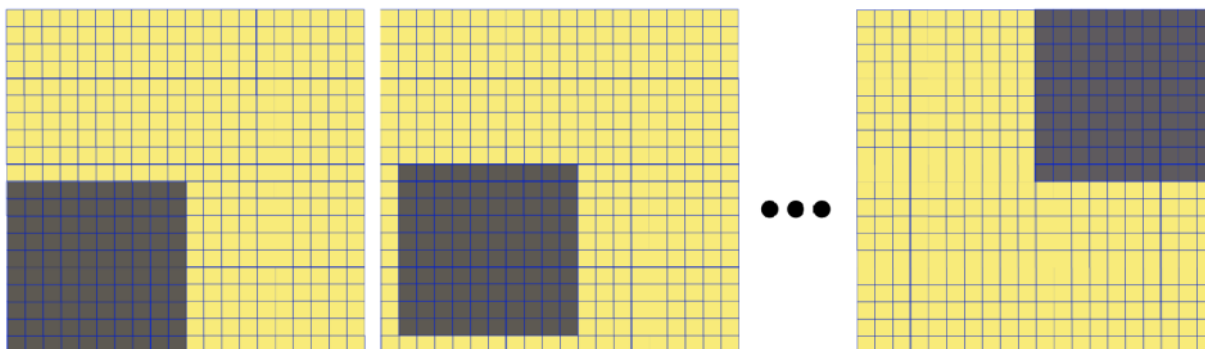
Dvejetainė skaičiavimo sistema, išrasta Gotfrydo Leibnico, yra sudaryta tik iš dviejų skaitmenų: 0 ir 1. Ši sistema yra pagrindas visų dvejetainių kodų, naudojamų užrašyti kompiuterių arba procesorių atliekamas komandas arba dvejetainį tekstą, kurį kasdien matome ir skaitome.

Dvejetainę skaičiavimo sistemą nesunkiai galime rasti mūsų kasdieniame gyvenime. Lemputės ir perjungikliai – puikūs šios sistemos naudojimo pavyzdžiai, ką matome iš šios užduoties. Naudodami šią sistemą galime koduoti pranešimus, kuriuos lengva persiųsti kitiems. Morzės abėcėlė yra kitas pavyzdys.

Daugiau informacijos galima rasti straipsnyje: https://lt.wikipedia.org/wiki/Dvejetain%C4%97_skai%C4%8Diavimo_sistema (lietuvių k.), https://en.wikipedia.org/wiki/Binary_code (anglų k.).

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
					12	9

38. Vaizdų glaudinimas



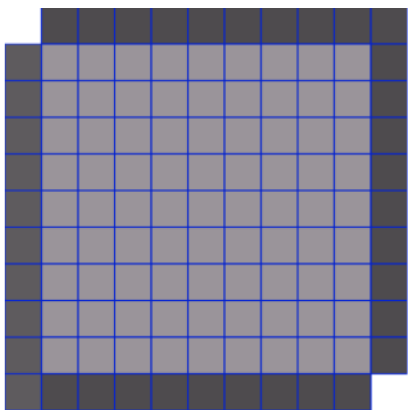
Skaitmeninis taškinis paveikslas – tai spalvotų kvadratėlių, vadinamų taškais arba pikseliais, stačiakampis tinklėlis. Vaizdo įrašas – tai paveikslų, vadinamų kadrais, seka, kurioje kiekvienas kadras šiek tiek skiriasi nuo ankstesnio. Paprasčiausias būdas saugoti vaizdo įrašą – tai įrašyti tik tuos pikselius, kuriais tolesnis kadras skiriasi nuo ankstesnio. Pavyzdžio paveiksle pavaizduotas 10×10 dydžio tamsios stačiakampės srities judesys iš kairiojo apatinio kampo į dešinį viršutinį 20×20 dydžio šviesios srities kampą. Kiekviename kadre stačiakampis juda įstrižai (po vieną pikselį horizontaliai ir vertikalčiai). Toks judesys užima 11 kadrų. Jei laikytume šį vaizdo įrašą įprastu formatu, mums reiktų $(20 \times 20) \times 11 = 4400$ pikselių.

Kiek vaizdo įrašui reiktų pikselių, jei jį laikytume efektyvesniu formatu?

Paaškinimas

Atsakymas: 780

- ⌚ Pirmam kadrai turime įsiminti $20 \times 20 = 400$ pikselių.
- ⌚ Kiekvieno kadro pokytis, palyginus su tolesniu kadru, užima 38 pikselius, kaip parodyta paveiksle.



