

ISSN 2783-7599



2024

www.bebas.lt



Bebras[®]

Informatikos ir informatinio
mąstymo uždavinių rinkinys

Šiame rinkinyje pateikiama XXI informatikos ir informatinio mąstymo konkurso (iššūkio) „Bebras“ I etapo, vykusio 2024 metų rudenį, užduotys jų atsakymai ir paaiškinimai, koks informatikos turinys ir konceptai atskleidžiami, kaip ir kuo uždavinys įdomus ugdant informatinį mąstymą. Visos užduotys (įskaitant grafiką ir kitą medžiagą) licencijuojamos pagal Kūrybinių bendrijų licenciją – „Creative Commons Attribution-ShareAlike 4.0 International License“. Šis užduočių rinkinys skiriamas 1–12 klasių mokinių informatinio mąstymo gebėjimams ugdyti.

„Bebro“ konkursas rengiamas kasmet lapkričio–vasario mėnesiais įvairaus amžiaus vaikų grupėms nuo 6 iki 19 metų. Konkurso tikslas yra skatinti moksleivių domėjimąsi informatika, ugdyti gilesnį technologijų supratimą, skatinti gebėjimą spręsti algoritminius, loginius uždavinius, ugdyti kritinį mąstymą, programavimo ir kompiuterinio raštingumo įgūdžius, taip pat pritraukti daugiau gabių jaunuolių į informatikos studijas.

Dėkojame Daumilui Ardickui, Daivai Gaučytei, Nojui Gudinavičiui, Karoliui Jasučiiui, dr. Tatjanai Jevsikovai, Alvidai Lozdienei, Vaidai Masiulionytei-Dagienei, dr. Broniui Skūpui, dr. Gabrielei Stupurienei, Indrai Sudeikienei, Rimantui Žakauskui, talkinusiems verčiant ir adaptuojant užduotis. Taip pat dėkojame tarptautinei „Bebro“ konkurso bendruomenei ir uždavinių autoriams.

Sudarė Vaiva Žukauskaitė
Konsultavo Valentina Dagienė
Redagavo Viktoras Dagys
Viršelį kūrė Vaidotas Kinčius



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

Įvadas

„Bebras“ yra integruotas informatinio ugdymo modelis, kurio tikslas – populiarinti informatikos mokslą ir skatinti informatinį mąstymą (angl. Computational thinking) ne tik įvairaus amžiaus mokiniams, bet ir mokytojams bei visuomenei.

Lietuvoje 2024 metų „Bebro“ uždavinius sprendė 60 424 mokiniai:

2 185 nykštukai (1–2 klasės),
4 528 mažyliai (3–4 klasės),
17 602 bičiuliai (5–6 klasės),
17 330 draugų (7–8 klasės),
15 246 jaunieji (9–10 klasės),
3 533 kolegos (11–12 klasės).

2024 metais Lietuvoje kiekviena amžiaus grupė sprendė po 18 uždavinių, išskyrus nykštukus (12 uždavinių) ir mažylius (15 uždavinių): trečdalis buvo lengvesnių, trečdalis – vidutinio sunkumo ir trečdalis – sunkesnių. Uždaviniams spręsti skiriamos 45 minutės.

Lietuvoje taškai skaičiuojami taip:

- Prieš pradėdamas spręsti, kiekvienas dalyvis turi 54 taškus (mažylių grupėje – 36 taškus; 18 uždavinių $\times$ 3 arba 12 uždavinių $\times$ 3);
- Už teisingai išspręstą užduotį skiriama 6, 9 arba 12 taškų (priklausomai nuo užduoties sunkumo lygio);
- Už neišspręstą užduotį – 0 taškų;
- Už klaidingą atsakymą atimamas trečdalis užduočiai skiriamų taškų, t. y., atitinkamai 2, 3 arba 4 taškai.

Daugiau informacijos apie „Bebro“ konkursą pateikiama interneto svetainėse:

- Tarptautinė „Bebro“ iššūkio svetainė: www.bebbras.org
- Lietuvos „Bebro“ svetainė: bebras.lt
- Konkurso sistema: lt.bebbras.lt



Lentelėje pateikiamas šio rinkinio uždavinių skirstymas pagal amžiaus grupes. Skaičius langelyje prie uždavinio nurodo sudėtingumo lygį:

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



Kiekvieno uždavinio viršuje nurodoma, kuriai amžiaus grupei jis skiriamas ir jo sudėtingumo lygis. Pustlapio apačioje pateikiamas uždavinio atsakymas ir paaiškinimas, kaip uždavinys susijęs su informatika.

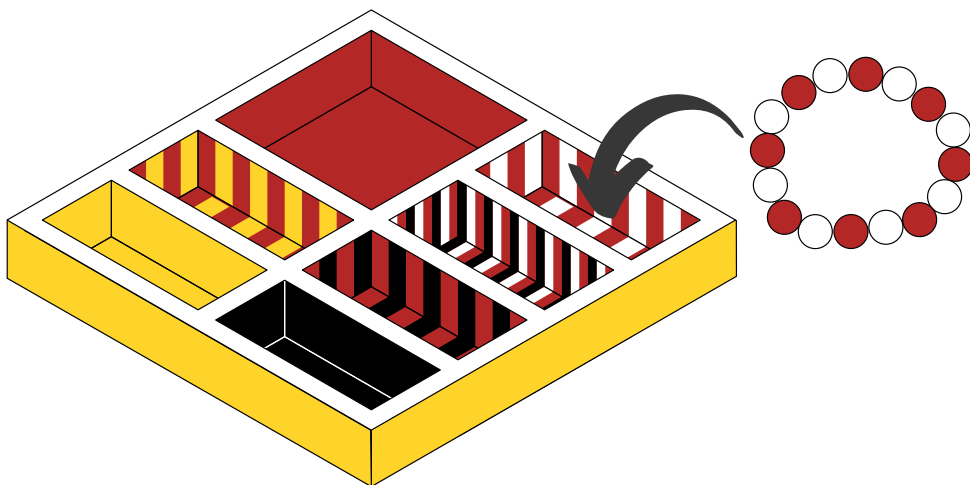
Nr.	Uždavinio pavadinimas	Uždavinio identifikatorius	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
1	Apyrankių rūšiavimas	2024-BR-04	6	6				
2	Indai lentynoje	2024-LT-01	6					
3	Pabirę pieštukai	2024-LT-06	6					
4	Kortelės	2024-LT-07	6					
5	Kortelės su baubais	2024-CH-03b	9	6				
6	Kelias	2024-ID-04	9	6				
7	Pabirę lapeliai	2024-IE-01b	9	6				
8	BBQ Vakarėlis	2024-KR-02	9	6				
9	Spalvinimo lapas	2024-FI-01	12	9	6			
10	Skaidrus vamzdelis	2024-KR-03a	12	9	6			
11	Gimtadienio dovana	2024-NL-02	12	9	6			
12	Švyturių žemėlapis	2024-CZ-05	12		6			
13	Burvaltės piešimas	2024-PL-03		9	6			
14	Oliverio barškutis	2024-SK-01b		9	6			
15	Picų vakarėlis	2024-DE-02		12	9	6		
16	Stebuklingoji gėlė	2024-DE-03		12	9	6		
17	Savaitgalis prieplaukoje	2024-MT-02		12	9	6		
18	Klasiokų skambutis	2024-MY-03		12	9	6		
19	Kelias robotui	2024-PL-04		12	9	6		
20	Kortelės su baubais (interaktyvus)	2024-CH-03a			9	6		
21	Knygų gražinimas	2024-SI-01			12	9	6	6
22	Saulėtos dienos	2024-DE-04a			12	9	6	

23	Piešiame žaidimą	2024-DE-07			12	9	6	
24	Žygis miške	2024-DE-08			12	9	6	
25	Tiesa apie bebrą Alfredą	2024-HU-02			12	9	6	
26	Kortelių piešiniai	2024-IN-04			12	9	6	
27	Sąrašas	2024-AT-03				12	9	6
28	Žodžių grandinės	2024-CA-02				12	9	6
29	Balionų mašina	2024-DE-06a				12	9	6
30	Lobis	2024-HU-03				12	9	6
31	Ilgiausia apyrankė	2024-TW-03				12	9	6
32	Statybinių blokelių mašina	2024-BE-01a				12	9	
33	Kamuoliukai	2024-BG-01b					12	9
34	Maisto žemėlapis	2024-IT-01b					12	9
35	Draugai	2024-LT-05					12	9
36	Geležinkelio bėgių tinklas	2024-PK-03					12	9
37	Raidžių seka	2024-SK-04					12	9
38	Piešiantis robotas	2024-AU-03					12	
39	Saulėtos dienos	2024-DE-04b						9
40	Bebrų karaliaus lobis	2019-VN-04						12
41	Paveikslų šifravimas	2024-AU-01						12
42	Statybinių blokelių mašina	2024-BE-01b						12
43	Tyrinėjimas	2024-DE-05						12
44	Plytų tvora	2024-FI-03						12
45	Palago	2024-HU-04						12

1. Apyrankių rūšiavimas

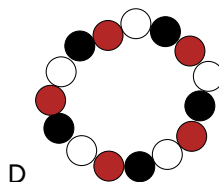
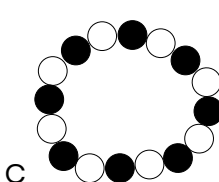
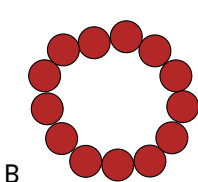
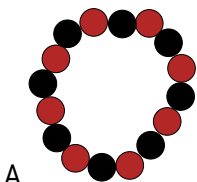
2024-BR-04

Viktorija laiko savo apyrankes dėžutėje su septyniais skyriais. Kiekvieną apyrankę ji deda į skyrių, kuris atitinka tos apyrankės spalvų raštą. Paveikslėlyje parodyta, kur ji deda vieną iš savo apyrankių.



Kuri iš šių apyrankių NEatitinka nė vieno dėžutės skyriaus?

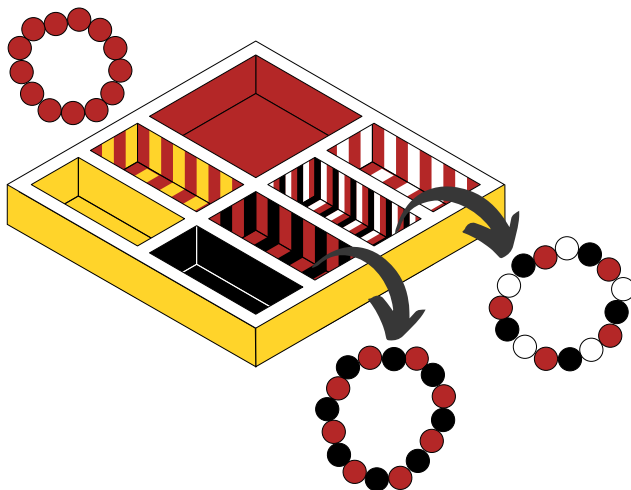
Atsakymų variantai:



Teisingas atsakymas: C

Paiškinimas

Paveikslėlyje pavaizduoti dėžutės skyriai, į kuriuos Viktorija deda apyrankes iš (A), (B) ir (D) variantų. Atkreipkite dėmesį, kad apyrankės ir skyriai yra to pačio spalvų rašto. Be to, Viktorijos dėžutėje nėra skyrelio su juodos ir baltos spalvų raštu, taigi, nėra skyrelio, kuriam būtų galima priskirti (C) variante pateiktą apyrankę.



Tai informatika!

Bandydami išspręsti uždavinį ar problemą dažnai susiduriame su daugybe duomenų. Šioje užduotyje apyrankės turi daug savybių, kuriomis remdamiesi galime apyrankes sudėti į tinkamus skyrelius. Svarbu gebėti atskirti mums nereikalingus duomenis, kad būtų lengviau atsirinkti svarbius duomenis ir surasti sprendimą. Šioje užduotyje mums reikia žiūrėti tik į spalvas, nesvarbu, kokios formos yra karoliukai, kiek jų yra ar kokia tvarka jie išsidėstę. Norint išspręsti užduotį mums net nereikia matyti visos apyrankės, užtenka pamatyti tik dalį paveikslėlio.

2024-LT-01

A circular diagram illustrating the water cycle using household items. At the top is a yellow mug with black spots. An arrow points from the mug to a yellow bowl with black spots on the right. From the bowl, an arrow points to a yellow teapot with black spots at the bottom. From the teapot, an arrow points back to the bowl. Finally, an arrow points from the bowl back to the mug, completing the cycle.

Teisingas atsakymas: C

Paiškinimas

Lentynoje C indai sudėti neteisingai.

Ar pastebėjote, kad lentynoje iš eilės padėtų arbatinukų skaičius turi būti lyginis? Taip yra todėl, kad pagal schemą arbatinukų dėjimas į lentyną kartojasi: kai padedamas pirmas arbatinukas, turi būti padedamas ir antras, kai padedamas trečias arbatinukas, turi būti dedamas ketvirtas ir t. t. Kitaip negalima pereiti prie kito schemoje pavaizduoto žingsnio.

Tik C variante iš eilės padėtų arbatinukų skaičius yra nelyginis, todėl tokia seka neatitinka schemoje pavaizduotų taisyklių. Žinoma, reikia atidžiai patikrinti A, B ir D lentynose esančius indus, ar jie sudėti tiksliai pagal pateiktą schemą.

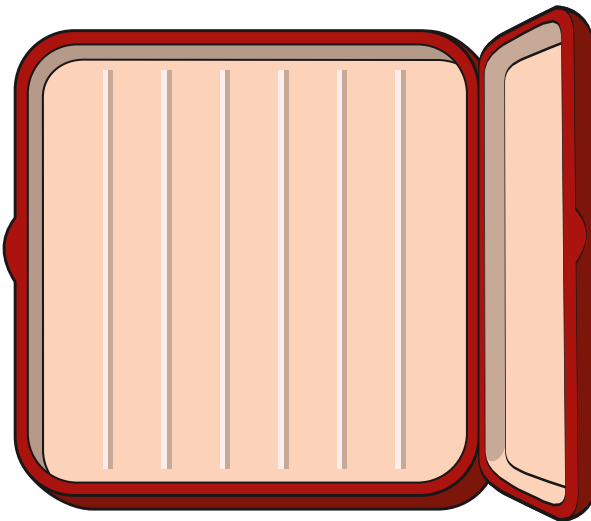
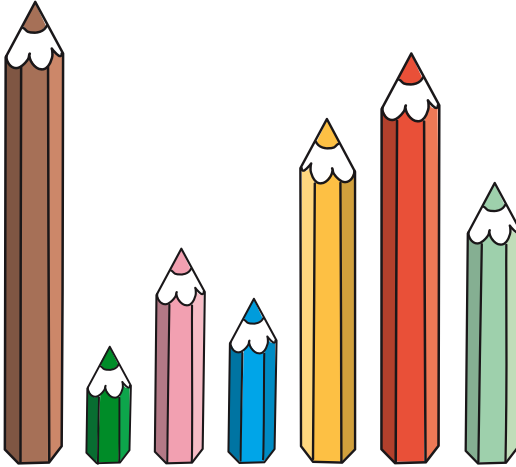
Tai informatika!

Tai klasikinis informatikos uždavinys, kai pateikiama schema (šiuo atveju tai – automatas) ir reikia išsiaiškinti, kaip ji veikia ir kokius rezultatus gali pateikti. Lengviausias būdas tai padaryti – panagrinėti pateiktus pavyzdžius ir pastebėti dėsningumus. Taigi, ieškant varianto, kuriame indai būtų sudėti ne pagal pateiktą schemą, daugiausia dėmesio skiriama vienai svarbiausių schemos dalių – kartojimo veiksmui (arbatinukų išdėstymui). Šioje užduotyje arbatinukų skaičius indų eilėje visada turi būti lyginis.

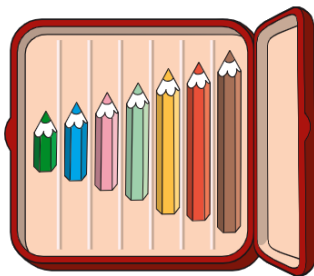
3. Pabirę pieštukai

2024-LT-06

Pabiro Andriaus pieštukai, padėkite jam sudėti pieštukus į penalą nuo mažiausio (kairėje) iki didžiausio (dešinėje).



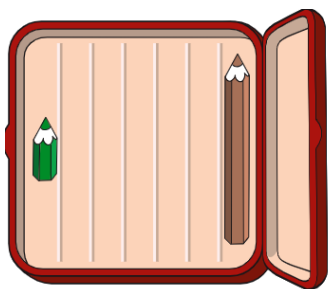
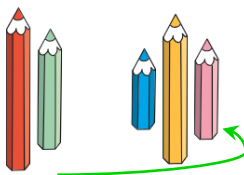
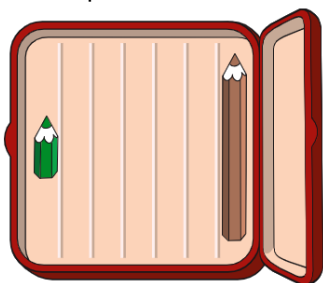
Teisingas atsakymas



Paiškinimas

Tai rikiavimo uždavinys ir jis gali būti sprendžiamas įvairiai. Suprantama, kad uždavinys prašo išrikiuoti pieštukus pagal jų aukštį. Pavyzdžiui, lengva pastebėti, kad rudas pieštukas yra aukščiausias, o tamsiai žalias – trumpiausias.

Panašaus dydžio pieštukus galima palyginti vieną pieštuką nutempus prie jam artimo aukščio pieštuko.



Tai informatika!

Mokydamiesi informatikos pradinių klasių mokiniai susipažįsta, kaip rinkti ir rūšiuoti duomenis pagal vieną ar du nurodytus požymius. Daiktai grupuojami, klasifikuojami (rūšiuojami) pagal vieną ar kelis nurodytus požymius (spalvą, dydį, formą, padėtį erdvėje ir kt.), daiktai dėliojami į eilę (rikiuojami) pagal tą patį požymį (spalvos intensyvumą, laiko kaitos sąvokas: vakar, šiandien, rytoj; praėjusį, dabar, ateityje; para, savaitė, ir pan.). Anglų kalboje rūšiavimas ir rikiavimas yra vadinamas vienodai sorting.

Šiame uždavinyje reikia surikiuoti pieštukus jų aukščių didėjimo tvarka (iš kairės į dešinę). Pieštukai penale dėliojant vaizduojami ne horizontaliai, o įstriža eile.

Taip ugdomas ir erdvinis mąstymas.

4. Kortelės

2024-LT-07

Rita turi šešias korteles ir dėlioja jas vieną ant kitos. Kuri kortelė padėta KETVIRTA?



Teisingas atsakymas: ketvirta padėta kortelė su stalo kalendoriumi.



Paiškinimas

Reikia atidžiai pažiūrėti, kuri kortelė guli labiausiai apačioje. Matome, kad tai kortelė su namuku. Toliau apžiūrime, kas guli ant jos – tai yra kortelė su medžiu. Ant jos šonu prigludusi trečia kortelė – su tvorele. Ant tvorelės padėta ketvirta kortelė – su stalo kalendoriumi, toliau – kortelė su puodeliu ir pagaliau paskutinė kortelė – su knyga.

Tai informatika!

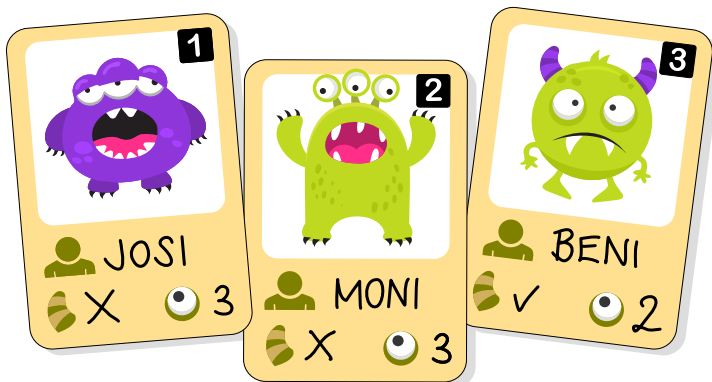
Mokydamiesi informatikos pradinį klasių mokiniai susipažįsta su duomenų sąvoka. Aptariami kasdienės aplinkos duomenų pavyzdžiai: batų numeriai, drabužių dydžiai, prekių kainos parduotuvėje, mokyklos langų skaičius, turimų pieštukų rūšiavimas ir rikiavimas pagal spalvas, surinktų sėklų ar lapų skirstymas pagal rūšis, lego kaladėlių grupavimas pagal formą ir pan. Informatikoje dažnai naudojama duomenų struktūra, veikianti pagal principą „paskutinis į maišą, pirmas iš maišo“. Tokia struktūra vadinama dėklu (steku, ang. *stack*).

Šiame uždavinyje reikia paveikslėlyje atpažinti kortelių sudėliojimo tvarką ir skaičiuoti nuo pirmosios padėtos kortelės. Tai atitinka dėklo principą: paskutinė padėta kortelė bus imama pirmoji.