- ⌚ Po pirmo kadro yra 10 kadrų, taigi iš viso reikia įsiminti $400 + (38 \times 10) = 780$ pikselių.

Tai informatika!

Duomenų glaudinimas yra svarbi informatikos sritis, ypač naudinga vaizdo ir garso duomenims. Kai kurie gerai žinomi paveikslų formatai, pvz., JPEG, glaudinant praranda dalį duomenų: mažėja kai kurių spalvų, prarandamos tam tikrų sričių vizualios ribos. Šioje užduotyje aprašytas nenuostolingasis glaudinimas, t. y. glaudinimas, kurio metu neprarandama informacijos. Išdėsčius kadrus vieną po kito galima laikyti, kad kiekvienas kadras yra trimatis masyvas. Pokytis ekrane atsitinka tik tuose pikseliuose, kurie skiriasi nuo ankstesnių kadrų.

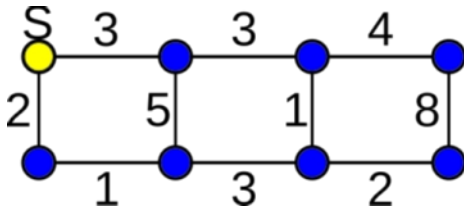
https://en.wikipedia.org/wiki/Data_compression#Video

Reikia turėti omenyje, kad pateiktame algoritme reikia įsiminti ne tik pikselio spalvą, bet ir jo koordinates.



39. Bebrų keliai

Jūs dirbate siuntinių išvežiojimo įmonėje „Byber“. Išvažiuojate iš taško S ir turite pristatyti siuntinius į septynias vietas, kurios pažymėtos mėlynais skrituliukais. Vietovės sujungtos keliais, kurie pavaizduoti atkarpomis. Prie kiekvienos atkarpos esantis skaičius rodo mokesčio kainą, važiuojant atitinkamu keliu. Jums mokama nuo bendros kelių mokesčiams išleistos sumos, todėl esate suinteresuotas sumokėti kuo daugiau kelių mokesčių. Kiekvieną vietovę galite aplankyti tik vieną kartą, tačiau užbaigti darbą galima bet kurioje vietovėje.



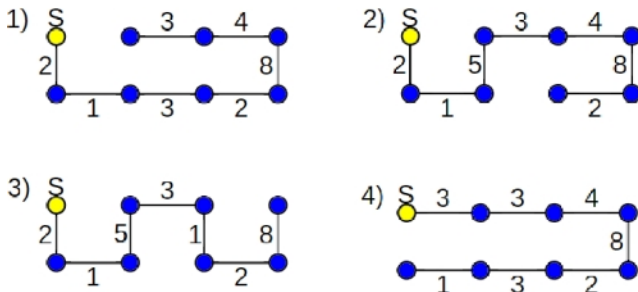
Kokią didžiausią sumą jūs galite išleisti kelių mokesčiams išvežiodamas šiuos 7 siuntinius?

- A) 23 B) 24 C) 25 D) 26

Paiškinimas

Atsakymas: C) 25

Yra tik 4 galimi maršrutai, kai startuojama taške S ir tik vieną kartą aplankomos visos vietovės:



Kiekvienu atveju kelių mokesčiai bus tokie:

- 1) $2+1+3+2+8+4+3 = 23$
2) $2+1+5+3+4+8+2 = 25$
3) $2+1+5+3+1+2+8 = 22$
4) $3+3+4+8+2+3+1 = 24$

(sumuojama pradedant S taške ir keliaujant išilgai maršruto). Didžiausia suma (25) yra sumokama išvežiodami siuntinius antruoju maršrutu.

Galima pastebėti, kad aplankant kiekvieną vietovę tik vieną kartą visada yra naudojami 7 keliai ir lieka 3 nepanaudoti keliai. Nepanaudotas kelias gali būti kiekvienai vietovei tik vienas, jų jokių būdu negali būti dviejų, nes tokiu atveju tektų sugrįžti į tą pačią vietovę antrą kartą. Kiekviena vietovė su kitomis yra sujungta 2 arba 3 keliais.

Tai informatika!

Programavime tokio tipo schemas yra vadinamos grafais. Vietovės atitinka grafo viršūnes, o keliai – briaunas.

Geriausiam ar blogiausiam grafo maršrutui rasti yra naudojami skirtingi algoritmai. Šiame uždavinyje reikalaujama rasti maršrutą, kuriuo visos viršūnės aplankomos tik vieną kartą. Toks maršrutas vadinamas Hamiltono maršrutu.

https://lt.wikipedia.org/wiki/Hamiltono_mar%C5%A1rutas

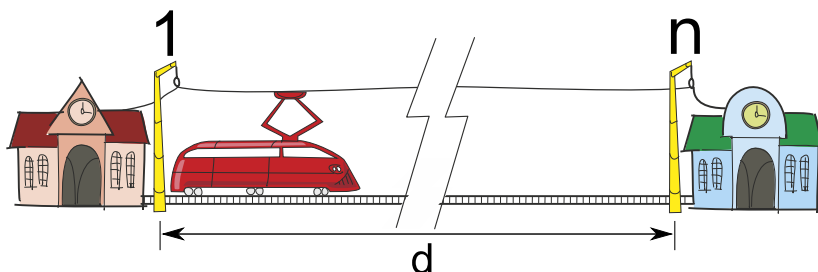
Šis uždavinys reikalauja ne tik rasti Hamiltono maršrutą, bet ir patikrinti, ar jis yra pats pelningiausias. Tai dar vienas garsus informatikos (programavimo) uždavinys.



40. Geležinkelio elektrifikavimas

Vyksta geležinkelio atkarpos tarp dviejų stočių elektrifikavimas. Elektros laidų stulpai turi būti statomi vienodais atstumais: pirmasis stulpas ties pirmąja stotimi, o paskutinis - ties antrąja stotimi. Stulpus stato robotas pagal pateiktą programą.

Atstumas tarp stočių yra d metrų, robotas turi pastatyti n stulpų.



Papildykite programą trūkstamais elementais, kad robotas galėtų atlikti užduotį.

Nutempkite teisingus reiškinius į atitinkamas programos vietas.

Paaiškinimas

Atsakymas: n , $n-1$, $d/(n-1)$

Sąlygoje nurodyta pastatyti n stulpų, todėl imama n stulpų.

Padėjus pirmąjį stulpą, lieka $n - 1$ stulpas, todėl robotui teks $n - 1$ kartą kartoti tą patį, t. y. kiekvieną kartą pastatyti po stulpą.

Atstumą d reikia dalinti iš $n - 1$ karto, nes pirmas stulpas yra pirmojoje stotyje. Padalijus gaunamas atstumas, kuris turi būti tarp dviejų gretimų stulpų.

Tai informatika!

Procedūros, programų veiksmų sekos - tai veiksmingi programavimo įrankiai, atliekant konkrečią užduotį. Reikšmės procedūroms perduodamos naudojant kintamuosius arba parametrus. Ta pati procedūra gali būti naudojama programoje daug kartų, kai kreipiantis į procedūrą perduodamos skirtingos kintamųjų reikšmės.

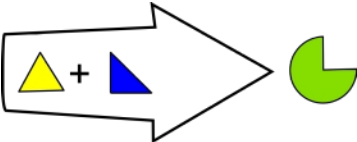
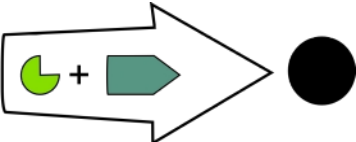
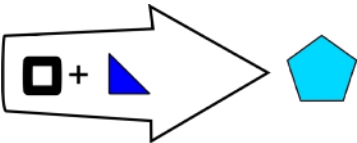
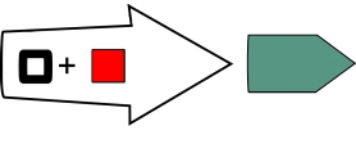
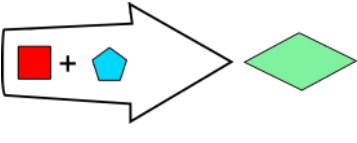
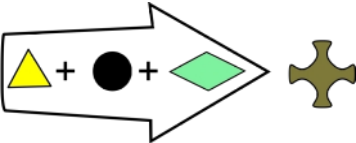
41. Mokslininkų eksperimentai

Trys mokslininkai eksperimento pradžioje turi tik keturias chemines medžiagas:    

Jų tikslas – sukurti naują medžiagą: 

Kiekvienas mokslininkas dirba dvi dienas. Kiekvieną dieną dirba tik po vieną mokslininką, kuris gali atlikti tik po vieną cheminę reakciją.

Viena reakcija – tai iš dviejų cheminių medžiagų sukūrimas naujos medžiagos, kurią vėliau galima panaudoti neribotame skaičiuje kitų reakcijų. Kiekvienas mokslininkas gali atlikti tik dviejų tipų reakcijas:

Chemikė Celestina (C)		
Fizikė Fausta (F)		
Biologas Baris (B)		

Žemiau nurodytos raidžių sekos, kurios aprašo mokslininkų darbo eiliškumą.