5. Kortelės su baubais

2024-CH-03b

Barbora renka korteles su baubų paveikslukais. Nagrinėsime šias baubų savybes: vardą, ragų ir akių skaičių. Visi baubai yra skirtingi, todėl jų savybių reikšmės skiriasi. Štai trys kortelių pavyzdžiai, kuriuose pateiktos baubų savybių reikšmės:



Kiekvienai savybei jos reikšmė nurodoma taip:

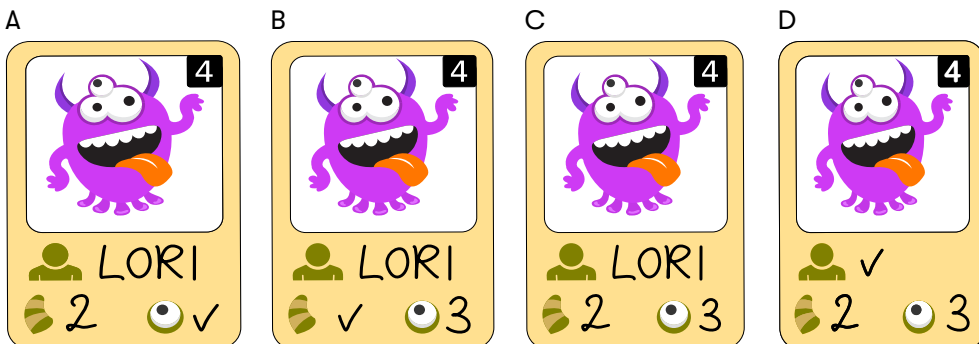
 Reikšmės
pateikiamos tekstu.

 Reikšmės
pateikiamos X arba ✓.

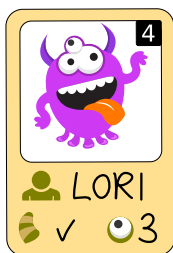
 Reikšmės
pateikiamos skaičiumi.

Kurioje kortelėje visos savybių reikšmės nurodytos teisingai?

Atsakymų variantai:



Teisingas atsakymas: B



Paaiškinimas

Pateiktoje lentelėje parodyta, kodėl atsakymas yra B:

	Reikšmės			
Savybė	A	B	C	D
	LORI	LORI	LORI	✓
	2	✓	2	2
	✓	3	3	3

Tik B kortelės visų savybių reikšmės yra teisingų rūšių.

- ☐ A, C ir D kortelėse pabaisa LORI turi 2 ragus, bet ragų  reikšmė gali būti tik X arba ✓.
- ☐ A kortelėje pabaisa LORI turi akis, bet akių  reikšmė turi būti akių skaičius, šiuo atveju 3.

Tai informatika!

Kortelėse esančios reikšmės apibūdina pabaisas. Pabaisų savybių reikšmės gali būti laikomos kiekvienos kortelės kintamaisiais. Kiekvienas kintamasis turi su juo susietą duomenų tipą, kuris naudojamas nustatyti, kokio tipo reikšmės yra priimtinos arba ne. Parodytose pabaisų kortelėse pavadinimo reikšmės tipas yra tekstas, akių – skaičius, o ragų – **taip** arba **ne**.

Priklausomai nuo to, kaip duomenys vaizduojami, kompiuteris gali atlikti skirtingus skaičiavimus (arba operacijas). Prieš vaizduodami duomenis kompiuteryje, turime apgalvoti, ką norėsime vėliau su jais daryti ir pagal tai nuspręsti, kokį duomenų tipą naudoti. Duomenų tipas ne tik nusako, kokios reikšmės yra leistinos, bet taip pat nurodo, kokias operacijas galima atlikti su šiomis reikšmėmis.

6. Kelias

2024-ID-04

Lina, eidama į mišką, pakeliui pamatė šiuos dalykus:

pirmiausia pamatė raudoną ir geltoną gėlę,



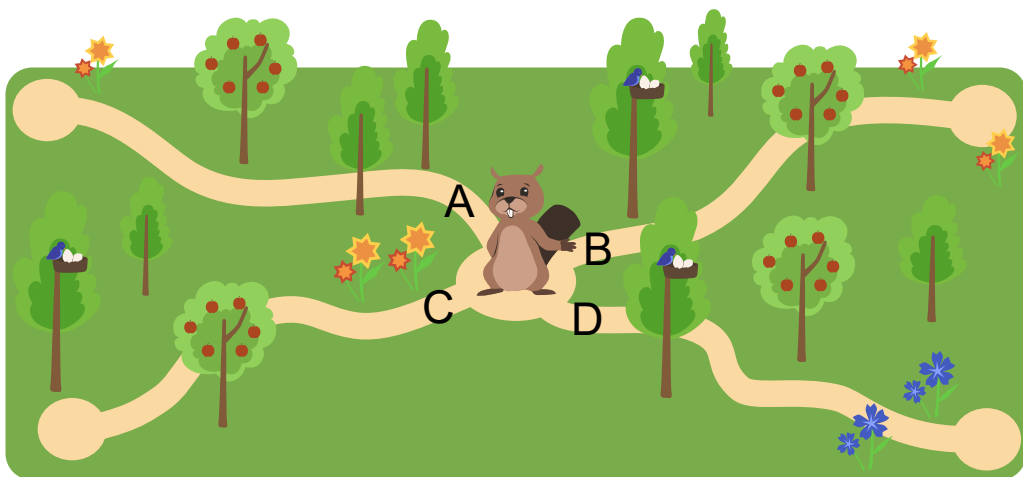
po to – obelį,



o galiausiai – medį su paukščio lizdu.



Kuriuo keliu Lina atėjo į mišką?



Teisingas atsakymas: B.

Paiškinimas

Pakeliui į mišką Lina pirmiausia pamatė raudoną ir geltoną gėlę, po to – obelį, o dar vėliau – medį su lizdu. Pakeliui namo ji matys tuos pačius objektus, tik priešinga tvarka. Taigi, eidama namo ji pirmiausia pamatys medį su lizdu, tada – obelį, o paskui – raudoną ir geltoną gėlę. Vadinasi, B yra teisingas atsakymas.

Tai informatika!

Šioje užduotyje imituojama dėklo (angl. *stack*) duomenų struktūra, sudaryta pagal principą „paskutinis į, pirmas iš“ (angl. *LIFO, Last In First Out*). Linos matomus objektus galima laikyti į dėklą dedamais objektais. Pirmiausia į dėklą dedamas anksčiausiai pamatytas objektas (raudona ir geltona gėlės), po to – antras pamatytas objektas (obelis) ir tada – trečias pamatytas objektas (medis su paukščio lizdu). Tokį dėklą galima pavaizduoti taip:

Pirmas objektas	Antras objektas	Trečias objektas
		Medis su lizdu
	Obelis	Obelis
Raudona ir geltona gėlės	Raudona ir geltona gėlės	Raudona ir geltona gėlės

Kai Lina eina priešinga kryptimi, kiekvienas objektas iš dėklo turi būti išimamas. Pirmas dalykas, kurį Lina pamato eidama namo, yra medis su paukščio lizdu, tad jis ir išimamas iš dėklo, o dėklo viršuje lieka obelis. Tada iš dėklo išimama obelis. Galiausiai iš dėklo išimamos raudona ir geltona gėlės.

7. Pabirę lapeliai

2024-IE-01b

Simas pametė savo mėgstamiausią dobilo lapelį. Nukritęs lapelis susimaišė su kitais dobilo lapais! Štai Simo mėgstamiausio lapelio nuotrauka:



Kuris yra Simo mėgstamiausias lapelis?

Atsakymų variantai:

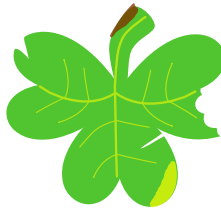
A



B



C



D



Teisingas atsakymas: D.

Paaiškinimas

Pageltęs lapelio kraštas ir parudavusi kotelio dalis yra tose pačiose vietose. Taip pat sutampa ir įplyšusi lapelio vieta bei nukąsta dalis.

Atsakymas A neteisingas, nes čia nėra pageltusios lapelio dalies.

Atsakymas B neteisingas, nes čia nėra parudavusios kotelio dalies.

Atsakymas C neteisingas, nes šis lapelis įplyšęs ir nukąstas kitose vietose.

Tai informatika!

Šiandieniniame pasaulyje informacijos dažnai ieškome pasitelkdami kompiuterines programas, o šios paiešką vykdo atpažindamos ir apdorodamos įvairius šablonus. Atpažinus šablonus informaciją galima geriau filtruoti, analizuoti ir suprasti. Pašalinus šablonų neatitinkančią informaciją, galima rasti tikslesnius rezultatus.

Šablonų galima rasti pačių įvairiausių tipų duomenyse – jie gali būti erdviniai, laikini, nepriekaištingi ar su klaidomis, netgi transformuoti (pvz., apversti).

Šablonų būna netgi duomenų aibių skirtumuose. Šiame uždavinyje šablonai yra paslėpti pasukus ir apvertus paveikslėlį – taip parodoma, kad duomenis gali reikėti apdoroti, kad rastume mums reikalingą šabloną.

8. Kepsnių vakarėlis

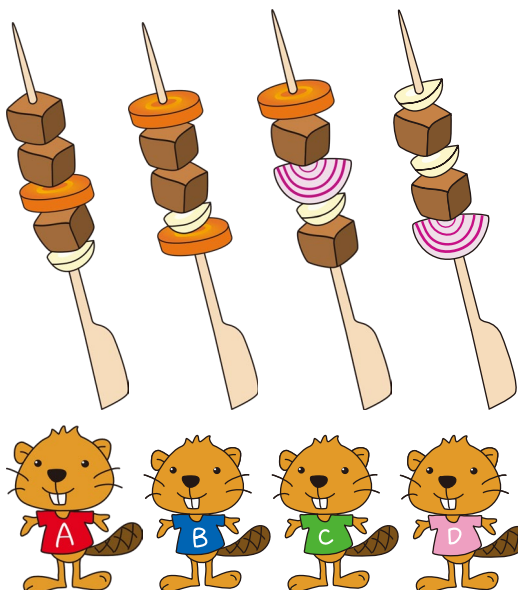
2024-KR-02

Kepsnių vakarėlyje keturi bebrai apibūdina norimus iešmelius:

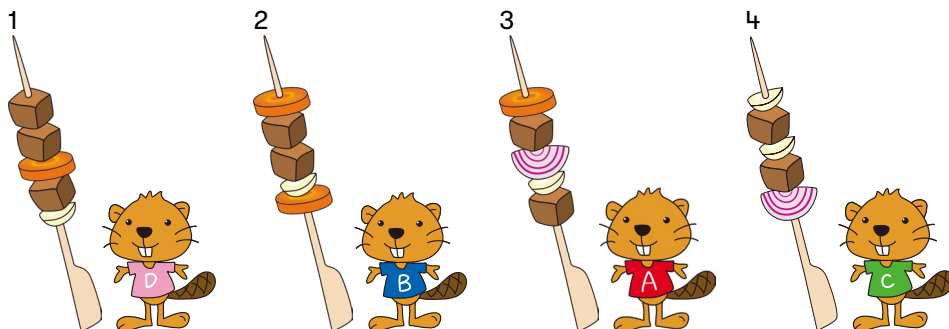


Yra keturi iešmeliai. Padalinkite iešmelius taip, kad visi bebrai būtų kuo labiau patenkinti.

Atsakymų variantai:



Teisingas atsakymas:



Paaiškinimas

Sunumeruokime iešmelius:

1	2	3	4

Tik 4-tas iešmelis turi daugiau nei vieną skiltelę česnako. Bebras C gauna 4-tą iešmelį.

Tik 1-as iešmelis turi daugiau nei du gabalėlius mėsos. Bebras D gauna 1-ą iešmelį. Iš dviejų likusių iešmelių tik 3-ias turi svogūno griežinėlių, todėl negali atitekti bebrui B, bet tinka bebrui A.

Bebrui B lieka 2-as iešmelis.

Tai informatika!

Apribojimų tenkinimo uždaviniams (angl. *Constraint satisfaction problem*, CSP) reikia rasti sprendimą, atitinkantį konkrečias taisykles, vadinamas apribojimais. Sprendžiant tokius uždavinius reikia išdėstyti ar parinkti elementus taip, kad būtų įvykdyti visi pateikti apribojimai. Atliekant šią užduotį iešmeliai turi būti paskirstyti taip, kad atitektų visus bebrų išsakytus apribojimus.

Apribojimų tenkinimo metodas taikomas įvairiose srityse: kuriant dirbtinio intelekto sistemas, sudarant tvarkaraščius, paskirstant išteklius, formuojant kompiuterinės įrangos konfigūraciją.

Nykštukai
12

Mažyliai
9

Bičiuliai
6

Draugai

Jauniai

Kolegos

9. Spalvinimo lapas

2024-FI-01

Beata piešia paveikslą. Ji turi penkias pilnas dažų tūbeles. Dideliems paveikslo plotams nuspalvinti reikia daugiau dažų nei mažiems. Kai Beata baigia piešti, tūbelėse lieka štai tiek dažų:

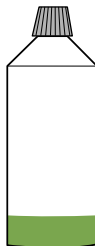
Dangus



Saulė



Žolė



Gėlės



Bebrai



Kurį paveikslą galėjo nupiešti Beata?

Atsakymų variantai:

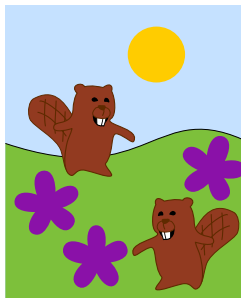
A



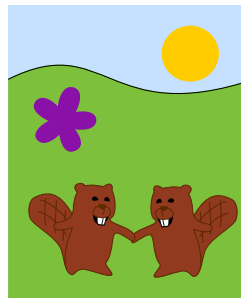
B



C



D



Teisingas atsakymas: D.

Paiškinimas

Norėdami rasti sprendimą, palyginkime nupiešto paveikslo spalvotus plotus su dažų tūbelėmis. Iš dažų likučių tūbelėse matome, kad daugiausia buvo naudojama žalios spalvos, antroje vietoje – mėlynos. Tai atitinka A, C ir D paveikslus – juose žalios spalvos plotas yra didžiausias, o mėlynos – antras pagal dydį.

B paveikslą galime atmesti iš karto, nes mėlynos spalvos plotas yra daug didesnis negu žalios.

Toliau nagrinėdami dažų tūbeles pastebėsime, kad geltonos ir violetinės spalvos buvo naudota po lygiai. Įvertinkime, kad A ir C paveiksluose bendras violetinis plotas (3 gėlės) yra maždaug 3 kartus didesnis nei geltonas plotas (saulė). Tik D paveiksle geltonas ir violetinis plotai (saulė ir gėlė) yra maždaug vienodo dydžio.

Tai informatika!

Nagrinėjome spalvas pateiktuose paveiksluose. Iškėlėme, pavyzdžiui, tokius klausimus: Kurios spalvos naudojamos? Kuri spalva naudojama dažniausiai? Kurių spalvų naudojama nedaug? Ar viena spalva naudojama dažniau nei kita? Kad atsakytumėme į šiuos klausimus, turėjome išskirti skirtingų spalvų paveikslų plotus, įvertinti jų dydžius, palyginti juos ir priimti sprendimus.

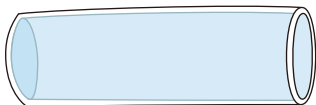
Spalvas paveiksluose galime išrikiuoti nuo mažiausiai nuspalvinto bendro ploto iki didžiausio. Taip pat tūbelėse galime išrikiuoti spalvas nuo mažiausiai panaudoto dažų kiekio iki daugiausiai. Šie du rikiavimai turi atitikti vienas kitą. Tokių analizę galima atlikti automatiškai naudojantis kompiuterinėmis programomis. Tokios programos gali padėti surasti nuotraukas, laikomas mobiliajame telefone. Naudodama spalvų analizę, kompiuterinė programa gali, pavyzdžiui, iš portretinių nuotraukų išskirti kraštovaizdžio nuotraukas: atpažįstama pagal viršuje vyraujančią mėlyną spalvą, o apačioje – žalią. Analizuodami palydovinių nuotraukų spalvas, kompiuteriai gali nustatyti, kaip tankiai užstatyta vietovė, ar gamtiniame rezervate buvo neteisėtai iškirsti miškai ir pan.

Informatikos sritis, kurioje automatiškai analizuojami vaizdai, vadinama kompiuterine grafika. Tai svarbi informatikos mokslo sritis.

10. Skaidrus vamzdelis

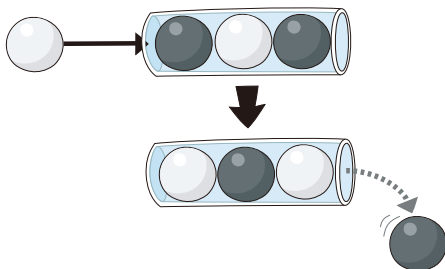
2024-KR-03a

Turime skaidrų, iš abiejų pusių atvirų vamzdelį, į kurį telpa 3 kamuoliukai.

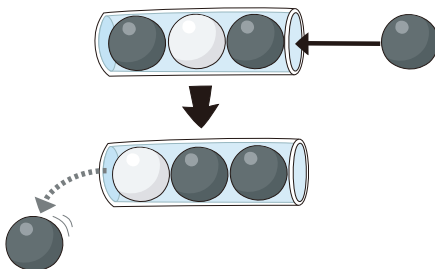


Kai į pilną vamzdelį bandoma įdėti dar vieną kamuoliuką, kitoje pusėje vienas kamuoliukas iškrenta:

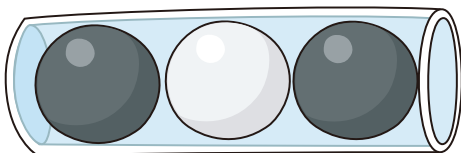
Dedamas kamuoliukas iš kairės ()



Dedamas kamuoliukas iš dešinės ()



Turime vamzdelį, užpildytą šitokiais kamuoliukais:



Į vamzdelį vienas po kito įdedami 4 nauji kamuoliukai:



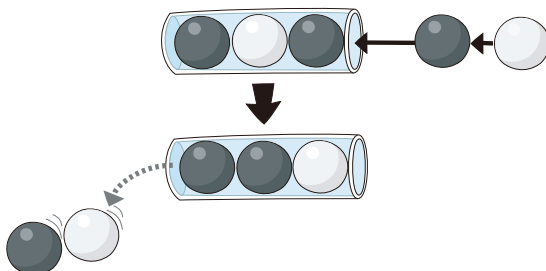
Kokie trys kamuoliukai liks vamzdelyje, į jį įdėjus šiuos kamuoliukus? Nutempkite kamuoliukus į tinkamas vietas.

Teisingas atsakymas:

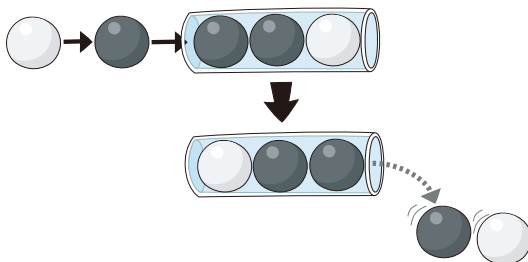


Paaiškinimas

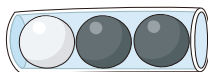
Į vamzdelį iš dešinės įstumiami du kamuoliukai, pirmiau juodas, o paskui baltas. Todėl kairėje iškrenta du, pirmiau juodas, paskui baltas:



Tada į vamzdelį iš kairės įstumiami du kamuoliukai, pirmiau juodas, o paskui baltas. Taigi šį kartą dešinėje iškrenta du rutuliai, pirma baltas, paskui juodas.



Taip gaunamas šitoks rezultatas:



Sprendimą galima rasti ir paprasčiau: kadangi du kamuoliukai iš pradžių į vamzdį įstumiami iš dešinės, o paskui du iš kairės, tai kamuoliukas, kuris iš pradžių buvo vamzdžio dešinėje, yra dešinėje, o į kairę nuo jo (iš dešinės į kairę) yra du paskutiniai į vamzdį įstumti kamuoliukai.

Tai informatika!

Šioje užduotyje vamzdelis veikia kaip dvipusė eilė (angl. *deque*, santrumpa iš *double-ended queue*) – duomenų struktūra, kurioje duomenys gali būti įdedami arba pašalinami iš abiejų pusių. Ši duomenų struktūra efektyviai išsaugo duomenis pagal prioritetą arba eiliškumą.

Užduotyje turime eilę (vamzdelį), į kurį galima tik įdėti duomenis. Pateiktas duomenų struktūros pavyzdys, kai negalime kaupti informacijos, nes turime ribotą talpą (atkreipkite dėmesį, kad vamzdelis gali būti užpildytas ne daugiau kaip 3 kamuoliukais). Vadinasi, jei reikės viršyti duomenų kiekį, prarasime informaciją: vamzdelis praras kamuoliuką, kai bandysime į jį įdėti ketvirtą kamuoliuką. Tai atspindi svarbų duomenų saugojimo informatikoje aspektą – talpyklos ribojimus.

11. Gimtadienio dovana

2024-NL-02

Fausta gimtadienio proga nori kamuolio.
Ji norėtų, kad kamuolys būtų toks:

- neturėtų juostelių;
- ant jo būtų žvaigždė;
- ant jo nebūtų mėnulio.

Kurį kamuolį nupirkti Faustai dovanų?

Atsakymų variantai:

A



B



C



D



E



F



Teisingas atsakymas: A.

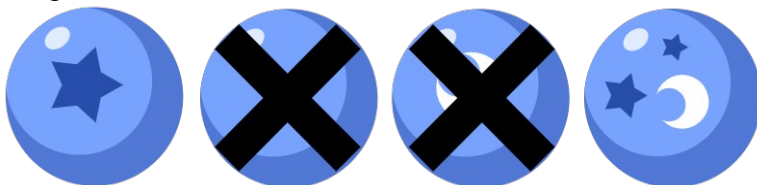
Paiškinimas

Galime panaudoti Faustos taisykles ir nustatyti, kurio kamuolio ji nori.

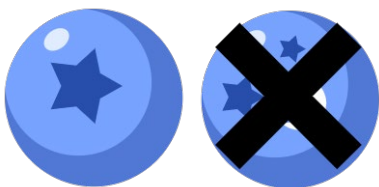
Ji nori kamuolio be juostelių, todėl mes pašaliname kamuolius, kurie turi juosteles:



Ji nori kamuolio su žvaigžde, todėl mes pašaliname visus kamuolius, kurie neturi žvaigždės:



Paskutinė taisyklė nurodo, kad neturi būti mėnulio, taigi mes pašaliname kamuolį su mėnuliu ir lieka vienintelis kamuolys su žvaigžde, kuris atitinka Faustos taisykles:



Tai informatika!

Šioje užduotyje naudojamas labai paprastas algoritmas, pagal kurį nustatoma, kuris kamuolys yra tinkamas. Algoritmas – tai taisyklių rinkinys, kurio kompiuteris turi laikytis sprendamas uždavinį.

Galime tai atvaizduoti naudodami pasirinkimų medį (srautų diagramą). Pagal kiekvieną sąlygą kamuolių grupę padalijame į dvi kategorijas: tuos, kurie atitinka sąlygą, ir tuos, kurie jos neatitinka. Tai kartojama tol, kol randame sprendimą.

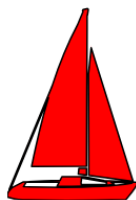
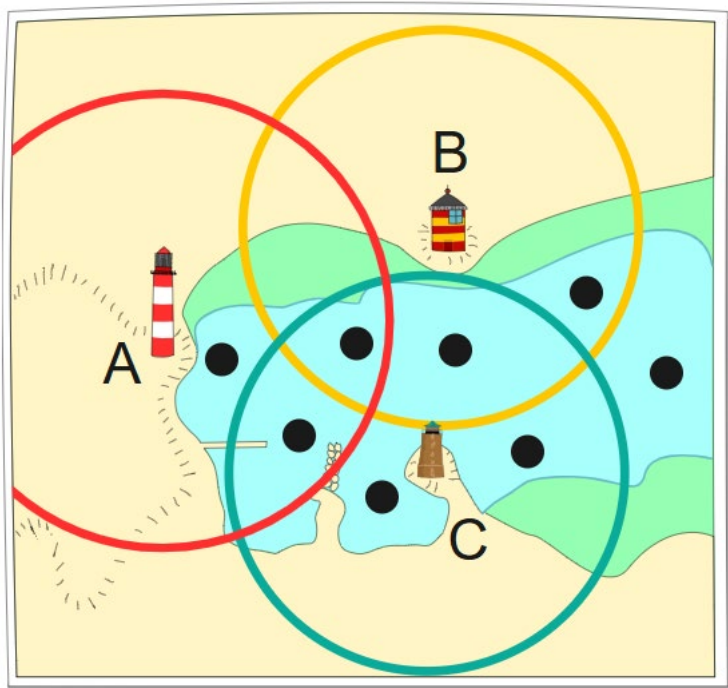
Kitas sprendimas – apžiūrėti kiekvieną kamuolį atskirai ir patikrinti, ar jis atitinka nurodytas sąlygas. Šis procesas vadinamas ciklu. Naudodami ciklus programavime galime efektyviai atlikti pasikartojančias užduotis. Mūsų užduotyje turime užtikrinti, kad visos Faustos sąlygos būtų įvykdytos, ir tai darome tikrindami ne daugiau kaip šešis kartus.

12. Švyturių žemėlapis

2024-CZ-05

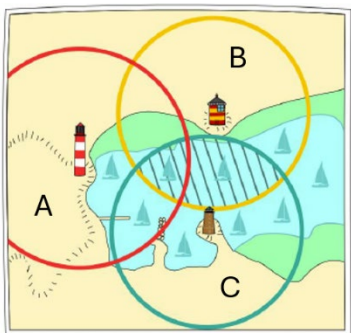
Kapitonas Benas yra laive. Jis turi uosto žemėlapij su trimis švyturiais, bet nežino savo laivo vietos. Aplink kiekvieną švyturį apibrėžtas skritulys. Kai laivas yra švyturio skritulyje, Benas mato tą švyturį.

Kapitonas Benas mato B ir C švyturius, tačiau nemato A švyturio. Nutempkite laivelį į taškus pažymėtą vietą žemėlapyje, kur yra Benas.

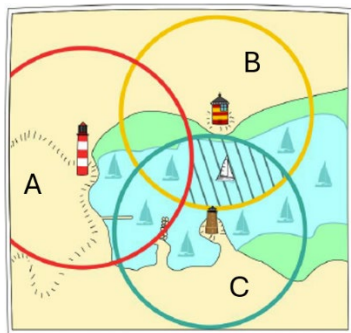


Paiškinimas

Benas mato du švyturius B ir C. Todėl jo laivas turi būti tame plote, kur susikerta šių švyturių skrituliai.



Tačiau Benas nemato A švyturio.
Taigi jo laivas nėra šio švyturio skritulyje.



Beno laivas yra čia, nes tai vienintelė vieta, kurioje tik B ir C skrituliai persidengia, o A skritulys jos nedengia.

Tai informatika!

Informatikoje dažnai tenka aprašyti ryšius tarp objektų ar objektų rinkinių. Paprastai naudojami ne žodiniai aprašymai, o simbolių ir taisyklių rinkinys, vadinamas Būlio algebra. Pateiktą uždavinį galima būtų užrašyti trumpai: B ir C, ir ne A.

Informatikoje tikslus problemos aprašymas ir sprendimo procesas yra labai svarbus. Naudinga priemonė tam yra teiginių logika: Benas vadovaujasi trimis teiginiais, kurių kiekvienas gali būti teisingas arba klaidingas:

Teiginys A: „Švyturys A yra matomas.“

Teiginys B: „Švyturys B yra matomas.“

Teiginys C: „Švyturys C yra matomas.“

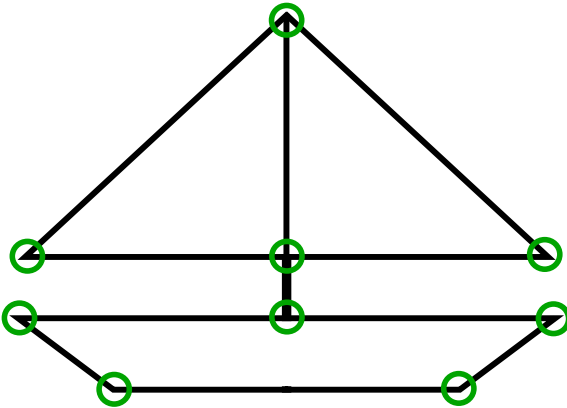
Beno atveju teiginiai B ir C yra teisingi, o A yra klaidingas (Benas mato švyturius B ir C, bet nemato švyturio A). Teiginių logikoje tai užrašoma šitaip: „B IR C, BET NE A“. Žemėlapyje, pavyzdžiui, „A IR B IR C“ atitinka sritį, kurioje visi trys skrituliai persidengia, nes tik ten galima pamatyti visus tris švyturius. „NE A IR NE B IR NE C“ atitinka sritį, kuri yra už šių trijų skritulių ribų. Nė vienas iš trijų švyturių čia nematomas.

Apskritai teiginių logika informatikoje naudojama žinioms atvaizduoti, programoms tikrinti ir skaitmeninėms grandinėms modeliuoti. Informatikoje dažnai tenka aprašyti ryšius tarp objektų arba objektų rinkinių. Mes nenaudojame žodinių aprašymų, bet naudojame simbolių ir taisyklių rinkinį, vadinamą logikos algebra. Mūsų uždavinį galima užrašyti taip: B ir C, ir ne A. Šiame uždavinyje naudojama diagrama vadinama Veno diagrama. Veno diagramoje vaizduojami persidengiantys skrituliai ar kitos figūros, kuriomis iliustruojami loginiai ryšiai tarp dviejų ar daugiau elementų rinkinių. Dažnai Veno diagrama padeda grafiškai suskirstyti daiktus, pabrėžiant, kaip elementai yra susiję.

13. Burvaltės piešimas

2024-PL-03

Sofija nori perpiešti pavaizduotą burinę valtį.



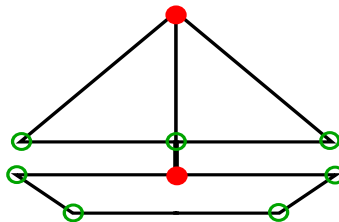
Ji nusprendžia laikytis šių taisyklių:

- Negalima pakelti pieštuko nuo popieriaus lapo.
- Negalima piešti tos pačios linijos du kartus.

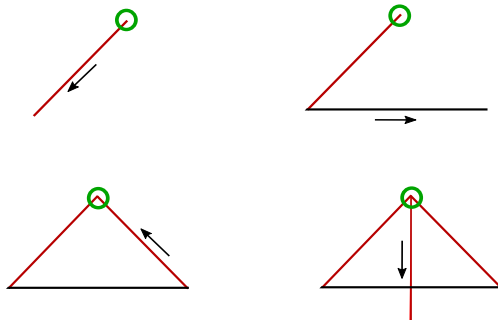
Parinkite tašką, nuo kurio Sofija turi pradėti piešti.

Paaškinimas:

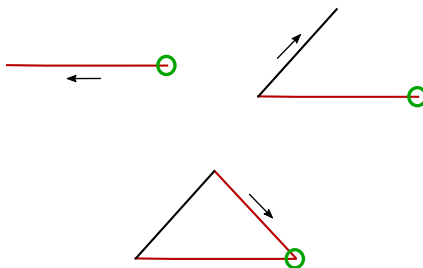
Norint nupiešti burinę valtį pagal taisyklės, galima pradėti nuo dviejų taškų, parodytų paveiksle.



Iš abiejų šių taškų išeina nelyginis linijų skaičius. Kai pradedame nuo taško, iš kurio išeina nelyginis linijų skaičius, nupiešiame pirmąją liniją ir grįžtame į tą patį tašką, nubrėždami antrą ir trečią linijas. Tuomet pakeičiame kryptį ir piešiame dar nenupieštą liniją, kaip parodyta paveikslėlyje.



Kiekvienas taškas su išeinančiu lyginiu linijų skaičiumi yra neteisingas, nes į tą tašką sugrįžę neturėsime išėjimo – nepanaudotos linijos piešinio pratęsimui.



Tai informatika!

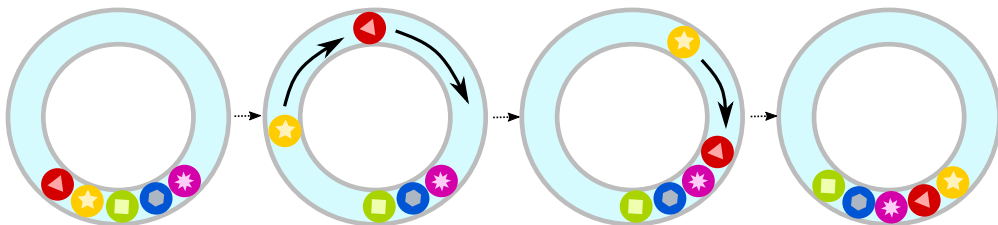
Ši užduotis susijusi su Oilerio ciklu. Oilerio ciklas baigtiniame grafe – ciklas, kuriuo galima apeiti visas grafo briaunas lygiai po vieną kartą ir sugrįžti į tą pačią viršūnę. Oilerio ciklą galima rasti įvairiais algoritmais. Vienas iš populiariausių yra Flerio algoritmas Oilerio ciklui rasti. Algoritmas taikomas, jei grafas turi ne daugiau kaip dvi viršūnes, iš kurių išeina po nelyginį skaičių briaunų. Algoritmas pradedamas nuo vienos iš šių viršūnių.

Oilerio ciklas gali būti taikomas ir realiame gyvenime. Pavyzdžiui, kai lankote miestelio žymias vietas, Oilerio ciklas gali padėti suplanuoti maršrutą, kad kiekvienu keliu eitumėte tiksliai vieną kartą, negrįždami ir aplankytumėte visas norimas miestelio vietas.

14. Oliverio barškutis

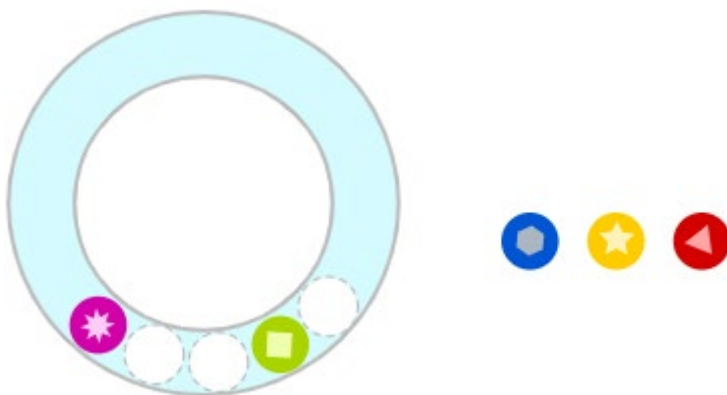
2024-SK-01b

Oliveris turi permatomą barškutį su spalvotais rutuliukais viduje. Kai jis papurto barškutį, dalis rutuliukų apskrieja ratu ir atsiduria kitoje barškučio pusėje, kaip matyti paveikslėlyje.



Oliveris papurto barškutį dar kartą.

Nuvalkite rutuliukus į baltas vietas taip, kaip jie dabar išsidėstys barškutyje.



Teisingas atsakymas:



Paiškinimas

Purtant barškutį rutuliukai nekeičia išsidėstymo eilės tvarkos. Rutuliukai Oliverio

barškutyje visada bus išdėstyti ta pačia tvarka. Po raudono rutuliuko  eina

geltonas  . Po geltono rutuliuko eina žalias  . Po žalia rutuliuko eina mėlynas

 . Po mėlyno rutuliuko eina violetinis  . O po violetinio rutuliuko eina vėl

raudonas  .

Po žalia rutuliuko trūksta vieno rutuliuko, kuris būtų tarp žaliao ir violetinio rutuliukų. Ankstesnėje pastraipoje aptarėme, kad tai turėtų būti mėlynas rutuliukas. Matome, kad tarp violetinio ir žaliao rutuliuko trūksta dviejų rutuliukų. Tai turėtų būti geltonas ir raudonas rutuliukai. Samprotaudami pasiremiame pradine seka ir nustatome, kad prieš žalią rutuliuką eina geltonas. O tarp violetinio ir geltono rutuliukų yra raudonas rutuliukas.

Tai informatika!

Kompiuterių specialistai duomenis pateikia ir kaupia įvairiais būdais. Skirtingi duomenų atvaizdavimai turi įvairių privalumų, kurie priklauso nuo tam tikrų duomenų savybių.

Šioje užduotyje naudojamas vadinamasis žiedinis susietasis sąrašas, kuris čia atitinka žaisliuką braškutį, o jame laikomi elementai yra rutuliukai. Šiame ratu susietame sąraše kiekvienas elementas turi vadinamąją rodyklę į kitą elementą. Paskutinio elemento rodyklė nukreipia į pirmąjį elementą ir taip uždaro ratą. Tai reiškia, kad šioje užduotyje kiekvienam rutuliukui žinome, kuris rutuliukas eina po jo. Po paskutinio rutuliuko eina pirmas rutuliukas. Tikrindami rutuliukų eiliškumą galime pradėti nuo bet kurio rutuliuko ir turime pereiti visus rutuliukus (visą sąrašą) teisinga tvarka.

Yra du privalumai, kai elementams laikyti naudojame žiedinį susietąjį sąrašą. Pirmasis privalumas yra tas, kad žiediniame susietajame sąraše nėra galinio elemento. Kadangi sąrašas cikliška grįžta į pradžią, elementams niekada nepritrūksta duomenų, kuriuos galima patikrinti ar panaudoti. Antrasis privalumas – naršymo sąraše efektyvumas. Sąraše lengva judėti nuo vieno elemento prie kito netaikant specialių taisyklių pasiekus galinį elementą. Judama tiesiog viena kryptimi.

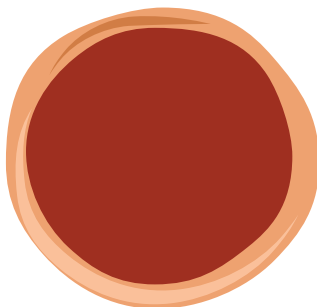
15. Picų vakarėlis

2024-DE-02

Jonas rengia picų vakarėlį. Jis žino, kokius picos priedus mėgsta jo draugai.

Alisa	 (paprikos)	 (grybai)	 (sūris)
Benas	 (grybai)	 (ananasas)	 (svogūnai)
Cezaris	 (grybai)	 (gražgarstės)	 (sūris)
Daina	 (grybai)	 (sūris)	 (svogūnai)

Jonas gamins vieną picą su trimis priedais. Jis nori parinkti tris priedus taip, kad patenkintų kuo daugiau draugų norų. Kokie tai bus priedai?



Paiškinimas

Visi Jono draugai nori grybų (4 norai), 3 draugai nori sūrio, o 2 draugai – svogūnų. Iš viso tenkinami 9 jų norai. Nė vienu kitokiu priedų deriniu negalima tenkinti daugiau norų. Kaip rasti sprendimą? Galite tiesiog išbandyti visus derinius ir suskaičiuoti, kiek draugų norų tenkinama. Tačiau galima taikyti sisteminį metodą.

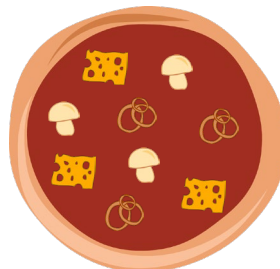
1 žingsnis: Apskaičiuojame kiekvieno priedo dažnumą

Draugų norus surašome į lentelę. Picos priedams skiriami stulpeliai, o draugų norams – eilutės. Jei draugas nori kurio nors priedo, tai šio priedo stulpelio ir draugo eilutės sankirtoje įrašome 1. Pavyzdžiui, Alisa nori paprikos, todėl pirmoje eilutėje ir pirmame stulpelyje įrašome 1.

						
Alisa	1	1			1	
Benas		1	1			1
Cezaris		1		1	1	
Daina		1			1	1
Vaikų, mėgstančių šį priedą, skaičius	1	4	1	1	3	2

2 žingsnis: Nustatome tris populiariausius priedus

Susumuojame kiekvieno stulpelio skaičius ir apatinėje eilutėje surašome sumas. Populiariausi yra grybai (4), po jų eina sūris (3) ir svogūnai (2). Kitų priedų prašoma tik po vieną kartą. Trys populiariausi priedai tenkina daugiausiai draugų norų, t. y., iš viso 9 norus.



Tai informatika!

Sprendžiant uždavinį naudojamos algoritminės strategijos, pavyzdžiui, skaičių masyvas arba dažnių masyvas. Nustatant kiekvieno priedo dažnį ir pasirenkant tris populiariausius, kad patenkintų vaikų skaičius būtų didžiausias, reikia algoritminio mąstymo, kuris yra vienas iš pagrindinių informatikos įgūdžių. Šis uždavinys yra paprastas optimizavimo pavyzdys. Norint išspręsti uždavinį reikia surinkti duomenis (draugų norus), apskaičiuoti naujus duomenis (norų dažnius), atsižvelgti į ribojimus (gaminama tik viena pica su trimis priedais) ir priimti sprendimus (pasirinkti priedus). Aiškinant šio uždavinio sprendimą taikyta dviejų žingsnių procedūra – iš tiesų tai yra algoritmas, galintis išspręsti šį optimizavimo uždavinį bendru atveju. Jis tinka ir esant kitokiems priedams ir kitiems norams.

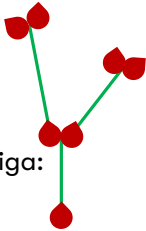
Algoritmus kompiuteris gali vykdyti automatiškai. Optimizavimas svarbus daugelyje gyvenimo sričių, pavyzdžiui, transporte (optimalus transporto priemonės maršrutas) arba pramonėje (optimalūs gamybos procesai).

16. Stebuklingoji gėlė

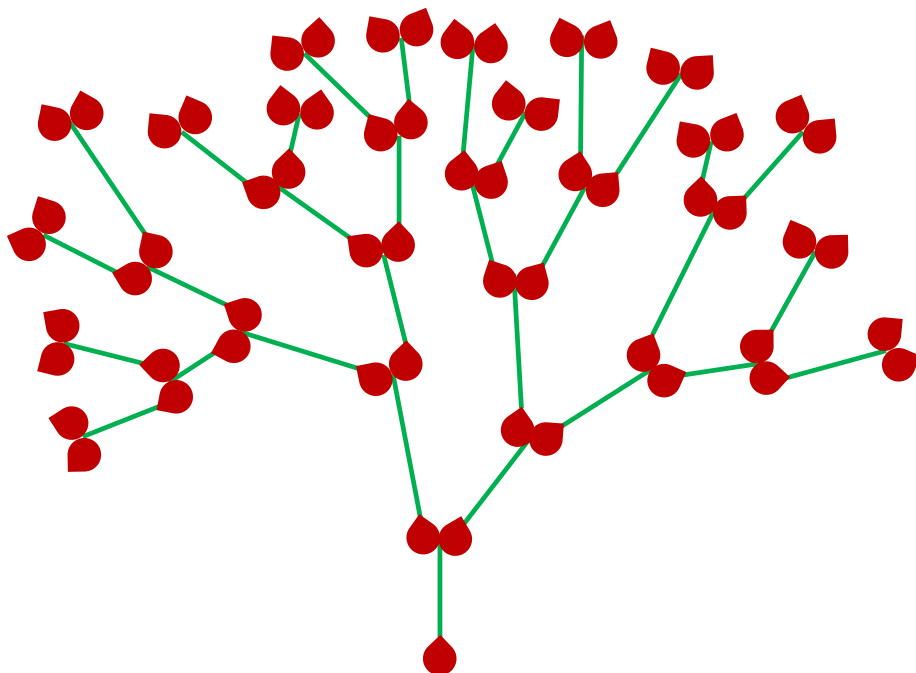
2024-DE-03

Dienos pradžioje iš kiekvieno stebuklingosios gėlės pumpuro išauga stiebas su dviem naujais pumpurais.