Kuriuo eiliškumu dirbdami mokslininkai negalės sukurti medžiagos  ?

A) C F B F C B

B) F F C C B B

C) C F B C F B

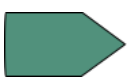
D) F B F C C B

Paiškinimas

Atsakymas: C.

Antrosios B (biologo Bario) reakcijos rezultatas – galutinis eksperimento produktas, todėl jis turi būti

nurodytas darbų eigoje. Kad galėtų įvykdyti antrąją reakciją, jam reikia dviejų medžiagų:  .

Kad gautų juodąjį apskritimą , C (chemikė Celestina) turi turėti  ir  bei dirbti laboratorijoje po F (fizikė Fausta). Taip pat C turi pirmiau įvykdyti savo pirmąją reakciją prieš atlikdama antrąją. F turi pirmiau įvykdyti savo pirmąją reakciją, kad B (biologas Baris) galėtų atlikti savo pirmąją reakciją.

Atsakyme C, F vienos reakcijos metu gali sukurti tik vieną medžiagą reikalingą kitoms dviem reakcijoms, todėl viena iš vėliau einančių reakcijų negalėtų įvykti. Dėl šios priežasties galutinė medžiaga negalėtų būti sukurta eiliškumu C F B C F B.

Tai informatika!

Informatikoje bei programavime mes turime procesus, kuriuose naudojame įvesties duomenis (angl. input), kad gautume išvestį (angl. output), t.y. rezultatą. Svarbu suprasti, kad proceso negalima pradėti tol, kol nėra įvesties duomenų, kurie yra kitų procesų rezultatai. Programuotojas turi atpažinti vieną nuo kito priklausančius procesus ir surikiuoti juos taip, kad visų susijusių procesų rezultatai būtų pasiekiami procesams, naudojantiems minėtus rezultatus įvestyje.

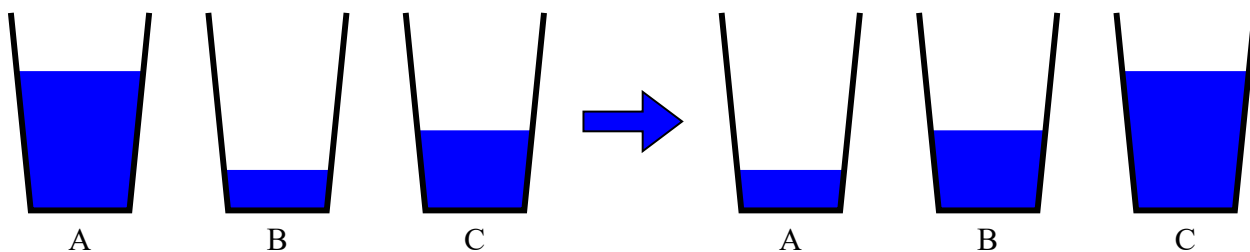


42. Vandens pilstymas

Trys indai A, B ir C, kuriuose yra skirtingi vandens kiekiai, sudėti į eilę. Pradžioje nė vienas indas nėra pilnas. Ant indų nėra kiekio skalės žymių. Tačiau yra įmanoma palyginti vandens kiekį induose. Galima naudoti tik toliau išvardytas leidžiamas operacijas (vieną ar daugiau iš jų) (atkreipkite dėmesį, kad ne visada visas operacijas įmanoma atlikti):

- „ištuštinimas“: paimkite indą ir supilkite visą jo turinį į kitą indą;
- „išlyginimas“: paimkite indą ir perpilkite iš jo į kitą indą tokį kiekį, kad tame antrame inde būtų tiek pat vandens, kiek yra trečiajame.
- „pripildymas“: paimkite indą ir supilkite kiek įmanoma daugiau jo turinio į kitą indą, kad kitas indas būtų pilnas.

Perpylimo užduotis yra pakeisti pradinę indų būseną į tokią, kaip parodyta pateiktame paveikslėlyje, nenaudojant papildomų indų. Galima taikyti tik aprašytas leidžiamas operacijas.



Po operacijų A inde turėtų būti vandens kiekis, kuris iš pradžių buvo inde B, inde B turėtų būti vandens kiekis, kuris iš pradžių buvo inde C, o inde C turėtų būti vandens kiekis, kuris iš pradžių buvo A inde.

Kuris iš šių teiginių yra teisingas?