Dienos pabaigoje gėlė nustoja augti. Taip kartojasi diena po dienos ir stebuklingoji gėlė tampa vis didingesnė.

Pirmos dienos pradžia:  Pirmos dienos pabaiga:  Antros dienos pabaiga: 

Kiek dienų auga ši stebuklingoji gėlė?



Teisingas atsakymas: 5.

Paiškinimas

Kiekvieną dieną iš kiekvieno stebuklingosios gėlės pumpuro išauga stiebas su dviem naujais pumpurais. Norint suskaičiuoti augimo dienas reikia sekti tik vieną šaką, pavyzdžiui, pačią dešiniausią.

Tai informatika!

Stebuklingosios gėlės auga pagal tam tikrą taisyklę: kiekvieną dieną iš naujo pumpuro išauga lygiai vienas stiebas ir du nauji pumpurai. Kitą dieną ta pati taisyklė vėl taikoma visiems naujai atsiradusiems pumpurams. Teoriškai stebuklingoji gėlė gali augti neribotą laiką.

Tai nenutrūkstamas procesas – ta pati taisyklė taikoma pakartotinai. Toks pakartotinis taisyklės taikymas dažnai pasitaiko kompiuterių valdomose sistemose. Informatikoje procesas, kai kelis kartus kartojami tie patys arba panašūs veiksmai, vadinamas iteracija.

Pavyzdžiui, šviesoforų valdymo sistemos nuolat generuoja tą pačią šviesoforo signalų seką. Lošimo automate tą patį žaidimą galima paleisti vėl ir vėl.

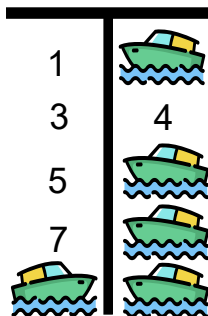
Kitas iteracijos (netgi rekursijos) pavyzdys yra Prancūzijos „Bebro“ konkurso logotipas. Ant šio logotipo nupiešto bebro marškinėlių matomas vėl to paties bebro atvaizdas, jis dėvi marškinėlius su bebro atvaizdu ir t. t.

17. Savaitgalis prieplaukoje

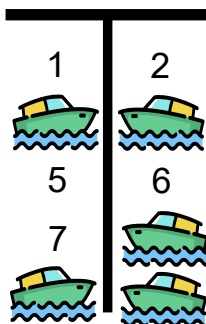
2024-MT-02

Bebras Tomas prieplaukoje padeda laiveliams rasti laisvą vietą ir prisišvartuoti. Tomas turi rasti tokias laisvas vietas, kuriose laiveliai galėtų praleisti dvi dienas iš eilės nuo penktadienio iki sekmadienio. Paveiksle pavaizduota, kurios vietos kokiomis dienomis yra užimtos.

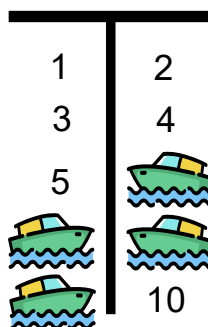
Penktadienis



Šeštadienis



Sekmdienis



Kurios švartavimosi vietos bus laisvos bent dvi dienas iš eilės?

Teisingas atsakymas: 1-a, 2-a, 5-a ir 7-a vietos.

Paiškinimas

Galimos 4 laisvos vietos. Štai visos tinkamos vietos:

1. 1-a vieta penktadienį ir šeštadienį; 1-a vieta šeštadienį ir sekmadienį
2. 2-a vieta šeštadienį ir sekmadienį
3. 5-a vieta penktadienį ir šeštadienį; 5-a vieta šeštadienį ir sekmadienį
4. 7-a vieta penktadienį ir šeštadienį

Tai informatika!

Šiai užduočiai išspręsti reikia sudaryti loginę grandinę dvejetainiams duomenims apdoroti. Dvejetainiais duomenimis čia vaizduojama švartavimosi vietų prieplaukoje būseną. Sprendimas reikalauja suprasti loginius elementus (IR, ARBA, NE) ir kaip juos galima sujungti dvejetainėms sekoms analizuoti. Informatikoje panašūs loginiai elementai ir jų grandinės naudojamos kompiuterių architektūroje, skaitmeniniams signalams apdoroti ir kitose srityse, kuriose reikia apdoroti ir analizuoti dvejetainius duomenis. Spręsdami šią užduotį mokiniai pagilins informatikoje būtinus loginio mąstymo ir uždavinių sprendimo įgūdžius.

18. Klasių skambutis

2024-MY-03

leva perėjo į kitą mokyklą, bet vis dar susiskambina su savo draugais iš ankstesnės mokyklos.

Vaizdo skambučio metu levos draugai sėdi vienas šalia kito, tačiau prie atskirų kompiuterių.

Štai kaip leva mato draugus savo kompiuterio ekrane:



Sudėliokite levos draugus tokia tvarka, kokia jie sėdi prie kompiuterių.

Paiškinimas

levos kompiuterio ekrane galima matyti, kas sėdi šalia kiekvieno mokinio.

Iš apatinio dešiniojo langelio sužinome, kas yra kraštinis (kairiausias) eilėje ir kas sėdi šalia jo.

Tokiu būdu peržiūrėję visus ekrano langelius galime išsiaiškinti visų devynių mokinių sėdėjimo tvarką:



Tai informatika!

Uždavinio sąlygoje vaizduojama vaizdo skambučių programa suskirsto ekraną į langelius, kuriuose matomas ne tik vieno mokinio atvaizdas, bet ir dalis šalia sėdinčių mokinių atvaizdų. Tai tarsi nuorodos į šalia sėdinčius mokinius.

Informatikoje panašios duomenų struktūros naudojamos labai dažnai.

Duomenų struktūra, panaši į mokinių rodymo ekrane būdą, vadinama dvikrypčiu sąrašu. Kiekvienas elementas dvikrypčiame sąraše turi nuorodą į ankstesnį elementą ir nuorodą į tolesnį elementą. Uždavinyje sąrašo elementai yra ekrano langeliuose rodomi mokinių atvaizdai, o daliniai iš kairės ir iš dešinės sėdinčių mokinių atvaizdai – tai nuorodos į kitus sąrašo elementus. Užduočiai išspręsti reikia pereiti visus sąrašo elementus pradedant nuo bet kurio galo.

Nuorodų dvikryptiškumas tokioje duomenų struktūroje leidžia pereiti ne tik prie kito, bet ir prie ankstesnio sąrašo elemento. Tokiu būdu norimą sąrašo elementą galima rasti greičiau. Žinoma, kiekvienam elementui įrašyti po dvi nuorodas reikia ir daugiau kompiuterio atminties.

19. Kelias robotui

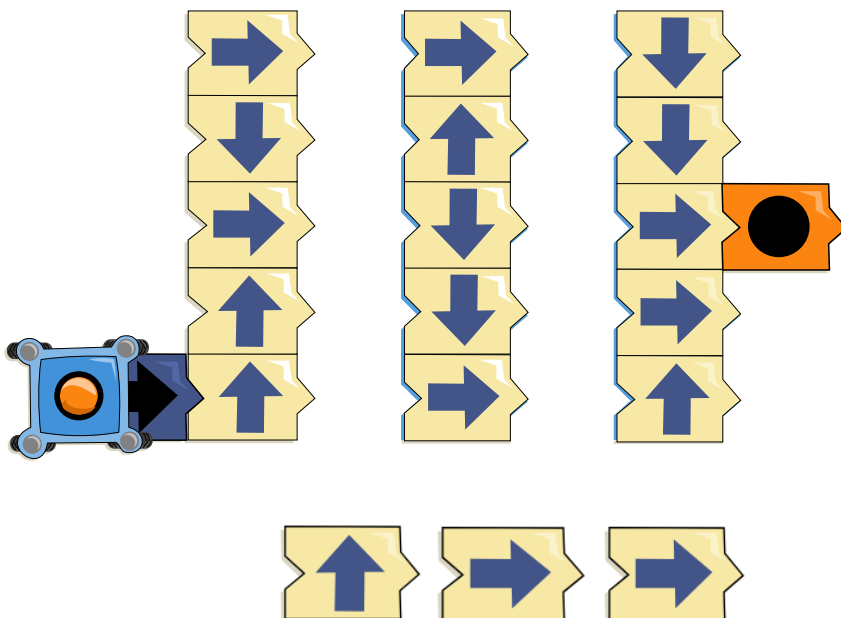
2024-PL-04

Tomo robotas juda plytelėmis pagal tam tikras taisykles:

- Robotas pradeda nuo plytelės, esančios kairėje.
- Robotas juda rodyklės, ant kurios stovi, kryptimi.
- Jei robotas nuvažiuoja nuo plytelės, jis sustoja.

Plytelės iš anksto išdėstytos taip, kaip toliau parodyta. Tomui yra duotos 3 papildomos plytelės, kurias turės panaudoti kelyje.

Padėkite Tomui sudėti visas 3 papildomas plyteles į tuščias vietas taip, kad jo robotas pasiektų dešinėje esančią plytelę su skrituliu.



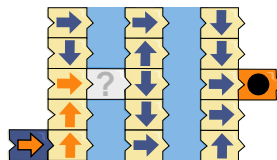
Paiškinimas

Šios užduoties tikslas – naudojantis trimis plytelėmis pereiti nuo vieno stulpelio prie kito ir pasiekti plytelę su skrituliu.

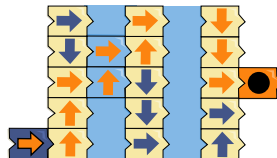
Iš pirmojo stulpelio į antrąjį galime pereiti tik per pirmojo stulpelio trečiąją plytelę, kaip parodyta paveikslėlyje.

Pirmoji naudojama plytelė turi būti padėta vietoje, pažymėtoje klausuku.

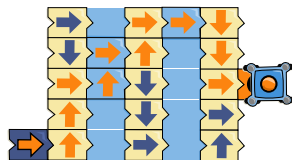
Užėjus ant šios plytelės yra dvi galimybės: eiti aukštyn arba į dešinę.



Jei eisime aukštyn, turime naudoti dar vieną plytelę, kad pereitume į antrą stulpelį.

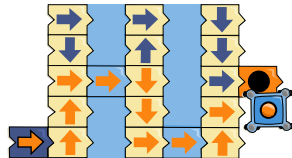


Sekdami antrojo stulpelio rodyklėmis pasiekiamo pirmąją (viršutinę) antrojo stulpelio plytelę. Dabar paskutinė papildoma plytele galime pereiti į trečiąjį stulpelį ir galiausiai pasiekti plytelę su skrituliu.

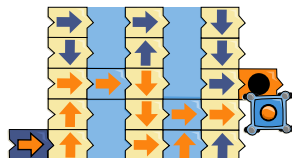


Jei eisime į dešinę, pasieksime apatinę antrojo stulpelio plytelę. Vėl turime dvi galimybes.

a) Pasinaudojame papildoma plytele, kad pakiltume į viršų. Tuomet naudojame trečiąją plytelę ir, deja, robotas nepasieks žalios plytelės su skrituliu ir išvažiuos lauk, kaip parodyta paveikslėlyje.



b) Naudojame papildomą plytelę, kad eitume į dešinę. Šiuo atveju robotas irgi nepasieks plytelės su skrituliu ir išvažiuos, kaip parodyta paveikslėlyje. Taip pat nepanaudojome trečiosios plytelės, o tai prieštarauja taisyklėms.



Tai informatika!

Šioje užduotyje pateikiamos plytelės roboto keliui nutiesti ir reikia surasti tą kelią, kuris vestų iki tikslo. Yra keletas dalykų, kuriuos žinote – galite naudoti tik tris plyteles (dvi plyteles su rodykle į dešinę ir vieną plytelę su rodykle į viršų) ir jas galite dėti tik raudoname plote, todėl kiekviename stulpelyje reikia rasti plytelę su rodykle į dešinę, kad galėtumėte judėti. Informatikoje tai vadinama apribojimais. Be apribojimų ši užduotis galėtų turėti daugiau sprendinių ir ją būtų sunkiau išspręsti, nes reikėtų galvoti apie daugybę galimų baigčių. Pavyzdžiui, jei ieškote maršruto tarp dviejų vietų, jų gali būti keletas, tačiau pridėję kelis apribojimus galėtumėte rasti geriausią. Galbūt reikia sustoti prie parduotuvės arba dalį maršruto norite įveikti autobusu. Visi šie apribojimai sumažina galimų maršrutų skaičių ir padeda išsirinkti tinkamiausią.

20. Kortelės su baubais

2024-CH-03a

Barbora renka korteles su baubų paveikslukais. Štai jos baubų kortelės:



Kortelėje nurodomos baubo savybės: vardas () , ar baubas turi dantų () ir kita. Savybės turi reikšmes. Pavyzdžiui, 2-os kortelės  reikšmė yra MONI, o  reikšmė ✓, kas reiškia, kad šios kortelės baubo vardas MONI ir jis turi dantis. Barbara pastebi, kad reikšmės kortelėse yra tik trijų skirtingų rūšių. Jos vadinamos „tipais“:

- ☐ **ABC...** Tekstas: kelios raidės iš eilės
- ☐ **123...** Skaičius: 1, 2, 3 ir taip toliau
- ☐ **✓/×** ženklai ✓ ir X, kurie reiškia „taip“ ir „ne“

Priskirkite savybėms tinkamus tipus.



Teisingas atsakymas:

**ABC...****123...****123...****✓/×****✓/×**

Paaiškinimas:

Šioje lentelėje pateikiamos visų šešių kortelių penkių savybių reikšmės:

1	JOSI	3	2	X	✓
2	MONI	3	2	X	✓
3	KILI	2	2	✓	✓
4	BENI	2	2	X	✓
5	LORI	2	5	X	✓
6	PHIL	2	2	X	✓

Iš lentelės matome, kad:

visos savybės reikšmės yra tekstas;

visos ir savybių reikšmės yra skaičius;

visos ir savybių reikšmės yra taip/ne tipo.

Tai informatika!

Barbora pastebėjo, kad jos baubų kortelėse naudojamos skirtingos savybių reikšmės. Šių tipų atskyrimas gali padėti organizuoti korteles. Kadangi Barbora žino, kad visos kojų savybės reikšmės yra skaičiai, ji gali palyginti baubų kojas ir surikiuoti korteles pagal kojų skaičių. Baubų kortelė yra tarsi kompiuteryje laikomas duomenų rinkinys. Kompiuteriuose informacija koduojama 1 ir 0. Kadangi jie gali reikšti bet ką, duomenų tipai reikalingi tam, kad nurodytume kompiuteriui, kaip tuos 1 ir 0 interpretuoti ir apdoroti. Todėl kompiuterių programose taip pat naudojami „reikšmių tipai“, kurie informatikoje vadinami duomenų tipais, siekiant nustatyti, kaip kompiuteris turėtų interpretuoti ir tvarkyti šiuos 1 ir 0. Programose duomenų tipai palengvina darbą su duomenimis ir padeda išvengti klaidų. Šio „Bebro“ uždavinio reikšmių tipai atitinka svarbius programavimo duomenų tipus:

ABC... programavimo kalbose vadinamas „eilute“: duomenys yra ženklų (raidžių, skaitmenų ir kitų simbolių) sekos.

123... atitinka sveikąjį tipą arba „integer“: duomenys yra sveikieji skaičiai, t. y., ... -2, -1, 0, 1, 2,

✓/× atitinka loginį tipą arba „boolean“: duomenys yra dvi reikšmės, kurios vadinamos „true“ ir „false“ (taip ir ne).

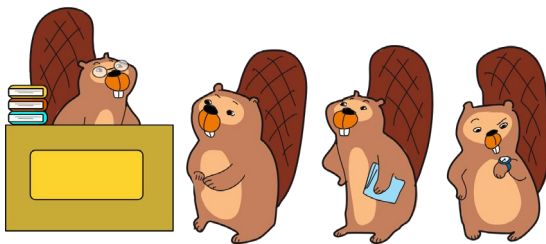
21. Knygų grąžinimas

2024-SI-01

Kietmedžio kaimo bebrai labai mėgsta skaityti. Todėl bibliotekoje nuolatos nusidriekia ilga eilė bebrų, norinčių grąžinti knygas. Bibliotekininė nusprendė taikyti tokią taisyklę:

Bebras, turintis mažiausiai knygų, aptarnaujamas pirmiausiai

Priimti vieną knygą bibliotekininkei užtrunka lygiai vieną minutę. Kai iš bebro priimamos visos knygos, grąžinti knygas pakviečiamas kitas eilėje stovintis bebras, kuris tuo metu turi mažiausiai knygų.



Ana atėjo, kai tik biblioteka atsidarė, todėl iš karto galėjo pradėti grąžinti savo 4 knygas.

Vardas	Atėjimo laikas	Knygų skaičius
Ana	9.00	4
Betė	9.02	6
Cilė	9.03	2
Dana	9.05	4
Emilis	9.11	1

Kuria tvarka bebrai grąžins savo knygas?

Teisingas atsakymas: Ana, Cilė, Dana, Betė, Emilis.

Paiškinimas

Ana į biblioteką atėjo pirmoji. Kadangi ji yra vienintelė eilėje, todėl gali iš karto grąžinti knygas. Savo 4 knygas ji baigia grąžinti 9.04.

9.04 eilėje yra 2 bebrai: Betė su 6 knygomis, atėjusi 09.02, ir Cilė su 2 knygomis, atėjusi 09.03. Cilė turi mažiau, todėl ji pradeda grąžinti knygas ir baigia 09.06.

09.06 eilėje yra 2 bebrai: Betė su 6 knygomis ir Dana su 4 knygomis, atėjusi 09.05.

Dana turi mažiau knygų negu Betė, todėl ji pradeda grąžinti ir baigia 09.10.

09.10 eilėje yra tik Betė, todėl ji grąžina savo 6 knygas iki 09.16.

09.16 eilėje yra tik Emilis su 1 knyga, atėjęs 09.11. Jis grąžina savo knygą iki 09.17.

Tai informatika!

Uždavinys iliustruoja, kaip veikia planuoklė. Planuoklė – tai programa, kuri yra operacinės sistemos dalis. Kai kompiuteris veikia, reikia atlikti daugybę užduočių. Kiekvienai užduočiai sukuriamas procesas, sprendžiantis užduotį. Planuoklė valdo tokių procesų eigą. Ši programa nustato, kada ir kiek laiko kompiuterio centrinis procesorius (CP) vykdo konkretų procesą. Yra įvairių skirtingų planavimo koncepcijų, pateikiame kelis pavyzdžius.

First Come First Serve (FCFS) – procesas, esantis pirmas eilėje, yra vykdomas pirmiausiai ir įvykdomas iki galo. Pagal šį principą veikia dauguma mūsų kasdienio gyvenimo eilių (pirmas atėjęs – pirmas aptarnautas).

Priority Scheduling – procesas, turintis aukščiausią prioritetą, yra vykdomas pirmiausiai ir įvykdomas iki galo.

Round Robin – procesai vykdomi paeiliui, skiriant kiekvienam tam tikrą laiką. Jei procesas nėra įvykdytas iki galo per tą laiką, jis vėl pastatomas į eilę, o vykdyti parenkamas kitas procesas.

Shortest Job First (SJF) – procesas, kurio numatomas vykdymo laikas yra trumpiausias, vykdomas pirmiausiai ir įvykdomas iki galo. Šis principas panaudotas mūsų užduotyje.

22. Saulėtos dienos

2024-DE-04a

Tomas sako: „Saulėtomis dienomis kiekviename tvenkinyje maudosi bent vienas bebras“. Kazys atsako: „Tai netiesa. Pavyzdžiui, praėjusį sekmadienį taip nebuvo.“



Sakykite, kad Kazys yra teišus. Kas vyko praėjusį sekmadienį? Pabaikite teiginį:
Praėjusį sekmadienį buvo saulėta ir _____.

Atsakymų variantai:

Parinkite vieną iš šių variantų:

- A. visuose tvenkiniuose maudėsi bebrai
- B. tvenkinyje su kriekliu bebrai nesimaudė
- C. bebras Mykolas maudėsi visuose tvenkiniuose
- D. bebras Mykolas visai nesimaudė

Teisingas atsakymas: B.

Paaiškinimas

Pagal šį atsakymą bebrai maudėsi ne visuose tvenkiniuose (tvenkinyje su kriokliu nesimaudė). Taip pat žinome, kad praėjęs sekmadienis buvo saulėtas. Taigi, praėjęs sekmadienis buvo saulėta diena, kai bent viename tvenkinyje bebrai nesimaudė. Todėl Tomas klysta sakydamas, kad kiekvieną saulėtą dieną kiekviename tvenkinyje maudosi bebrai.

Kiti atsakymų variantai neįrodo, jog Tomas yra neteisus:

- a) visuose tvenkiniuose maudėsi bebrai – tai reiškia, jog kiekviename tvenkinyje maudėsi bebrai, ir atitinka Tomo teiginį;
- b) bebras Mykolas maudėsi visuose tvenkiniuose – tai reiškia, jog kiekviename tvenkinyje maudėsi bent vienas bebras – Mykolas – ir atitinka Tomo teiginį;
- c) bebras Mykolas visai nesimaudė – visuose tvenkiniuose galėjo maudyti kiti bebrai, ir tai atitiktų Tomo teiginį.

Tai informatika!

Tomas ir Kazys aiškinasi, ar Tomo teiginys yra tiesa, ar netiesa. Mokslas, nagrinėjantis teiginius, kurie gali būti arba tiesa, arba netiesa, vadinamas logika. Naudojant operatorius, pavyzdžiui, IR, ARBA, NE, ir kiekinius žodžius, pavyzdžiui, KIEKVIENAS, BENT VIENAS, galima sudaryti itin sudėtingus teiginius. Bendravime remiantis logika galima išvengti nesusipratimų ir neteisingų išvadų. Be to, logika yra kompiuterių darbo principų pagrindas. Visų kompiuterinių programų veikimas paremtas loginių operatorių, tokių kaip NE ar IR, interpretavimu. Taip pat ir programavimo kalbose, kuriomis rašomos komandos kompiuteriui, naudojami loginiai teiginiai, pavyzdžiui, „JEI dabar vasara IR šviečia saulė, TAI įjunk kondicionierius“.

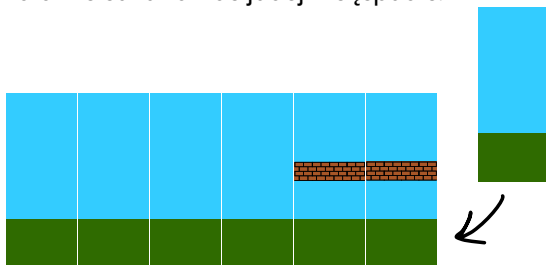
Tai informatinis mąstymas:

Kadangi kompiuterių veikimas pagrįstas logika, dirbant kompiuteriu su logika susidurti tenka ypač dažnai. Pavyzdžiui, programuotojas realaus pasaulio situaciją turi aprašyti loginiais teiginiais, kad tinkamai pateiktų komandas kompiuteriui. Arba gali tekti nustatyti tikslią priešingą sąlygą duotajai, kad rašoma programa galėtų apdoroti visus įmanomus atvejus. Tokioms užduotims reikalingi dekomponavimo įgūdžiai, o tai – informatinio mąstymo dalis. Informatiniam mąstymui labai artimas yra ir loginis mąstymas.

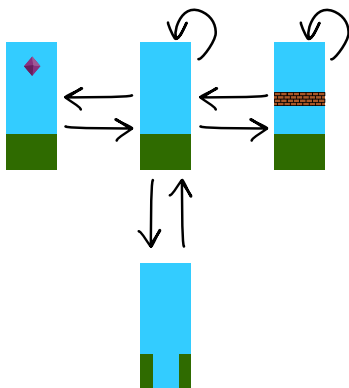
23. Piešiamo žaidimų

2024-DE-07

Kompiuterinio žaidimo foną sudaro plytelių seka. Kompiuteris nuolat prideda po naują plytelę sekoje iš dešinės ir tuo pat metu pašalina po plytelę iš kairės. Tokiu būdu žaidime sukuriamas judėjimo įspūdis.

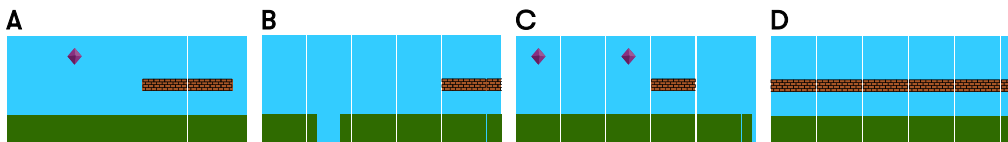


Kompiuteris pridėdamą plytelę parenka pagal pavaizduotą schemą – patikrina iš vėliausiai pridėtos plytelės išeinančias rodykles ir atsitiktinai parenka vieną iš plytelių, į kurias tos rodyklės rodo.



Pavyzdžiui, po tokios plytelės  kompiuteris gali pridėti šią  arba tokią  plytelę. Viename iš paveikslėlių pavaizduotas neteisingas žaidimo fonas. Kuriame?

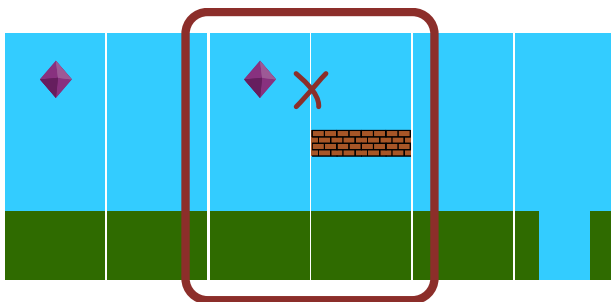
Atsakymų variantai:



Teisingas atsakymas: C.

Paiškinimas

Atsakymo variante C pavaizduotas fonas sudarytas ne pagal pateiktą schemą.



Po plytelės su rombu gali būti vienintelė plytelė , o čia taip nėra.

Yra keletas šio uždavinio sprendimo būdų. Paprasčiausias – tikrinti kiekvieną visų fono paveikslėlių plytelę, ar ši atitinka schemos sąlygas. Greičiausias sprendimo būdas yra toks: iš schemos pasirenkame tas plyteles, iš kurių išeina tik po vieną rodyklę.

Laikykime tokias rodykles tam tikrais ribojimais. $A \rightarrow B$ reiškia, kad po plytelės A turi būti plytelė B. Tuomet patikriname, ar šie ribojimai išlaikomi kiekviename fono paveiksle.

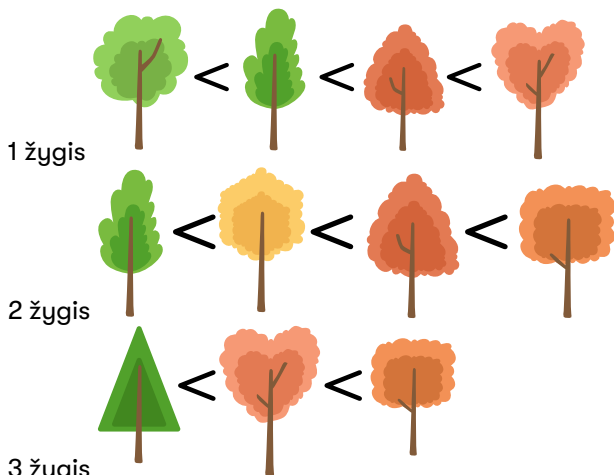
Tai informatika!

Kai kuriuose kompiuteriniuose žaidimuose, pavyzdžiui, tuose, kur veikėjas bėga, fonas juda horizontaliai ir taip sukurama iliuzija, jog veikėjas juda išgalvotame pasaulyje. Kartais toks fonas nėra tik statinis paveikslas, bet yra dinamiškai kuriamas kompiuterio. Tai vadinama procedūriniu generavimu. Jo metu maži elementai atsitiktinai jungiami vienas su kitu, taip sukuriant daugybę skirtingų fonų. Vis dėlto, toks elementų jungimas negali būti visiškai atsitiktinis, o turi atitikti tam tikras taisykles, kad žaidėjas nepatektų į neišsprendžiamą situaciją. Uždavinyje tokios taisyklės vaizduojamos schema, sudaryta iš plytelių ir rodyklių. Tokia schema vadinama orientuotuoju grafu, elementai grafe vadinami viršūnėmis, o rodyklės – orientuotomis briaunomis. Grafi plačiai taikomi modeliavime.

24. Žygis miške

2024-DE-08

Beata siūlo žygius po mišką ir pasakoja apie čia augančius medžius. Iš ankstesnių žygių ji žino, kurie medžiai ypač patinka keliautojams.



Čia 1 medis < 2 medis reiškia, kad 1 medis yra mažiau populiarus nei 2 medis.

Kitame žygyje Beata norėtų populiariesnius medžius parodyti vėliau, todėl nusprendžia rodomus medžius surikiuoti tokia eilės tvarka, kuri atitiktų ankstesnių ekskursijų medžių populiarumą.

Pavyzdžiui, jei Beata planuoja parodyti medžius  ir , tuomet ji pirmiau turi parodyti , o tik paskui , kadangi 3-e žygyje 

buvo populiariesnis nei . Planuojamame žygyje Beata ketina parodyti šiuos medžius:



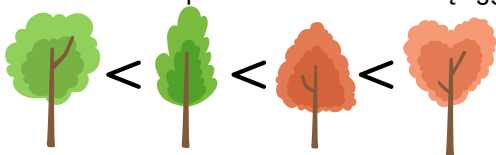
Surikiuokite medžius taip, kad eilės tvarka būtų teisinga.

Teisingas atsakymas:

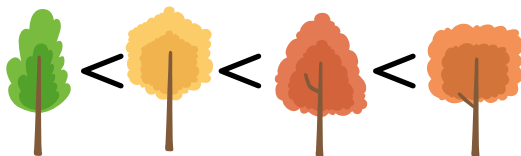


Paiškinimas

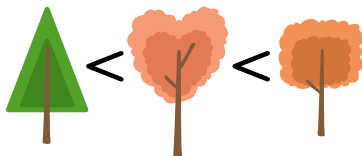
Tinkama tvarka neturi prieštarauti ankstesnių žygių medžių populiarumui:



1 žygis

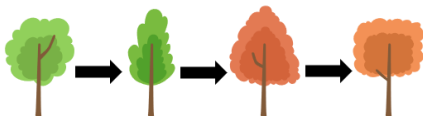


2 žygis

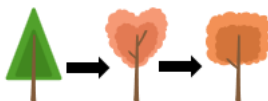
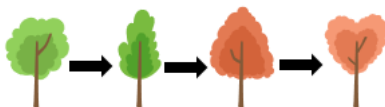


3 žygis

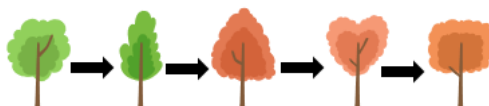
Pabandykime sujungti tris žygius ir pateikti juos kaip diagramą. Diagramoje du medžiai visada yra sujungti rodykle, rodančia, kurį medį reikia aplankyti po kurio. Štai 1 žygio diagrama:



Šią informaciją galima susieti su 3 žygiu. Pirmiausia nustatome, kad tiek 1 žygyje, tiek 3 žygyje buvo aplankytas tas pats medis:



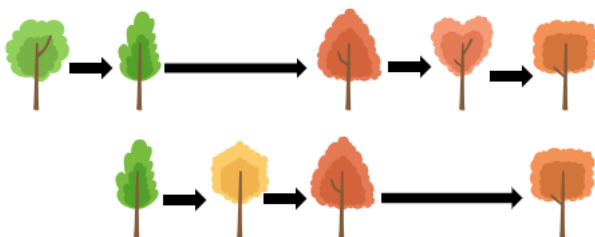
Pastebėjus šį sutapimą galima sujungti diagramas ir gauti:





neįtraukiame, nes negalime tiksliai nustatyti, kurioje diagramos vietoje ji turėtų būti.

Galiausiai bandome ką tik gautą diagramą sujungti su 2 žygio diagrama:



Trys iš keturių medžių jau įtraukti ir tik vieną dar reikia prijungti.

Gauta diagrama išrikiuoja medžius pagal jų populiarumą.

Dabar liko atkreipti dėmesį, kad planuojamame žygyje ne kiekvienas medis bus rodomas.

Gauname tokią seką:



Tai informatika!

Šioje užduotyje reikia medžių rinkinį surikiuoti didėjančia populiarumo tvarka. Turima tam tikros informacijos apie kai kurių medžių porų palyginimą. Tačiau



kai kurių medžių porų, pavyzdžiui, ir , tiesioginio palyginimo nėra. Nepaisant to, šiuos medžius taip pat galima palyginti.

Nagrinėdami šią užduotį atkreipkite dėmesį į dvi svarbias savybes.

1. Populiarumo palyginimas yra tranzityvus, tai yra, jei 1 medis $<$ 2 medis, o 2 medis $<$ 3 medis, tai ir 1 medis $<$ 3 medis.

2. Pateiktų trijų žygių medžių populiarumas vienas kitam neprieštarauja.

Pavyzdžiui, būtų prieštaravimas, jei viename žygyje būtų 1 medis $<$ 2 medis, o kitame jau 2 medis $<$ 1 medis.

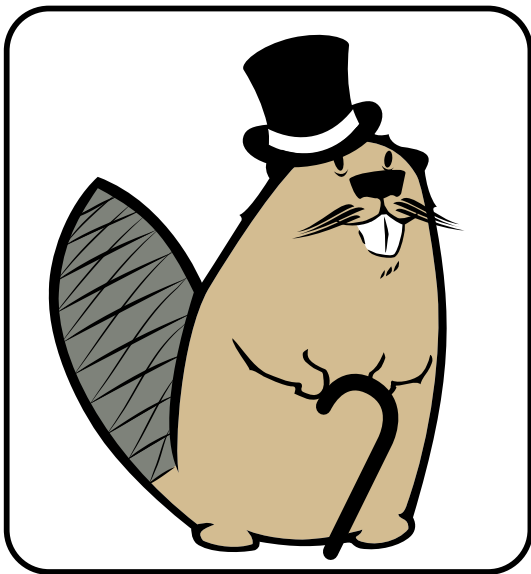
Šių savybių dėka galime kalbėti apie tvarką matematine prasme. Iš pradžių gali atrodyti, kad ši tvarka yra dalinė, nes mes nematome visų medžių palyginimo poromis. Tačiau šios užduoties medžiai orientuotuoju grafu gali būti išdėstyti visiškai surikiuota seka (topologinis rikiavimas). Tam yra gana paprastas algoritmas, tačiau, kad tai veiktų, pirminiuose palyginimuose neturi būti prieštaravimų.

Atidžiau pažiūrėjus galima pastebėti, kad galime vienareikšmiškai surikiuoti visus medžius. Dėl tranzityvumo visoms medžių poroms gauname vienintelį teisingą visų medžių topologinio rikiavimo sprendinį.

25. Tiesa apie bebrą Alfredą

2024-HU-02

Visuomenėje gerai žinomas bebras Alfredas atvyko į pokylį ir buvo nufotografuotas. Apie tai pranešė keturi nuomonių formuotojai, tačiau ne viskas, kas skelbiama socialiniuose tinkluose, yra tiesa.



Kuris iš šių teiginių teisingas?
Atsakymų variantai:

- A) Alfredas buvo užsidėjęs skrybėlę  ir neturėjo lazdos ?
- B) Alfredas ryšėjo kaklaraištį  arba turėjo lazdą ?
- C) Alfredas buvo užsidėjęs skrybėlę  ir ryšėjo peteliškę .
- D) Alfredas ryšėjo peteliškę  arba neturėjo lazdos ?

Teisingas atsakymas: B.

Paaiškinimas

Šioje užduotyje reikia nustatyti, ar pateikti teiginiai apie nuotrauką yra teisingi, ar ne (teisinga ar neteisinga). Kiekvieną teiginį sudaro du atskiri teiginiai, kurie yra sujungti jungtuku „ir“ arba jungtuku „arba“.

Teiginys, kuriame yra jungtukas „ir“, yra teisingas tik tada, jei abu jį sudarantys teiginiai yra teisingi.

Logikoje teiginys, kuriame yra jungtukas „arba“, yra teisingas tik tada, jei bent vienas iš jį sudarančių teiginių yra teisingas.

Užrašant šnekamąją kalbą kai kurie antrieji teiginiai yra sutrumpinti, kad nesikartotų žodžiai. Analizėje pateikiame abu nesutrumpintus teiginius.

A) Alfredas buvo užsidėjęs skrybėlę (teisinga) ir Alfredas neturėjo lazdos (neteisinga).
Visas teiginys yra klaidingas, nes ne abu daliniai teiginiai teisingi.

B) Alfredas ryšėjo kaklaraištį (neteisinga) arba Alfredas turėjo lazdą (teisinga).
Visas teiginys yra teisingas, nes bent vienas dalinis teiginys yra teisingas.

C) Alfredas buvo užsidėjęs skrybėlę (teisinga) ir Alfredas ryšėjo peteliškę (neteisinga).
Visas teiginys yra klaidingas, nes ne abu daliniai teiginiai teisingi.

D) Alfredas ryšėjo peteliškę (neteisinga) arba neturėjo lazdos (neteisinga).
Visas teiginys yra klaidingas, nes bent vienas jo dalinis teiginys nėra teisingas.

Tai informatika!

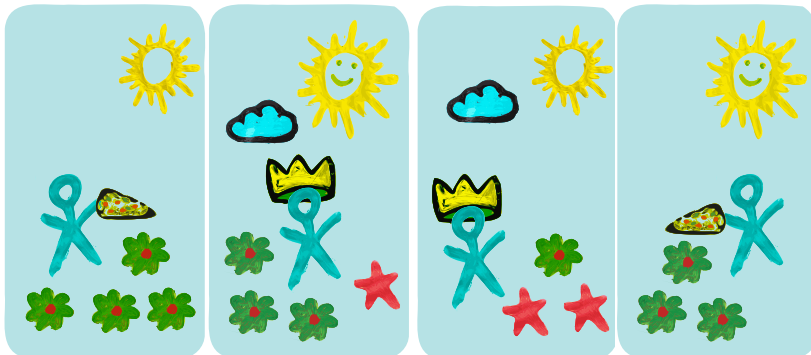
Jungtukai „ir“ (AND) ir „arba“ (OR) yra loginės operacijos, skirtos priimti sprendimams, kai reikia atsižvelgti į kelias sąlygas. Pavyzdžiui, jei asmuo yra 18 metų ar vyresnis ir šis asmuo nenešioja akinių, tai šis asmuo gali gauti vairuotojo pažymėjimą.

Kompiuterių sistemose „teisinga“ ir „neteisinga“ dažnai vaizduojami simboliais 1 (teisinga) ir 0 (neteisinga). Šių loginių operatorių naudojimas leidžia kompiuteriams atlikti sudėtingas logines ir matematines operacijas, kurios yra būtinos daugelyje informatikos sričių. Programuojant tam tikri veiksmai dažnai atliekami tik tuo atveju, jei tam tikras loginis teiginys (sąlyga) yra teisingas.

26. Kortelių piešiniai

2024-IN-04

Manas piešia ant kortelių vadovaudamasis viena paprasta slapta taisykle. Jis sukūrė keturias korteles, kurios atitinka šitą taisyklę:



Anilas apžiūri korteles ir nupiešia dar vieną kortelę, bet Manas sako, kad ji neatitinka taisyklės.



Kuri iš šių taisyklių gali būti slapta taisyklė, kuria vadovaudamasis Manas nupiešė korteles?

Atsakymų variantai:

- A) Kai debesuota, nėra gėlių.
- B) Kai debesuota, saulė nesišypso.
- C) Turi būti arba raudona žvaigždė, arba gabalėlis picos.
- D) Jei yra gabalėlis picos, nėra karūnos.

Teisingas atsakymas: D.

Paaiškinimas

A: Netinka, nes 2-oje ir 3-ioje kortelėse yra ir debesis, ir gėlės.

B: Netinka, nes 3-ioje kortelėje yra debesis ir saulė šypsosi.

C: Netinka, nes visos kortelės atitinka šią taisyklę. Anilo nupiešta kortelė irgi atitinka šią taisyklę, o tai prieštarauja Mano teiginiui.

D: Taip, tai gali būti tinkama taisyklė, nes nė vienoje iš keturių kortelių nėra picos ir karūnos kartu, o Anilo nupieštoje kortelėje yra ir gabalėlis picos, ir karūną, todėl Manas pasakė, kad jo taisyklės nesilaikoma.

Iš pateiktų parinkčių tai vienintelė galima taisyklė: „Jei yra gabalėlis picos, nėra karūnos“.

Tai informatika!

Šis uždavinys iliustruoja pagrindinę logiką, naudojamą informatikos moksle, mokant, kaip galima pateikti ir apdoroti sprendimus atliekant stebėjimus. Kompiuteriai vadovaujasi aiškiomis taisyklėmis, kad nustatytų rezultatus, panašiai kaip mes stebime modelius ir priimame sprendimus kasdieniame gyvenime.

Taisyklių aiškinimas: Jie supranta netiesiogines taisykles, perteiktas per vaizdinius elementus.

Sąlyginė logika: Jie supranta, kad tarp pavaizduotų elementų egzistuoja ryšiai „jei – tai“, padedantys suprasti sąlyginius teiginius ir logines struktūras.

27. Srašas

2024-AT-03

Ana, Benas ir Tomas mėgsta žaisti stiklo rutuliukais ir retkarčiais vienas iš kito juos skolinasi. Kad išvengtų klaidų, paskolintus rutuliukų kiekius surašo ant vieno popieriaus lapo.

Po savaitės jie sulygina savo užrašus:

Tomas

Tomas $\xRightarrow{2}$ Ana
 Benas $\xRightarrow{1}$ Tomas
 Benas $\xRightarrow{3}$ Ana
 Ana $\xRightarrow{2}$ Tomas

Ana

Tomas $\xRightarrow{2}$ Ana
 Benas $\xRightarrow{1}$ Tomas
 Tomas $\xRightarrow{3}$ Ana
 Ana $\xRightarrow{2}$ Tomas

Benas

Tomas $\xRightarrow{2}$ Ana
Benas $\xRightarrow{1}$ Tomas
Ana $\xRightarrow{2}$ Tomas
Tomas $\xRightarrow{3}$ Ana



Draugai įtaria, kad kažkas padarė klaidą. Jei tai tiesa, kas padarė klaidą?