- A) Norimą rezultatą galima pasiekti neatlikus operacijos „ištuštinimas“
- B) Norimą rezultatą galima pasiekti neatlikus operacijos „išlyginimas“
- C) Norimą rezultatą galima pasiekti neatlikus operacijos „pripildymas“
- D) Naudojant šias 3 operacijas negalima pasiekti norimo rezultato

Paaiškinimas

Atsakymas: D.

Atminkite, kad atlikus bet kurį teigiamą skaičių šių trijų operacijų, gausime, kad turi galioti bent viena sąlyga:

- Vienas iš indų yra tuščias
- Dviejuose induose vandens yra tiek pat
- Vienas iš indų yra pilnas

Reikiama galutinė būsena neturi nė vienos iš šių savybių, todėl transformacija yra neįmanoma.

Tai informatika!

Trys užduotyje atliekamos operacijos yra informacijos praradimo operacijų pavyzdžiai. Vienas iš būdų sužinoti, ar informacija išsaugoma po operacijos, yra tai, ar operaciją galima pakeisti.

Pradinę būseną galima pavaizduoti kaip tris skirtingus skaičius, didesnius nei 0 ir mažesnius nei 1. Jei atliksite ištuštinimo operaciją, vienas iš šių skaičių virsta nuliu. Jei atliksite išlyginamąją operaciją, du iš skaičių taps vienodi. Jei atliksite pripildymo operaciją, vienas iš skaičių virsta fiksuota verte 1. Kiekvienu atveju negalima atšaukti operacijos, kad būtų galima gauti pradinį skaičių.

Norėdami įsitikinti, kad svarbi informacija yra išsaugota, kompiuterių programuotojai turi gerai apgalvoti savo programų operacijas.

Gerai žinomas pavyzdys, kai programavimo metu reikia išsaugoti informaciją, yra dviejų kintamųjų

reikšmių keitimas. Tarkime, kad kintamojo a reikšmė yra 0,5, kintamojo b reikšmė yra 0,2, o mūsų tikslas yra sukeisti reikšmes. Naivus ir neteisingas algoritmas yra:

a: = b;

b: = a;

Gautos a ir b reikšmės yra 0,2, o pradinė kintamojo a reikšmė 0,5 yra prarandama.

Programuotojas gali įvesti tarpinį kintamąjį c, kad išsaugotų pradinę reikšmę:

c: = b;

b: = a;

a: = c;

Be to, programuotojas algoritmą gali perrašyti tokiu būdu:

a: = a - b;

b: = b + a;

a: = b - a;

Tokiu atveju pradinių reikšmių skirtumas išlieka ir tai gali būti panaudota pradinėms reikšmėms perskaičiuoti antroje ir trečioje programos teksto eilutėse.

Atliekant užduotį kiekviena leistina operacija sukelia informacijos praradimą. Tarpiniai kintamieji taip pat neįtraukiami, kadangi nėra papildomų indų.

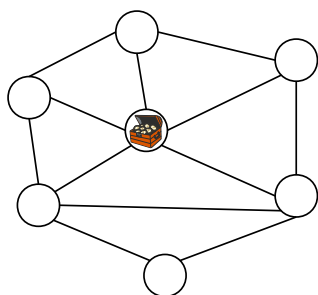
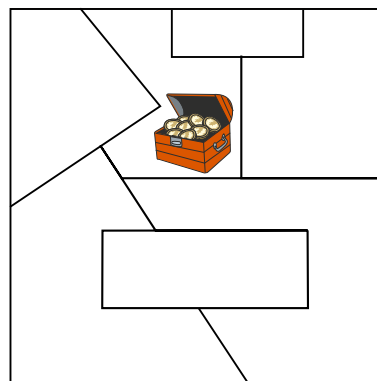
Kvantinis skaičiavimas yra dar viena informatikos sritis, kurioje svarbios informacijos išsaugojimo operacijos. Kvantiniame skaičiavime visos operacijos turi būti grįžtamos, kad būtų laikomos galiojančiomis operacijomis.



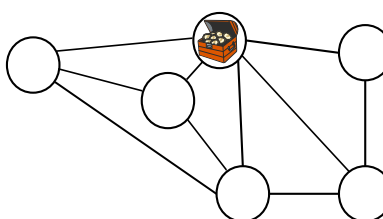
43. Šifruotas žemėlapis

Paveiksle vaizduojama bebrų šalis, kurioje yra 7 provincijos. Šios šalies valdovas paslėpė savo valdose lobį ir sukūrė šifruotą žemėlapi. Provincijos vaizduojamos apskritimais. Apskritimai tarpusavyje turi jungtį, jei juos atitinkančios provincijos turi bendras sienas. Kad suklaidentų vagis, valdovas sukūrė vieną teisingą žemėlapi ir tris klaidingus.