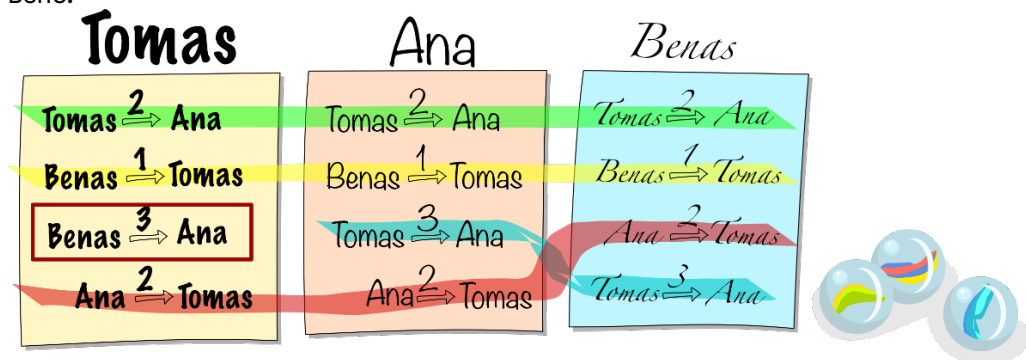
Atsakymų variantai:

- 1) Ana
- 2) Benas
- 3) Tomas
- 4) Nei vienas

Teisingas atsakymas: 3 (Tomas).

Paiškinimas

Žiūrėdami tik į skaičių sekas (2, 1, 3, 2), galime manyti, kad suklydo Benas, nes jo vienintelio seka skiriasi (2, 1, 2, 3). Tačiau taip nėra. Beno sąrašas skiriasi tik transakcijų tvarka, bet bendras rezultatas sutampa. Suklydo Tomas, nes pakeitė operaciją (Benas, Anai, 3) į (Tomas, Anai, 3). Jei tai liktų nepastebėta, jis gautų 3 rutuliukus iš Beno.



Tai informatika!

Kompiuterių moksle kiekvieną atskirą įrašą vadiname transakcija ir saugome duomenų bazėje. Mūsų istorijoje vaikai nusprendė turėti ne tik vieną centrinę duomenų bazę, bet ir kiekvienas po atskirą savo duomenų bazę. Taigi dabar yra kelios fiziškai atskiros duomenų bazės su, kaip tikimasi, vienodu turiniu. Šiame kontekste sąmoningai nuspręsta neturėti centrinės institucijos, kuria visi pasitikėtų. Vaikai yra lygiateisiai ir jų veiksmai vyksta lygiateisių dalyvių tinkle, kuriame kiekvienas dalyvis turi tokias pačias teises ir pareigas kaip ir visi kiti. Dalyvių lygiateisiškumas ir duomenų paskirstymas yra labai svarbus blokų grandinių (angl. *blockchains*) technologijoje.

Uždavinys paminti tas metodus veikia tik tuo atveju, jei dauguma dalyvių yra sąžiningi. Jei kyla nesutarimų, dalyviai susitaria, kurios duomenų bazės duomenys yra teisingi. Blokų grandinių technologija žengia dar vieną žingsnį į priekį įgyvendindama konsensuso algoritmą, kuris reikalautų, kad, pvz., piktavalius veikėjas kontroliuotų daugiau skaičiavimo galios nei visi kiti dalyviai kartu sudėjus. Tai faktiškai neįmanoma, jei tinklas yra pakankamai didelis. Tačiau šis uždavinys jau rodo, kaip pasitikėjimas gali būti sustiprintas a priori nepatikimoje aplinkoje, kai jėga paskirstoma tarp visų dalyvių.

28. Žodžių grandinės

2024-CA-02

Bebras kuria kodus, kurie susideda iš trijų raidžių. Žodžius jis deda į žodžių grandines, kuriose žodžiai surašyti taip, kad greta esantys žodžiai skirtųsi tik viena raide. Pavyzdžiui, XUG → XUD → XED → KED yra žodžių grandinė.

Bebras sukūrė devynis kodus: TOF, XEW, TEF, CET, COF, TEW, COT, CEF ir XEF. Jis grupuoja juos po tris į tris žodžių grandines. Kiekvienas kodas gali būti panaudotas tik kartą.

Visos pateiktos žodžių grandinės sudėtos teisingai, bet naudodami vieną iš jų, negalėsime sukurti trijų žodžių grandinių nekartodami kodų. Kurios iš šių grandinių panaudoti negalime?

Atsakymų variantai:

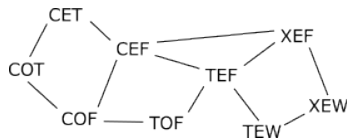
- A. CEF → CET → COT
- B. COF → COT → CET
- C. TEF → CEF → COF
- D. XEW → TEW → TEF

Teisingas atsakymas: C.

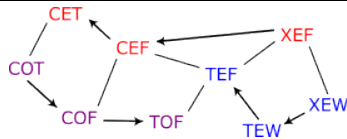
Paiškinimas

Norint išspręsti šią užduotį naudinga nusibraižyti diagramą.

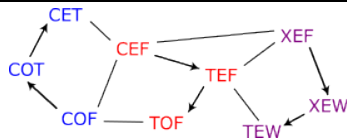
Šioje diagramoje du žodžiai sujungti linija, jei jie galėtų būti greta žodžių grandinėje.



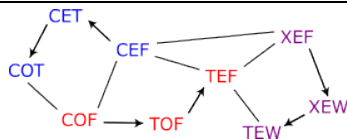
A variantas (XEW → TEW → TEF) žodžių grandinę bebras tikrai galėtų panaudoti. Tokiu atveju kitos dvi žodžių grandinės galėtų būti SAT → CAT → CAB ir COB → COT → BOT, kaip pavaizduota diagramoje:



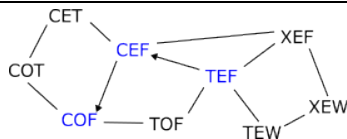
B variantas (COF → COT → CET) žodžių grandinę taip pat galima panaudoti. Šiuo atveju kitos dvi žodžių grandinės galėtų būti: CAT → BAT → BOT ir SAT → SAD → BAD, kaip pavaizduota diagramoje:



D variantas (CEF → CET → COT) žodžių grandinę taip pat galima panaudoti. Tokiu atveju kitos dvi žodžių grandinės galėtų būti: COT → BOT → BAT ir SAT → SAD → BAD, kaip pavaizduota diagramoje:



C variantas (TEF → CEF → COF) žodžių grandinės panaudoti bebrui nepavyks, nes diagramoje TOF sujungtas tik su TEF ir su COF. Duotoji žodžių grandinė panaudoja ir TEF, ir COF, taigi žodžiui TOF nebelieka jokių kitų nepanaudotų žodžių, su kuriais jis galėtų sudaryti žodžių grandinę.



Tai informatika!

Vienas iš būdų spręsti įvairius informatikos uždavinius – naudoti specialias diagramas, vadinamas grafais. Grafą sudaro viršūnės (šioje užduotyje – žodžiai) ir briaunos, jungiančios viršūnes. Šioje užduotyje briaunomis sujungiami žodžiai, galintys būti greta vienas kito žodžių grandinėje. Grafai patogūs tuo, kad leidžia aiškiai pamatyti visus sąryšius. Be to, naudojant grafus taip pat galima taikyti ir įvairias sprendimo strategijas, pavyzdžiui, spalvinti viršūnes, kaip buvo daryta ir sprendžiant šią užduotį.

Grafai yra populiarius įrankis, mokslininkų ir inžinierių naudojamas sudėtingoms situacijoms modeliuoti. Kompiuterinės programos, naudodamos grafų logiką, gali išspręsti daugybę problemų, kurias būtų sudėtinga spręsti kitais būdais. Pavyzdžiui, navigacijos sistemos – tai programos, kurios naudoja labai didelius grafus, kad žingsnis po žingsnio pavaizduotų kelionę nuo pradinio taško iki kelionės tikslo. Galima įsivaizduoti, kad kiekviename žingsnyje automobilis „važiuoja grafo briauna“ nuo vienos grafo viršūnės (pavyzdžiui, sankryžos) link kitos grafo viršūnės (pavyzdžiui, kitos sankryžos).

29. Balionų mašina

2024-DE-06a

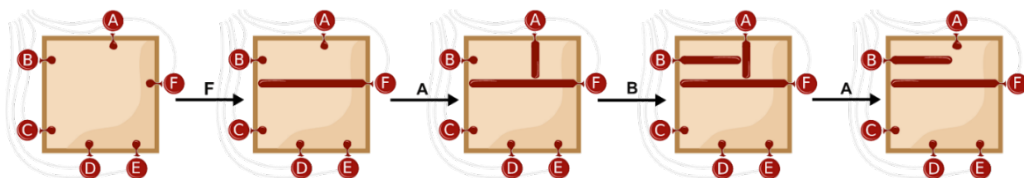
Balionų pūtimo mašina kuria ornamentus pripūsdama balionus kvadratiname rėmelyje.

Balionai sužymėti raidėmis A, B, C, D, E ir F.

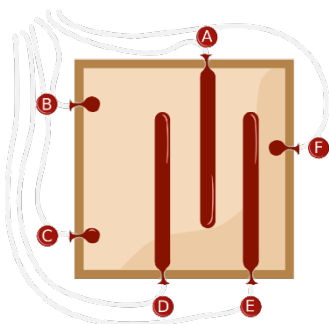
Mašinai pateikiamos raidės, kurias ji skaito po vieną iš kairės į dešinę. Perskaičiusi raidę mašina atlieka tokius veiksmus:

- jeigu tos raidės balionas nebuvo pripūstas, jis bus pučiamas į ilgį tol, kol atsirems į priešais esantį rėmelio kraštą ar jau pripūstą balioną.
- jeigu balionas jau buvo pripūstas, jis bus išleidžiamas.

Pavyzdžiui, kai nė vienas balionas nebuvo pripūstas ir mašinai buvo pateikta tokia raidžių seka: F, A, B, A, pūtimo veiksmai atrodys taip:



Pradžioje nė vienas balionas nebuvo pripūstas, o mašina, gavusi devynių raidžių seką, sukūrė šitokį ornamentą.

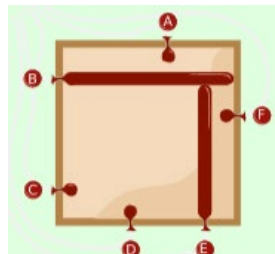


Vienas iš teisingų atsakymų yra: B E B C A C B D B.

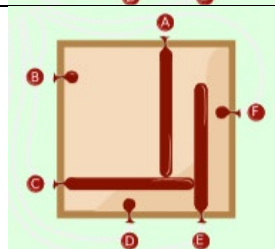
Paaiškinimas

Jvykdžius raidžių B ir E komandas balionai rėmelyje atrodo taip:

Matyti, kad balioną B reikėjo pripūsti pirmiau nei E, kad B sudarytų kliūtį balionui E ir pastarasis būtų trumpesnis.

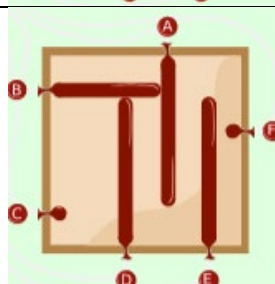


Po sekos B E B C A įvykdymo balionai atrodo taip:



Balionas C tampa kliūtimi balionui A, todėl turi būti pripūstas pirmiau nei A. Be to, balioną B reikia išleisti prieš pučiant balioną A, kitaip balionas A nepasieks reikiamo ilgio.

Po sekos B E B C A C B D įvykdymo balionai rėmelyje atrodo taip:



Balionas B sudaro kliūtį balionui D, o balioną C reikia išleisti prieš pučiant balioną D. Iš viso galimi 4 teisingi atsakymai:

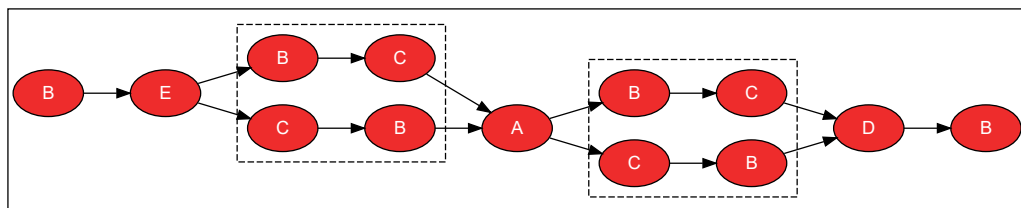
B E B C A C B D B

B E C B A C B D B

B E B C A B C D B

B E C B A B C D B

Visus teisingus atsakymus rasti bus lengviau, jei uždavinį pavaizduosime orientuotuoju grafu naudojant topologinį rikiavimą. Balionus pažymėkime raidėmis. Rodyklė iš B į E rodo, kad balionas B turi būti pripūstas anksčiau, nei balionas E ir t. t. Yra du atvejai, kai balionų B ir C eiliškumas gali skirtis (pažymėta punktyriniais stačiakampiais).



Kiekvienas teisingas atsakymas (pripučiamų balionų seka) yra šio grafo topologinis išrikiavimas. Šiuo atveju turime 4 topologinius išrikiavimus, todėl galimi keturi teisingi sprendimai.

Pastebėjimas sprendimo teisingumui įvertinti: galutiniame ornamente pripūstų balionų raidės (šiuo atveju A, D ir E) sekoje turi būti pateiktos nelyginį kartų skaičių. Tų balionų, kurie galutiniame ornamente yra nepripūsti (čia B, C ir F), raidės sekoje turi kartotis lyginį kartų skaičių (įskaitant 0).

Tai informatika!

Raidžių seka yra tarsi komandų seka, kuri nurodo mašinai, ką daryti. Kiekviena raidė yra tarsi komanda, kuris priverčia mašiną pripūsti arba išleisti balioną. Kaip ir daugumoje tokio tipo kompiuterinių programų, esminę reikšmę turi komandų eiliškumas. Pavyzdžiui, seka B E priverčia mašiną sukurti kitokį vaizdą nei seka E B.

Atsakymo paaiškinime naudojamas orientuotasis beciklis grafas. Tokie grafai visuomet turi bent vieną topologinį išrikiavimą. Topologiniam išrikiavimui sudaryti yra žinomi greitieji algoritmai. Topologiniai išrikiavimai plačiai taikomi ne tik informatikoje: tvarkaraščių sudarymui, failų versijų tvarkymui, projektų planavimui ir kt.

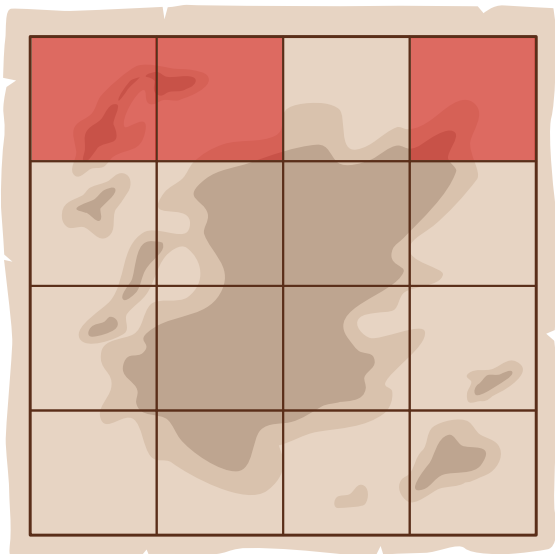
30. Lobis

2024-HU-03

Piratas Pipas ieško lobio negyvenamoje saloje.

Pipas turi salos žemėlapij, kuriame parodyta, kur paslėptas lobis. Žemėlapis padalintas į 16 vienodų langelių. Pipas turi įtaisą, kuriame gali pažymėti bet kokį skaičių langelių ir gauti atsakymą, ar lobis yra kuriame nors iš nurodytų langelių, ar ne.

Pavyzdžiui, jei įtaisas rodo „Taip“, lobis yra viename iš pažymėtų langelių.



Pipas nori kuo greičiau sužinoti, kuriame langelyje yra lobis.

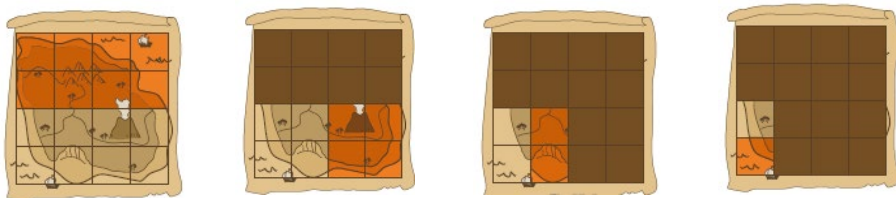
Kiek mažiausiai kartų Pipas turi pasinaudoti įtaisu, kad rastų lobį?

Teisingas atsakymas: 4.

Paaškinimas

Pipas pasirenka strategiją, kai kiekvieną kartą langelių aibę dalijama perpus. Lobis gali būti bet kuriame iš 16 langelių. Taigi 1-u bandymu Pipas nurodo 8 jo pasirinktus langelius. Jei įtaisas atsako „taip“ – lobis paslėptas kažkuriame iš nurodytų 8 langelių. Jei įtaisas atsako „ne“ – lobis paslėptas kažkuriame iš likusių nenurodytų 8 langelių. Tuomet 2-u bandymu Pipas renka 4 langelius iš tų 8 langelių, kuriuose lobis yra. Jei pataiko, tai tuos 4 langelius dalija pusiau ir 3-u bandymu renka 2 kitus langelius. Jei nepataiko, tai pusiau dalija likusius 2 langelius. Ketvirtuoju klausimu Pipas renka vieną iš dviejų likusių langelių. Jei atsakymas „taip“, tai lobis yra pasirinktame langelyje, jei atsakymas „ne“, tai lobis kitame langelyje.

Paveikslėliuose parodyta galima klausimų su pasirinktais langeliais seka. Langeliai, kuriuos įtaisu nurodo Pipas, pažymėti šviesiai rusva spalva. Antrajame paveikslėlyje parodyta, kad įtaisas atsakė „ne“, todėl Pipas renka 4 langelius iš apatinių 2 eilučių. Įtaisas vėl atsakė „ne“, todėl Pipas iš likusių 4 langelių renka du ir t. t. Iš paskutinių dviejų langelių pasirinkus kampinį langelį įtaisas vėl atsakė „ne“, todėl Pipas jau žino, kad lobis yra paskutinio paveikslėlio šviesiame langelyje.



Tai informatika!

Metodas, kurį Pipas naudoja ieškodamas lobio, informatikoje vadinamas dvejetainine (binarine) paieška. Terminas binarinis yra kilęs iš lotyniško žodžio bis (du kartus). Aibėje ieškant reikiamo objekto dvejetainės paieškos metodu, objektų aibę vis dalijama per pusę (į dvi dalis) užduodant klausimus, – iš čia ir žodis „dvejetainė“. Aibę galima padalyti pusiau, jei joje esančius objektus galima surikiuoti, pavyzdžiui, pagal dydį; tai, be kita ko, taikoma kiekvienai skaičių aibei. Jei yra nelyginis objektų skaičius, tuomet aibėje yra vidurinis objektas ir jį galima palyginti su ieškamu objektu. Jei vidurinis objektas nėra tas, kurio ieškote, vis tiek žinote, kurioje aibės pusėje yra ieškomas objektas, ir šioje aibės dalyje vėl ieškote dvejetainiu būdu. Taip labai greitai rasite ieškomą objektą. Kai ieškoma 1000 objektų, reikia atlikti apie 10 paieškos veiksmų, o kai ieškoma 1000000 objektų – apie 20. Apskritai galima sakyti, kad ieškant tarp n objektų reikia atlikti maždaug $\log(n)$ veiksmų. Funkcija $\log$ yra logaritmas, kurio pagrindas 2. Kadangi dvejetainė paieška labai greita, ji dažnai naudojama kompiuterių programose tvarkingiems duomenims ieškoti.

Šioje užduotyje paieškos erdvė yra salos žemėlapis langelių aibė. Langelius galima sutvarkyti sunumeravus juos iš viršaus į apačią ir iš kairės į dešinę. Tačiau šiuo atveju taip pat reiktų perpus sumažinti langelių, kuriuose dar galima

rasti lobį, skaičių. Tada reikia prisiminti, kuriame langelių rinkinyje dar galima rasti lobį kitu paieškos etapu ir jį vėl sumažinti perpus.

Nykštukai

Mažyliai

Bičiuliai

Draugai

12

Jauniai

9

Kolegos

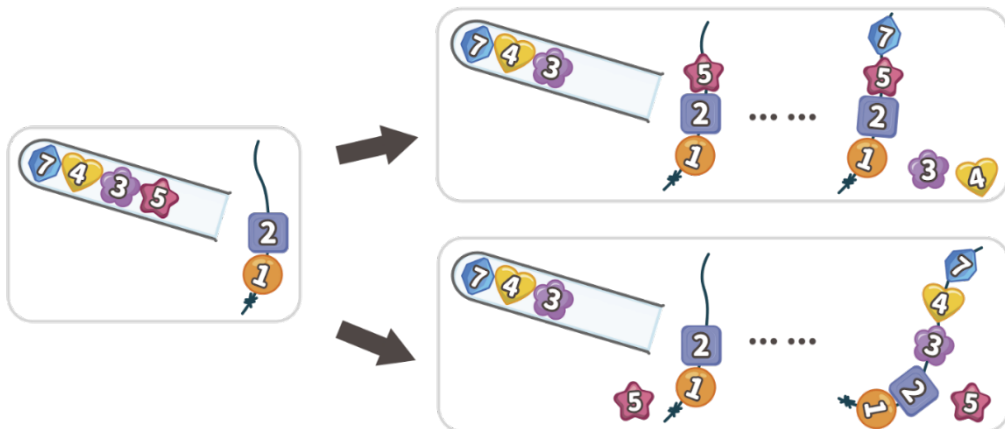
6

31. Ilgiausia apyrankė

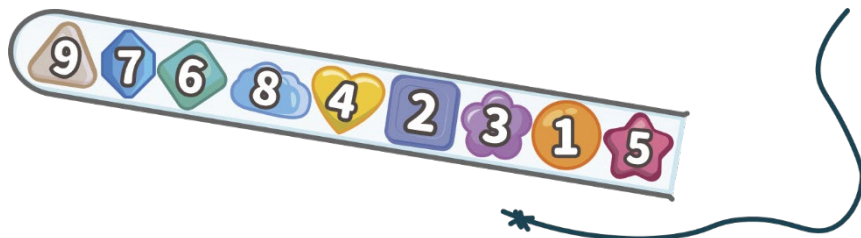
2024-TW-03

Bebras gamina apyrankę. Iš stiklinio vamzdelio jis po vieną ima sunumeruotus karoliukus ir pasirenka, ar verti ant virvelės, ar padėti į šalį ir nenaudoti. Bebras gali užverti karoliuką ant virvelės tik, jei: virvelė tuščia arba paimto karoliuko numeris yra didesnis nei ką tik užveto karoliuko.

Paveiksle pateiktame pavyzdyje vėliausiai užvertas ant virvelės yra karoliukas nr. 2. Taigi, bebras gali užverti karoliuką Nr. 5 ant virvelės arba padėti į šalį ir jo nenaudoti. Jei bebras užvers karoliuką Nr. 5, galės suverti apyrankę iš keturių karoliukų: 1257. Jei nenaudos karoliuko Nr. 5, galės suverti daugiau karoliukų: 12347.



Bebras kuria naują apyrankę iš karoliukų, esančių stikliniame vamzdelyje. Suverkite ilgiausią įmanomą apyrankę pagal minėtas taisykles.



Teisingas atsakymas: 124679 arba 134679.

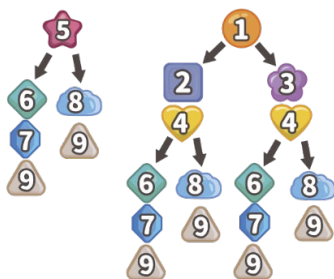
Paaiškinimas

Galime sudaryti visas įmanomas apyrankes ir pamatysime, kurios turi daugiausiai karoliukų. Tačiau tai sudėtinga. Atidžiau pažvelkime į skaičius:



Pirmojo karoliuko numeris yra 5. Jei jis bus užvertas, didesni numerį turintys karoliukai yra tik 6, 7, 8 ir 9, todėl galimos sekos būtų 5679 arba 589.

Jei karoliukas Nr. 5 atidedamas į šalį, o kitas karoliukas su numeriu 1 užveriamas, yra daugiau ir ilgesnių galimų variantų, kas aiškiai parodyta čia:



Jei ant virvelės pirmiausia užveriamas koks nors kitas karoliukas, ilgiausia įmanoma galutinė seka jau įtraukta į vieną iš aukščiau pateiktų, pradedant nuo 1 arba 5. Pavyzdžiui, ant virvutės užvėrus karoliuką Nr. 2 pirmiausia atsiranda sekos 24679 arba 2489, kurios abi įtrauktos į vieną iš aukščiau pateiktų ilgesnių sekų (atitinkamai 124679 ir 12489).

Iš to mes suprantame, kad apyrankės, turinčios daugiausiai karoliukų iš stiklinio vamzdelio, prasideda nuo karoliuko Nr. 1 ir susideda iš 6 karoliukų: 124679 arba 134679.

Tai informatika!

Kompiuterių moksle „seka“ reiškia duomenų išdėstymą į eilę, pavyzdžiui, simbolių seką (dar vadinama eilute) arba skaičių seką. Kiekvienas sekos narys vadinamas elementu, o posekis sudaromas pasirenkant elementus iš originalios sekos nekeičiant šių elementų tvarkos.

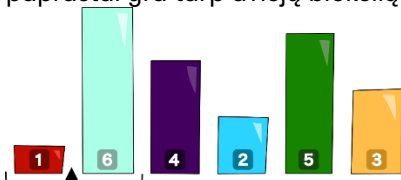
Tiek karoliukai vamzdelyje (tokia tvarka, kokia karoliukai iškrenta iš vamzdelio), tiek apyrankės gali būti laikomi skaičių sekomis, jei kitos savybės, kaip forma ir spalva, ignoruojamos. Ši užduotis prašo surasti ilgiausią didėjančią seką vamzdelyje, tai yra, ilgiausią posekį, kuriame skaičiai tik didėja.

Sekoms, kurios yra (daug) ilgesnės nei vamzdelio seka šioje užduotyje, visų galimų didėjančių posekių sudarymas, siekiant rasti ilgiausią, paprastai užtruks per daug laiko, kad būtų įmanoma tai atlikti. Laimei, kompiuterių mokslininkai sukūrė efektyvius algoritmus, kurie greitai randa ilgias didėjančias sekas. Problemos, susijusios su specifinių posekių radimu, dažnai gali būti sprendžiamos naudojantis dinaminio programavimo metodais.

32. Statybinių blokelių mašina

2024-BE-01a

Mašina gali perkelti šešis skirtingo aukščio blokelių. Tam ji naudoja žymeklį, kuris paprastai yra tarp dviejų blokelių. Pradžioje blokeliai ir žymeklis išdėstyti taip:

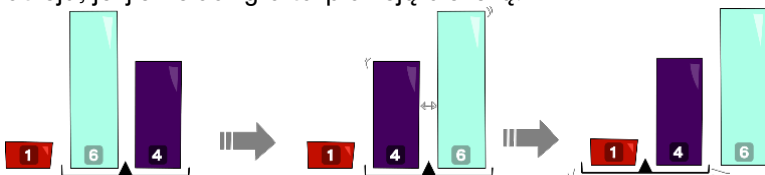


Mašina veikia pagal šias taisykles:

Jei į kairę nuo žymeklio esantis blokelis yra žemesnis už blokelį žymeklio dešinėje, žymeklis pasislenka į dešinę.

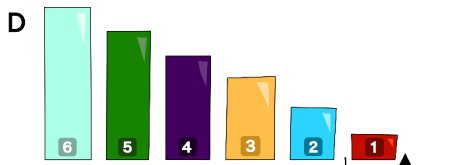
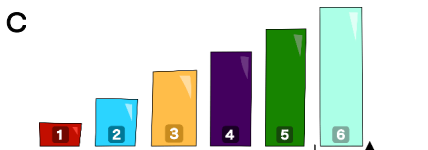
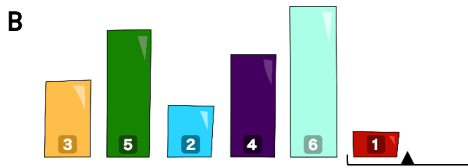
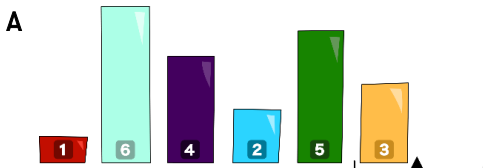


Jei į kairę nuo žymeklio esantis blokelis yra aukštesnis už blokelį žymeklio dešinėje, blokeliai sukeičiami vietomis. Atliekant šį veiksmą žymeklis pasislenka į kairę tik tuo atveju, jei jis vis dar yra tarp dviejų blokelių.



Mašina sustoja, kai žymeklis atsiduria šalia blokelių dešinėje pusėje. Kaip bus išsidėstę blokeliai, kai mašina sustos?

Atsakymų variantai:



Teisingas atsakymas: C.

Paiškinimas

Svarbu išsiaiškinti, kad visi blokeliai, esantys į kairę nuo žymeklio, visada yra išrikiuoti. Žymeklis juda į dešinę tik tada, kai blokelis, esantis į kairę nuo žymeklio, yra žemesnis už blokelį, esantį į dešinę nuo jo. Jei į kairę perkliamas žemesnis blokelis, žymeklis taip pat pasislenka į kairę. Taip aiškiai parodoma, kad visi blokeliai į kairę nuo žymeklio jau yra surikiuoti didėjimo pagal aukštį tvarka.

Žemesnis blokelis keliamas toliau į kairę, kol tampa didesniu už blokelį, esantį kairėje (taigi ir už visus kitus, esančius kairėje).

Tada žymeklis vėl pasislinks į dešinę, kol atsiras kitas mažesnis blokelis. Taip blokeliai išrikiuojami žingsnis po žingsnio didėjimo tvarka.

Kai visi blokeliai surikiuojami didėjimo tvarka, žymeklis juda į dešinę, kol atsiduria blokelių dešinėje, ir mašina sustoja.

Tai informatika!

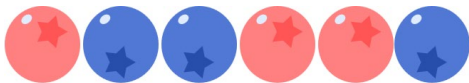
Duomenų rikiavimas yra įprastas procesas informatikoje – rikiuojama pagal dydį, kainą ar abėcėlę. Todėl informatikos mokslininkai ieško kuo efektyvesnių rikiavimo metodų.

Šioje užduotyje naudojamas metodas vadinamas Gnomu rikiavimu. Jis panašus į rikiavimą didėjimo tvarka (Burbulo rikiavimo algoritmas). Jis nėra itin efektyvus – gerokai prastesnis, negu efektyvūs rikiavimo metodai, pavyzdžiui, greito rikiavimo algoritmas ([angl.](#) quicksort) ar sąlaivos rikiavimas ([angl.](#) mergesort). Kita vertus, Gnomu rikiavimas yra lengviau suprantamas.

33. Kamuoliukai

2024-BG-01b

Duota raudonų ir mėlynų kamuoliukų seka:



Skaičiuojame mėlynus kamuoliukus iš kairės į dešinę, pradėdami nuo pirmo kamuoliuko, tuomet vėl skaičiuojame mėlynus kamuoliukus, tačiau pradėdami nuo antrojo kamuoliuko ir t. t. Tokiu būdu skaičiuodami gauname štai tokią seką:

3, 3, 2, 1, 1, 1

Iš gautos sekos sudarome naują seką. Jei skaičius lyginis – rašome 0, jei nelyginis – rašome 1. Gauname štai tokią eilutę:

1, 1, 0, 1, 1, 1

Duota tokia seka, sudaryta iš 0 ir 1:

0, 1, 1, 1, 0, 1, 0, 0

Sudėkite raudonų ir mėlynų kamuoliukų seką, iš kurios aprašytu metodu būtų galima gauti šią seką.

Teisingas atsakymas:



Paiškinimas

Šį atsakymą galima gauti keliais būdais. Vienas iš būdų yra imti po vieną skaitmenį iš duotosios 0 ir 1 eilutės **iš dešinės** į kairę.

Pirmiausia galime pastebėti, kad dešiniausias kamuoliukas yra raudonas, nes jei jis būtų mėlynas, tai paskutinis eilutės skaitmuo būtų 1.

0	1	1	1	0	1	0	0
?	?	?	?	?	?	?	raudonas

Antroje pozicijoje iš dešinės yra 0. Nulį galime gauti tik tuomet, kai abu paskutiniai kamuoliukai yra mėlyni arba abu paskutiniai kamuoliukai yra raudoni. Kadangi paskutinis kamuoliukas yra raudonas, tai ir priešpaskutinis kamuoliukas turi būti raudonas.

0	1	1	1	0	1	0	0
?	?	?	?	?	?	raudonas	raudonas

Trečioje pozicijoje iš dešinės yra 1, kuris rodo, kad mėlynų kamuoliukų skaičius nuo šio taško turėtų būti nelyginis. Kadangi turime jau du raudonus kamuoliukus, tai trečiasis nuo galo kamuoliukas tikrai yra mėlynas.

0	1	1	1	0	1	0	0
?	?	?	?	?	mėlynas	raudonas	raudonas

Ketvirtas skaitmuo iš dešinės yra 0. Tai rodo, kad mėlynų kamuoliukų skaičius nuo šio taško turėtų būti lyginis. Kadangi iki šiol turime tik vieną mėlyną kamuoliuką, tai ir ketvirtas nuo galo kamuoliukas turi būti mėlynas.

0	1	1	1	0	1	0	0
?	?	?	?	mėlynas	mėlynas	raudonas	raudonas

Jau galime pastebėti tendenciją, kad žiūrint iš dešinės pusės eilutėje skaitmuo lieka toks pats, kaip esantis dešiniau, jei toje pozicijoje yra raudonas kamuoliukas. Jei skaitmuo skiriasi nuo esančio iš dešinės, tuomet toje pozicijoje yra mėlynas kamuoliukas.

Taigi, galiausiai gauname štai tokią kamuoliukų seką:

0	1	1	1	0	1	0	0
skiriasi	Toks pats	Toks pats	skiriasi				
mėlynas	raudonas	raudonas	mėlynas	mėlynas	mėlynas	raudonas	raudonas

Tai informatika!

Šioje užduotyje susidūrėme su kodavimu, viena iš esminių informatikos sąvokų. Kodavimas – tai veiksmas, kuriuo ženklui arba kitokiam objektui suteikiamas kodas. Šiame pavyzdyje spalvotiems kamuoliukams buvo priskirtas kodas iš skaitmenų 0 ir 1. Dėl techninių priežasčių skaitmenimis 0 ir 1 koduojama labai dažnai, nes tik taip užkoduotą informaciją kompiuteriai ir gali apdoroti. Yra daugybė kodavimo būdų. Yra informacijos, kurios neįmanoma užkoduoti dalies jos neprarandant. Tačiau yra ir būdų taip užkoduoti informaciją, kad ji užimtų dar mažiau kompiuterio atminties, nei saugant įprastu pavidalu. Pavyzdžiui, šioje užduotyje gaunamas kodas nesaugo kiekvieno kamuoliuko spalvos ar grafinio vaizdo ir taip galima gerokai sutaupyti užimamos kompiuterio atminties. Nepaisant to, žinodami, kaip buvo koduojama, galime nesunkiai atgauti pradinę informaciją – spalvotų kamuoliukų eilę. Tai galioja ir kitiems kodavimo būdams: norint iškoduoti informaciją, reikia žinoti, kaip informacija buvo užkoduota (koks algoritmas buvo panaudotas).

Algoritmai, kuriuos naudojant failo dydis tai pačiai informacijai išsaugoti sumažėja, yra vadinami glaudinimo algoritmais. Pavyzdžiui PNG formato paveikslėliams išsaugoti naudojama keletas glaudinimo algoritmų, taigi tokie paveikslėliai užima mažiau kompiuterio atminties.

34. Maisto žemėlapis

2024-IT-01b

Bebras Nojus paslėpė savo maisto atsargas po 9 iš 17 medžių, augančių greta tvenkinio. Nojus nubraižė žemėlapij, kurio kiekviename palei krantą esančiame langelyje užrašė skaičių, žymintį, kiek su tuo langeliu ribojasi medžių, po kuriais yra paslėpta maisto.

Pavyzdžiui, langelyje raudonai apvestas skaičius 3 reiškia, jog po trimis iš penkių pažymėtų besiribojančių medžių yra maisto.

						
	2	2	1	1	2	
	2				2	
	3	2	1		2	
			1		1	

Nojaus draugė bebrukė Birutė sukritikavo tokį Nojaus sumanymą ir buvo teisi! Iš tiesų, pagal savo sudarytą žemėlapij bebras Nojus užtikrintai galės nustatyti tik 7 iš 9 medžių, po kuriais yra paslėpta maisto.

Kurie tai 7 medžiai?

						
	2	2	1	1	2	
	2				2	
	3	2	1		2	
			1		1	

Paaiškinimas

Septyni medžiai, po kuriais yra paslėpta maisto, paveiksle žemiau pažymėti žvaigždutėmis.

	A	B	C	D	E	F	G
1							
2		2	2	1	1	2	
3		2				2	
4		3	2	1		2	
5				1		1	

Žemėlapyje galima nustatyti du medžius, po kuriais tikrai yra maisto, ir vieną medį, po kuriuo maisto tikrai nėra (pažymėtas „X“).

	A	B	C	D	E	F	G
1							
2		2	2	1	1	2	
3		2				2	
4		3	2	1		2	
5	X			1		1	

Pradėkime nuo žemėlapyje apačioje ir viduryje esančio langelio ir nagrinėkime medžius eidami kairėn. Po pirmuoju medžiu turi būti maisto, kad būtų įvykdyta sąlyga, nustatyta greta esančiais vienetais. Po antruoju medžiu taip pat turi būti maisto, nes greta yra dvejetas. Jei maisto būtų ir po medžiu, esančiu apatiniame kairiajame kampe, tuomet negalėtume įvykdyti dvejeta, apibrėžto apskritimu, sąlygos. Panašiai mąstydami, kairiojoje žemėlapyje dalyje galime nesunkiai rasti medžius, po kuriais dar yra maisto.

	A	B	C	D
1	X			
2		2	2	1
3	X	2		
4		3	2	1
5	X			1

Tuo tarpu dešiniojoje žemėlapyje dalyje galimi du variantai:

D	E	F	G	D	E	F	G
	X	X			X	X	?
1	1	2	?	1	1	2	
		2				2	
1		2		1		2	?
1		1	?	1		1	

Tad čia galime pažymėti tik vieną medį.

Pabandykime formalizuoti užduotį. Žemėlapyje apačioje žvaigždutėmis pažymėję pirmuosius du medžius, po kuriais tikrai yra maisto, kitus medžius pažymėkime raidėmis.

E	F	G	H	I	J	K
D	2	2	1	1	2	L
C	2				2	M
B	3	2	1		2	N
A			1		1	O

Skaičiai mėlynuose langeliuose reiškia sąlygą. Pradėję nuo žemėlapyje vidurio apačioje nagrinėkime sąlygas eidami pagal laikrodžio rodyklę. Pirmosios trys sąlygos (1, 1, 2) jau yra tenkinamos. Kad būtų tenkinama sąlyga „3“, iš aibės {A, B, C} reikia parinkti vieną elementą. Tokiu būdu visą užduotį galima formalizuoti šitaip:

- iš aibės $\{A, B, C\}$ parinkti vieną elementą;
- iš aibės $\{B, C, D\}$ parinkti du elementus;
- iš aibės $\{C, D, E, F, G\}$ parinkti du elementus;
- iš aibės $\{F, G, H\}$ parinkti du elementus;
- iš aibės $\{G, H, I\}$ parinkti vieną elementą;
- iš aibės $\{H, I, J\}$ parinkti vieną elementą;
- iš aibės $\{I, J, K, L, M\}$ parinkti du elementus;
- iš aibės $\{L, M, N\}$ parinkti du elementus;
- iš aibės $\{M, N, O\}$ parinkti du elementus;
- iš aibės $\{N, O\}$ parinkti vieną elementą.

Patikrinkime visus variantus: iš pirmosios aibės negalima parinkti elemento A, nes tuomet reikėtų rinktis B arba C iš antrosios; jei iš pirmosios aibės parinktume elementą B, tuomet reikėtų rinktis elementą D iš antrosios ir t. t. Šitaip gauname du galimus sprendinius:

$\{B, D, F, H, K, M, N\}$, $\{B, D, F, H, L, M, O\}$.

Elementai (medžiai), kuriuos tikrai galima pažymėti žvaigždutėmis, yra B, D, F, H, M (dviejų gautų sprendinių aibių sankirta), o jie ir atitinka sprendimo pradžioje pateiktą paveikslėlį.

Tai informatika!

Ši užduotis paremta gerai žinomu loginiu kompiuteriniu žaidimu. Yra daug jo variantų, o vienas iš geriausiai žinomų – „Išminuotojas“. Šiame žaidime teisingą sprendimą gali lemti ir sėkmė. Net ir atvėrus didelį plotą gali pasitaikyti situacijų, kuomet nepakanka informacijos saugiai tęsti žaidimą, tad tenka kliautis sėkme. Būtent tokia situacija naudojama ir uždavinys: dešinėje tvenkinio pusėje saugaus ir garantuoto sprendimo nėra.

Bendruoju atveju, jei aplink tvenkinį yra n medžių, juos galima išdėstyti $2n$ variantų. Tuomet žymiai lengviau yra patikrinti, ar duotas sprendimas atitinka uždavinys sukurtas sąlygas, nei tą sprendimą surasti!

Šia užduotimi taip pat parodoma, kaip dažnai uždavinį yra lengviau spręsti dalimis. Informatikos požiūriu įdomiausia dalis yra uždavinio formalizavimas. Pritaikant tinkamą duomenų struktūrą (kaip ir padaryta paskutinėje paaiškinimo dalyje) parodoma, kokius sudėtingus skaičiavimus reikia atlikti uždaviniui išspręsti, o ir sprendimų gali būti ne vienas.

35. Draugai

2024-LT-05

Klasėje mokosi 7 bebrai: Adas, Beata, Cilė, Domas, Erikas, Fredas ir Ginta. Kai kurie bebrai pažįsta vienas kitą, kiti – ne.

Lentelėje pažįstamų bebrų poros pavaizduotos varnelėmis.

	Adas	Beata	Cilė	Domas	Erikas	Fredas	Ginta
Adas		✓	✓				
Beata	✓						✓
Cilė	✓				✓		
Domas						✓	
Erikas			✓				
Fredas				✓			
Ginta		✓					



Kiekvienas bebras iš ne savo klasės draugų gavo po skirtingą žinutę ir pasidalijo ją su visais savo klasės pažįstamais bebrais. Kai tik bebras gauna naują žinutę iš besimokančio klasėje bebro, jis iš karto gauta žinutę dalijasi su visais savo klasės pažįstamais bebrais.