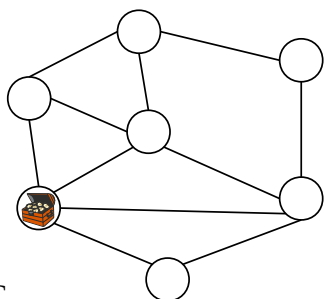
Kuris žemėlapis yra teisingas?



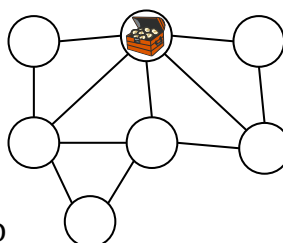
A



B



C



D

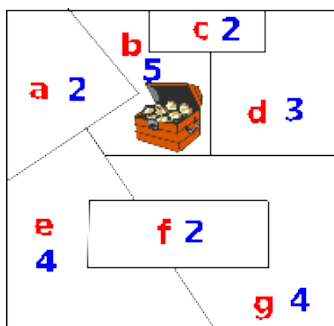
Paiškinimas

Atsakymas: D.

Žemėlapis vaizduoja 7 provincijas. B atvejis yra neteisingas, nes jame tik 6 apskritimai – trūksta vienos provincijos.

Provincija, kurioje yra lobis, ribojasi su penkiomis kitomis kaimynėmis. Todėl C atvejis taip pat neteisingas, nes jame nėra nei vienos provincijos, kuri turėtų 5 kaimynes.

Kad nustatytume, kuris žemėlapis – A ar D – teisingas, reikia panagrinėti sąryšius tarp apskritimų. Tam tikslui sužymėkime provincijas raudonomis raidėmis, o jų kaimynių skaičių – mėlynais skaičiais:



Provincija a turi 2 kaimynes (b ir e).

Provincija b turi 5 kaimynes (a, c, d, e ir g).

Provincija c turi 2 kaimynes (b ir d).

Provincija d turi 3 kaimynes (b, c ir g).

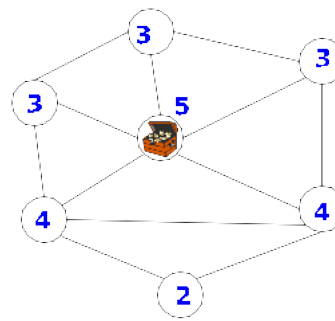
Provincija e turi 4 kaimynes (a, b, f ir g).

Provincija f turi 2 kaimynes (e ir g).

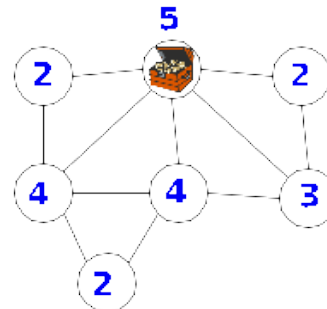
Provincija g turi 4 kaimynes (b, d, e ir f).

Iš viso trys provincijos turi po 2 kaimynes, viena – 3 kaimynes, dvi – po 4 kaimynes ir viena provincija turi 5 kaimynes.

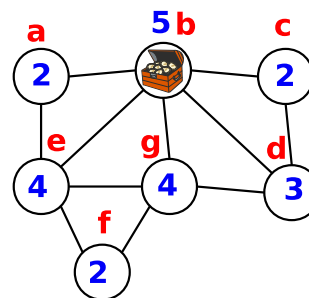
Skaičiuojant kaimynes A žemėlapyje, matome tris provincijas, turinčias po 3 kaimynes ir tik vieną provinciją su 2 kaimynėmis. Taigi, A yra neteisingas žemėlapis.



Skaičiuojant D žemėlapyje kaimynes, matome, kad jos atitinka pradinio žemėlapyje reikalavimus: trys provincijos turi po 2 kaimynes, viena – 3 kaimynes, dvi – po 4 kaimynes ir viena provincija turi 5 kaimynes.



Galutiniam įsitikinimui surašykime šalia provincijas vaizduojančių apskritimų galimas raides ir kaimynių skaičius.



Tai informatika!

Šis uždavinys – tai grafų naudojimo pavyzdys, kai sprendžiame tikrovę atitinkančius uždavinius. Grafas turi viršūnes (apskritimai) ir briaunas (atkarpos) tarp jų. Briaunos nusako ryšius tarp viršūnių. Šiame uždavinyje viršūnės atitinka provincijas, o briaunos žymi, ar dvi provincijos turi bendrą sieną. Grafas yra galingas instrumentas norint nusakyti ryšius tarp įvairių tikrovę atitinkančių objektų (šiuo atveju, provincijų). Matematikai ir programuotojai sukūrė daug naudingų algoritmų, kuriuos galima pritaikyti grafams (šiuo pavyzdyje tai yra viršūnių skaičiavimas).

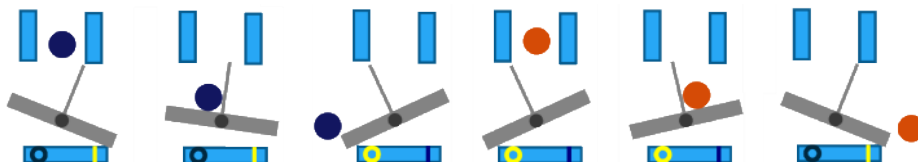
Daugiau informacijos apie grafus

[https://en.wikipedia.org/wiki/Graph_\(discrete_mathematics\)](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics))

	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
						12

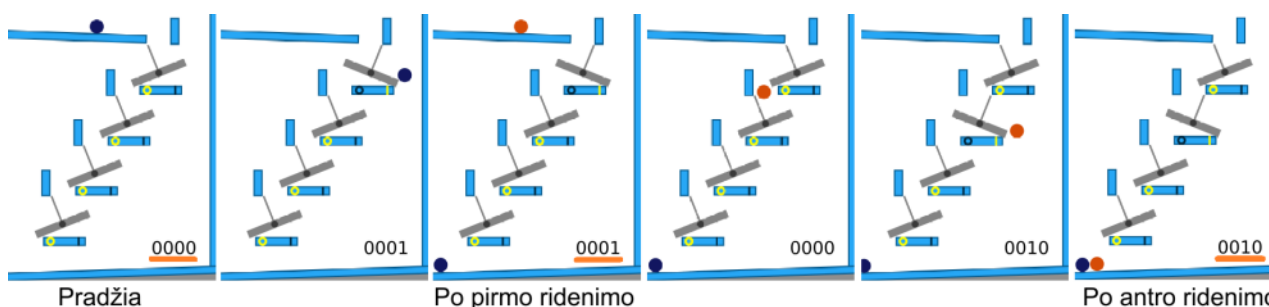
44. Dvejetainis skaitiklis

Turime mechaninę sistemą, kurią sudaro viena po kitos išdėstytos keturios svirtys. Jos gali būti pasuktos į kairę (būsena žymima 0) arba į dešinę (žymima 1). Kai iš viršaus atrieda kamuoliukas ir užkrenta ant svirties, ji pasisuka į kitą būseną, ir kamuoliukas nurieda toliau. Sistemos svirtys išdėstytos taip, kad į dešinę nuriedėjęs kamuoliukas kitų svirčių nebeužkabina.



Sistemos būseną apibūdina visų svirčių būsenos, surašytos iš apačios į viršų. Pradinė būsena yra 0000. Žemiau pavaizduota sistemos būsena po pirmų dviejų kamuoliukų ridenimo.

Kokia bus sistemos būsena po 5 kamuoliukų ridenimo? Įveskite 5 skaitmenis.



Paiškinimas

Atsakymas: 0101

Visa būsenų seka po penkių kamuoliukų ridenimo yra tokia: 0001, 0010, 0011, 0100, 0101

Tai mechaninis dvejetainis skaitiklis. Pradinė jo būsena yra 0000.

Pirmasis kamuoliukas pasuka 1 svirtį į dešinę. Skaitliklio būsena yra 0001.

Antrasis kamuoliukas pasuka 1 svirtį į kairę ir 2 svirtį į dešinę. Skaitliklio būsena yra 0010.

Trečiasis kamuoliukas pasuka 1 svirtį į dešinę ir nurieda į apačią. 2 svirtis išlieka pasvirusi į dešinę.

Skaitliklio būsena yra 0011.

Ketvirtasis kamuoliukas pasuka 1 svirtį į kairę, 2 svirtį į kairę ir 3 svirtį į dešinę. Skaitliklio būsena yra 0100.

Paskutinis penktasis kamuoliukas pasuka 1 svirtį į dešinę ir nurieda į apačią. Kitų svirčių padėties lieka nepakitusios. Skaitliklio būsena yra 0101.

Galima įžvelgti dėsninę sistemą. Kamuoliukas krenta ant svirčių tol, kol sutinka padėtyje 0 esančią svirtį. Tą svirtį pasuka į dešinę ir nurieda į apačią. Visos prieš tai buvusios svirtys pasukamos į kairę, jų būsenos tampa 0.

Tai informatika!

Skaiciuojant tik dviem skaitmenimis (0 ir 1), naudojama dvejetainė skaičiavimo sistema. Kadangi kompiuterių viduje esantys jungikliai turi dvi būsenas ir gali būti tik įjungti (1) arba išjungti (0), patogu naudoti dvejetainius duomenis. Šiuolaikiniuose kompiuteriuose yra daugybė jungiklių, kurie valdo visus kompiuterių naudojamus duomenis.

Ši lengvai pagaminama sistema – mechaninis dvejetainis skaitiklis, turintis 4 skaitmenis ir galintis skaičiuoti iki 15. Žemiau pateikiama visų 4 skaitmenų skaičių lentelė.

Uždavinių autoriai

Tony René Andersen
Haim Averbuch
Michelle Barnett
Michael Barot
Wilfried Baumann
Jan Berki
Linda Bergsveinsdóttir
Daniela Bezáková
Laura Braun
Špela Cerar
Mony Chanroath
Marios Choudary
Anton Chukhnov
Sébastien Combéfis
Kris Coolsaet
Allira Crowe
Andrew Csizmadia
Valentina Dagienė
Christian Datzko
Maria Suyana Datzko
Sarah Estrella Datzko
Susanne Datzko
Guillaumede Moffarts
Lanping Deng
Marissa Engels
Olivier Ens
Gerald Futschek
Sonali Gogate
Arnheiður Guðmundsdóttir
Martin Guggisberg
Vernon Gutierrez
Juraj Hromkovič
Alisher Ikramov
Thomas Ioannou
Tiberiu Iorgulescu

Takeharu Ishizuka
M. Faiz AhmadIsmail
Yong-ju Jeon
Felipe Jiménez
Anna Laura John
Mile Jovanov
Ungyeol Jung
Ilya Kaysin
Adem Khachnaoui
Injoo Kim
Jihye Kim
Vaidotas Kinčius
Mária Kiss
Jia-Ling Koh
Sophie Koh
Dennis Komm
Anja Koron
Bohdan Kudrenko
Regula Lacher
Anh Vinh Lê
Greg Lee
Inggriani Liem
Judith Lin
Lynn Liu
Violetta Lonati
Vũ Văn Luân
Karolína Mayerová
Mattia Monga
Samart Moodleah
Anna Morpurgo
Madhavan Mukund
Tom Naughton
Pia Niemelä
Tomohiro Nishida
Assylkan Omashev

Henry Ong
Margot Phillipps
Zsuzsa Pluhár
Wolfgang Pohl
Sergei Pozdniakov
Stavroula Prantsoudi
Nol Premasathian
J. P. Pretti
Milan Rajković
Chris Roffey
Andrea Schrijvers
Eljakim Schrijvers
Humberto Sermenó
Vipul Shah
Daigo Shirai
Taras Shpot
Jacqueline Staub
Nikolaos Stratis
Gabrielė Stupurienė
Maciej M. Sysło
Bundit Thanasopon
Monika Tomcsányiová
Peter Tomcsányi
Nicole Trachsler
Jiří Vaníček
Troy Vasiga
Ela Veza
Márton Visnovitz
Florentina Voboril
Michael Weigend
Jing-Jing Yang
Xing Yang
Khairul A. Mohamad Zaki

Airija, Australija, Austrija,
Azerbaidžanas, Baltarusija,
Belgija, Bosnija ir Hercegovina,
Brazilija, Bulgarija, Čekija, D. Britanija,
Egiptas, Estija, Graikija, Honkongas,
Indija, Indonezija, Iranas, Islandija,
Ispanija, Italija, Izraelis, Japonija,
Jungtinės Amerikos valstijos, Jordanija,
Kanada, Kazachstanas, Kinija, Kipras,
Kroatija, Latvija, Lenkija, Lietuva,
Malaizija, Malta, Mongolija, Naujoji
Zelandija, Nigerija, Nyderlandai,
Pakistanas, Pietų Afrikos respublika, Pietų
Korėja, Portugalija, Prancūzija, Rumunija,
Rusija, Serbija, Singapūras, Slovakija,
Slovėnija, Suomija, Šiaurės Makedonija,
Švedija, Šveicarija, Tailandas,
Taivanas, Tunisas, Turkija, Ukraina,
Vengrija, Vietnamas, Vokietija.

Šalys, kuriose
vyksta konkursas