Kuris (-ie) šios klasės bebras (-ai) iš viso gavo mažiausiai žinučių?

Atsakymų variantai:

Adas	Beata	Cilė	Domas	Erikas	Fredas	Ginta
------	-------	------	-------	--------	--------	-------

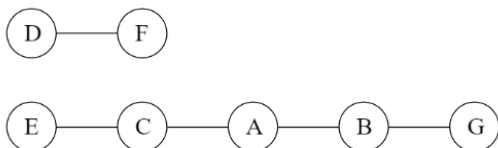
Teisingas atsakymas: Domas ir Fredas.

Paiškinimas

Klasės bebrų tarpusavio santykius galima pavaizduoti grafu. Kiekvienas bebras vaizduojamas apskritimu (grafo viršūnė), kurio centre įrašoma vardo pirma raidė. Jei du bebrai pažįstami, tai juos vaizduojantys apskritimai sujungiami atkarpa (grafo briauna).

Domas pažįsta tik Fredą, o Fredas pažįsta tik Domą, todėl jų apskritimai sujungti tik tarpusavyje.

Cilė pažįsta ir Adą, ir Eriką, todėl Cilės apskritimas sujungtas su Eriko ir Ado apskritimais. Beata pažįsta ir Adą, ir Gintą, todėl Beatos apskritimas sujungtas su Ado ir Gintos.



Nejungusis grafas vaizduoja, kad Domas gali gauti klasėje žinutę tik iš Fredo, o Fredas – tik iš Domo. Jie gali gauti tik po 2 žinutes.

Kiti klasės bebrai gavo po 5 žinutes, taigi, Domas ir Fredas gavo mažiausiai žinučių.

Tai informatika!

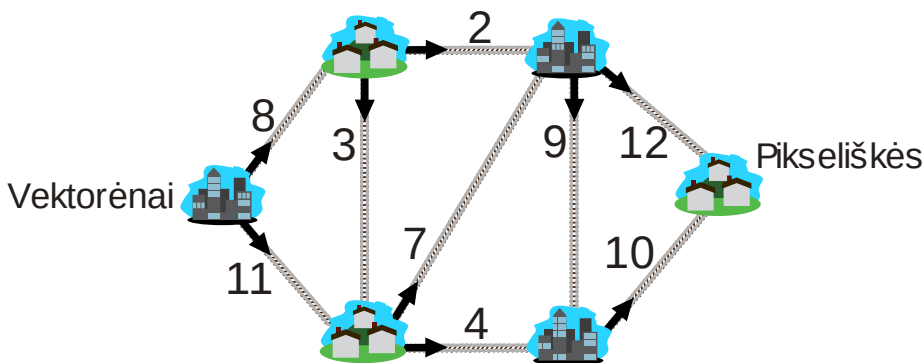
Grafai yra svarbi abstrakcijos priemonė informatikoje. Šioje užduotyje grafas pateikiamas kaip gretimybių (kaimynystės) matrica. Bebrų draugystės (ryšių) uždavinys susijęs su grafo jungumu. Šiuo atveju nagrinėjamas nejungusis grafas. Grafų teorijoje ir informatikoje gretimybių matrica yra kvadratinė matrica, naudojama baigtiniam grafiui atvaizduoti. Matricos elementai rodo, ar grafo viršūnių poros yra gretimos, ar ne. Šiame uždavinyje nagrinėjamas grafas yra neorientuotas (t. y., visos jo briaunos yra dvikryptės), nes gretimybių matrica yra simetrinė.

Gretimybių matrica gali būti naudojama kaip duomenų struktūra grafams vaizduoti kompiuterių programose, skirtose grafams tvarkyti. Pagrindinė alternatyvi duomenų struktūra, taip pat naudojama šiam tikslui, yra gretimybių sąrašas.

36. Geležinkelio bėgių tinklas

2024-PK-03

Bebrijoje kaimyninius miestelius jungia geležinkelio bėgių tinklas. Kiekvienu geležinkelio keliu per vieną dieną gali pravažiuoti ribotas skaičius traukinių. Šie skaičiai nurodyti paveikslėlyje. Skirtingi traukiniai gali naudoti tą patį kelią.



Planuojama vežti krovinius iš Vektorėnų į Pikselišės. Traukiniai gali važiuoti tik rodyklėmis nurodytomis kryptimis.

Kiek daugiausia traukinių gali nuvykti iš Vektorėnų į Pikselišės per vieną dieną?

Atsakymų variantai:

- A. 13
- B. 15
- C. 19
- D. 22

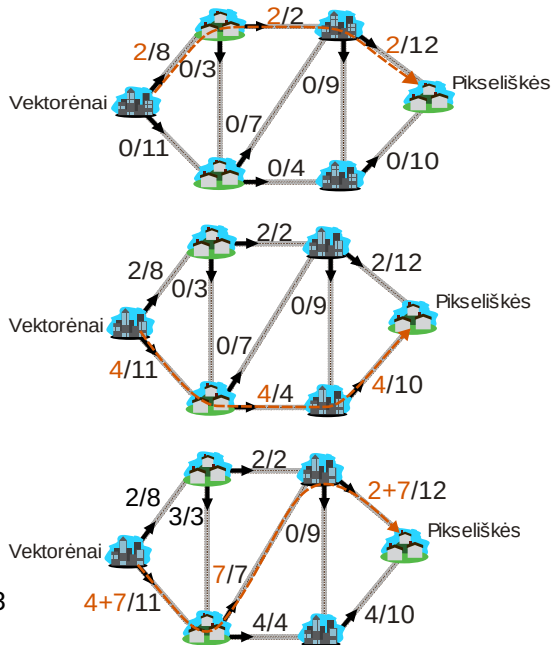
Teisingas atsakymas: A.

Paiškinimas

Pirmiausia pastebėjime, kad iš Vektorėnų (V) negali išvykti daugiau nei $8+11=19$ traukinių. Todėl atsakymas C (22) yra neteisingas. Vienas iš šios uždavinio sprendimo būdų – išskaidyti uždavinį į mažesnes dalis. Pirmiausia reikia rasti trumpiausią įmanomą kelią iš V į P (šiuo atveju 3 bėgių atkarpų ilgio) ir nustatyti didžiausią traukinių skaičių, kuris gali jomis vykti. Jis lygus 2.

Tada šį veiksmą pakartojame likusioje tinklo dalyje, nustatydami kiekviename kelyje likusį galimą traukinių skaičių. Pateiktose diagramose pavaizduoti dar du tokiu būdu rasti keliai. Taigi bendras visų trijų kelių pajėgumas yra $2+4+7=13$ traukinių.

Yra ir kitų būdų, kaip iš V į P nuvykti 13 traukinių, įskaitant kai kuriuos variantus, kai naudojami ilgesni keliai.



Norint įsitikinti, kad 13 iš tiesų yra didžiausias skaičius traukinių per vieną dieną, reikėtų atkreipti dėmesį į tris geležinkelio atkarpas tinklo vidurinėje dalyje. Šiais keliais gali vykti 2, 7 ir 4 traukiniai. Visi traukiniai, važiuodami iš Vektorėnų į Pikseliškės, turėtų vykti vienu iš šių kelių, todėl negalime gauti geresnio rezultato nei $2+7+4=13$.

Tai informatika!

Čia susidūrėme su vienu iš vadinamųjų didžiausio srauto uždavinių. Tokio tipo uždaviniai primena daug kam matytą galvosūkį, kuriame vanduo teka vamzdžiais: turime skirtingo storio vamzdžių tinklą ir norime išsiaiškinti, kiek vandens gali jais pratekėti.

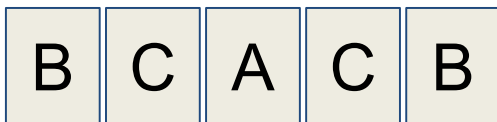
Spręsti didžiausio srauto uždavinius sukurta įvairių algoritmų: *Ford-Fulkerson*, *Edmonds-Karp*, *Dinic* ir daug kitų. Kiekvienas iš algoritmų turi savo ypatumų ir trūkumų, susijusių su vykdymo efektyvumu, įgyvendinimo paprastumu ir pritaikomumu skirtingų tipų tinklams. Algoritmo pasirinkimas priklauso nuo tinklo dydžio, pobūdžio, pageidaujamo optimizavimo lygio ir kitų veiksnių. Didžiausio srauto uždaviniai sutinkami ne vien galvosūkiuose – milžiniškus didžiausio srauto uždavinius sprendžia įvairių sričių inžinieriai ir mokslininkai. Didžiausio srauto algoritmai naudojami optimizuoti eismą (pavyzdžiui, reguliuojant šviesoforus), paskirstant gamybos išteklius, projektuojant kompiuterių tinklus, elektros ir vandens paskirstymo sistemas ir kitose srityse.

37. Raidžių seka

2024-SK-04

Sudėkite raides į seką ir surinkite kuo daugiau taškų. Jei dvi vienodos raidės yra greta, gaunate 2 taškus, jei trys – gaunate 3 taškus, ir taip toliau.

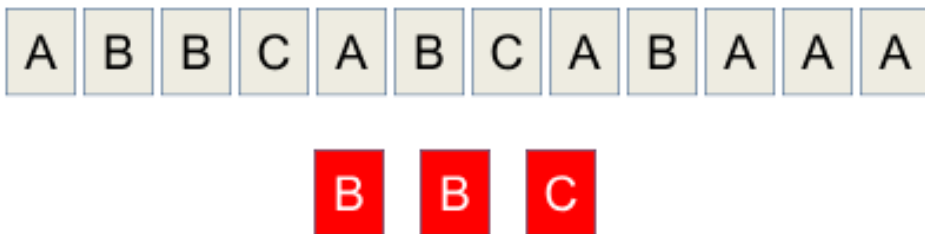
Pavyzdžiui, ši seka duotų 0 taškų, nes greta nėra vienodų raidžių:



Dabar pakeiskime pirmąją C kortelę į A, o antrąją C kortelę į B. Tokia seka pelnytų 4 taškus: 2 taškus už gretimas A ir 2 taškus už gretimas B raides.



Duota 12 raidžių seka. Jūs galite bet kurias tris raides pakeisti raidėmis B, B ir C. Nuvilkite raudonas korteles ir pakeiskite bet kurias tris raides taip, kad surinktumėte didžiausią galimą taškų skaičių.



Teisingas atsakymas:

A	B	B	C	C	B	B	B	B	A	A	A
---	---	---	---	---	---	---	---	---	---	---	---

Paaiškinimas

Šios kombinacijos bendras rezultatas yra $2 + 2 + 4 + 3 + 3 = 11$ taškų.

Daugiau taškų surinkti neįmanoma dėl šių priežasčių:

Daugiausiai būtų galima surinkti 12 taškų (jei visos kortelės priklausytų kuriai nors vienodų kortelių grupei).

Kairiausiai esanti A negali būti grupės dalis, nes nėra raudonų A kortelių, kuriomis būtų galima papildyti seką.

Papildžius seką raudonomis kortelėmis taip, kaip parodyta, kiekviena kita raidė (išskyrus kairiausią A) yra grupės dalis, todėl prisideda prie taškų skaičiavimo.

Grupės, kurioje yra raidė, dydis neturi jokios reikšmės – kiekviena bet kurioje grupėje esanti raidė duoda lygiai 1 tašką.

Ar tai vienintelis būdas surinkti 11 taškų? Norėdami surinkti kiek įmanoma daugiau taškų, turime taip padėti raudonas korteles B, B ir C, kad būtų suformuotos ar padidintos esamos vienodų raidžių grupės. Taip pat turime pasirūpinti, kad jau esamos grupės nebūtų išformuotos ar sumažintos. Padedant vieną kortelę galima gauti:

1 tašką, jei yra padidinama jau esanti grupė;

2 taškus, jei padėjus sukuriami nauja grupė iš dviejų kortelių arba prijungiamos dvi kortelės prie esamos grupės;

3 taškus, jei padėjus kortelę tarp dviejų vienodų kortelių sukuriami nauja grupė iš trijų kortelių.

Šios užduoties atveju iš pradžių turime 5 taškus. Naudojant duotas raudonas korteles matyti, kad neįmanoma viena kortele pelnyti 3 taškų. Taigi, norint surinkti 11 taškų, reikėtų su kiekviena raudona kortele pelnyti po 2 taškus, t. y., sukurti po naują dviejų raidžių grupę. Tam galime pasinaudoti tik 4 laisvomis pilkomis raidėmis: dviem C ir dviem B. Šią sąlygą tenkina tik vienas (jau pateiktas) variantas.

A	B	B	C	A	B	C	A	B	A	A	A
---	---	---	---	---	---	---	---	---	---	---	---

Be to, nesunku patikrinti visus variantus, kuriais raudonos kortelės duoda taškų nesumažindamos jau esamų grupių. Jų yra keturi:

B	B	B	C	C	B	B	A	B	A	A	A
B	B	B	C	B	B	C	C	B	A	A	A

A	B	B	B	B	B	C	C	B	A	A	A
A	B	B	C	C	B	B	B	B	A	A	

Ketvirtuoju variantu išties surenkame tiek pat taškų – 11. Kitais pavaizduotais atvejais surenkame po 10 taškų.

Tai informatika!

Šis pavyzdys iliustruoja, kaip mokslininkai ir programuotojai ieško geriausio sprendinio arba sprendinio paieškos metodo.

Jei nemėgintume ieškoti geriausio metodo, mums tektų patikrinti visus įmanomus raidžių sekos uždengimo raudonomis kortelėmis variantus. Jų būtų net $12 * 11 * 10 / 2 = 660$.

Žinoma, sprendimų skaičių galima gerokai sumažinti, jei atsižvelgsime į tai, kad raudonos kortelės su raidėmis B, B ir C duoda taškų tik tada, jei padedamos šalia tų pačių raidžių. Jei taip pat atsižvelgsime į tai, kad esamos grupės neturėtų būti mažinamos, sprendimų skaičius dar labiau sumažėja. Be to, atsižvelgus į tai, kad atskirų raidžių sujungimas duoda daugiau taškų nei esamų grupių didinimas, lieka tik keletas galimų variantų.

Kita vertus, kompiuterio greičiu patikrinti minėtus 660 variantų nėra sudėtinga. Tačiau, jei turėtume daug ilgesnę seką ir daug daugiau pakaitinių raidžių, net ir sprendžiant kompiuteriu būtų verta rasti greitesnį sprendimo būdą, nei patikrinti kiekvieną galimą atvejį. Informatikoje dažnai klausiama, kaip pasikeistų uždaviniui išspręsti reikalingi resursai didėjant uždavinio elementų skaičiui. Tokie uždaviniai kaip šis, kur reikia turimomis sąlygomis rasti geriausią variantą, vadinami optimizavimo uždaviniais. Pavyzdžiui, kai kalbame apie kodo optimizavimą kompiliatoriuje, turime omenyje kodo keitimo procesą, kad programa taptų efektyvesnė ir trumpesnė, bet nepasikeistų išvestis (programos darbo rezultatas) bei neatsirastų šalutinių poveikių. Tai svarbu, nes padeda programoms veikti greičiau ir naudoti mažiau išteklių.

Dar viena informatikos sritis, kurioje sprendžiami optimizavimo uždaviniai, yra duomenų glaudinimas, t. y., informacijos kodavimas naudojant mažiau bitų nei užima pirminė informacija. Tai atliekama siekiant sutaupyti vietos saugykloje arba sumažinti duomenų kiekį, kurį reikia perduoti.

38. Piešiantis robotas

2024-AU-03

Lino piešiantis robotas supranta ir vykdo dvi komandas:

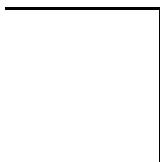
F – „pajudėk į priekį vieną atstumo vienetą“

R – „pasisuk į dešinę 90 laipsnių kampą“

Be to, bet kurią dar nenaudojamą raidę galima aprašyti priskiriant jai sudėtingesnes judėjimo sekas. Pavyzdžiui:

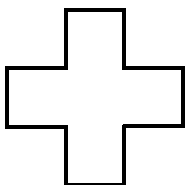
A = FFF reiškia, kad robotas turi pajudėti 3 atstumo vienetus į priekį;

B = ARA reiškia, kad pirmiausiai robotas įvykdo **A (FFF)**, tuomet pasisuka 90 laipsnių kampą į dešinę, o tuomet vėl vykdo **A**.



Robotui parašytoje programoje gali būti bet koks skaičius kodo eilučių, skirtų skirtingoms raidėms aprašyti. Paskutinė eilutė skirta komandai „Draw“, kurią vykdydamas robotas nupiešia galutinį brėžinį.

Linas sugalvojo nubraižyti čia pateiktą brėžinį, kuriame kiekviena trumpoji linijos atkarpa yra 1 atstumo vieneto ilgio.



Jo parašyta programa atrodė taip:

1 eilutė: **W = FR**

2 eilutė: **X = WRR**

3 eilutė: **Y = WXWX**

4 eilutė: **Draw YYYY**

Deja, Lino programoje įsivėlė rašybos klaida ir robotas braižo ne tokią figūrą. Vienoje kodo eilutėje yra vienas neteisingas simbolis.

Kurią kodo eilutę reikia pataisyti?

Atsakymų variantai:

A. 1 eilutę: **W = FR**

B. 2 eilutę: **X = WRR**

C. 3 eilutę: **Y = WXWX**

D. 4 eilutę: **Draw YYYY**

Teisingas atsakymas: C.

Paaiškinimas

Tarkime, kad robotas iš pradžių yra nukreiptas į dešinę. Įvykdykime kiekvieną kodo eilutę. Vykdydamas pirmą eilutę robotas juda vienu atstumo vienetu į priekį ir pasuka į dešinę.

1 eilutė: $W = FR$



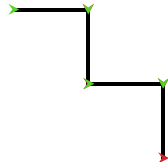
Vykdydamas antrą eilutę robotas pajuda vieną atstumo vienetą į priekį ir tris kartus pasisuka į dešinę po 90 laipsnių. Vadinasi, jis pasisuka į kairę 90 laipsnių kampu.

2 eilutė: $X = WRR$



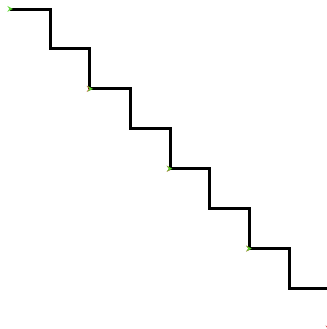
Trečioje eilutėje du kartus vykdomos dvi ankstesnės eilutės:

3 eilutė: $Y = WXWX$



4 eilutėje ankstesnė eilutė kartojama keturis kartus. Taip gaunamas visiškai kitoks brėžinys, nei norėjo Linas.

4 eilutė: Draw YYYY



Tiksliniame brėžinyje yra 12 vieno vieneto ilgio atkarpų. 4 eilutėje ankstesnė eilutė kartojama 4 kartus, o tai reiškia, kad vienas iš šių pakartojimų nubrėžia $\frac{1}{4}$ tikslinio brėžinio. $\frac{1}{4}$ brėžinio yra trys atkarpos, o tai reiškia, kad klaida yra 3 eilutėje, kuri nubrėžia keturias brėžinio atkarpas.

Galima pamanyti, kad kodą taip pat galima pataisyti pakeitus 1-ą eilutę į $W = F$, tačiau taip nėra. Jei pakeisite šią eilutę, tai paveiks ir kodą, skirtą Y 3 eilutėje, o per jį pasikeis ir Z 4 eilutėje, todėl galiausiai bus gautas kitoks brėžinys.

Tai informatika!

Šioje užduotyje pateikiama nauja programavimo kalba. Šia kalba komandos vykdomos iš eilės, taip pat galima apibrėžti naujas komandas – procedūras. Procedūrų arba funkcijų naudojimas programavime yra įprastas dalykas, dėl jų programa tampa lengviau skaitoma. Šiuo atveju programą skaityti yra sunkiau dėl naujoms procedūroms suteiktų neprasmingų pavadinimų.

Pirmasis šios programos derinimo žingsnis – išsiaiškinti, ką ji daro: turite pabandyti patys pabūti kompiuteriu ir vykdyti programą savo galvoje arba popieriaus lape. Kadangi žinote, kad buvo įvelta tik viena klaida, turite ieškoti, kurią vieną eilutę būtų galima pakeisti, kad programa elgtųsi taip, kaip norite.

39. Saulėtos dienos

2024-DE-04b

Tomas sako: „Saulėtomis dienomis kiekviename tvenkinyje maudosi bent vienas bebras“. Kazys atsako: „Tai netiesa. Pavyzdžiui, praėjusį sekmadienį taip nebuvo.“



Sakykite, kad Kazys yra teišus. Sudarykite tokį teiginį, kuris paneigtų Tomo teiginį: Praėjusį sekmadienį _____ ir _____.

Atsakymų variantai:

Į pirmąją tuščią vietą parinkite vieną iš šių variantų:

- buvo saulėta
- nebuvo saulėta

Į antrąją tuščią vietą parinkite vieną iš šių 4 variantų:

- visuose tvenkiniuose maudėsi bebrai
- tvenkinyje su kriekliu bebrai nesimaudė
- bebras Mykolas maudėsi visuose tvenkiniuose
- bebras Mykolas visai nesimaudė

Teisingas atsakymas: buvo saulėta ir tvenkinyje su kriekliu bebrai nesimaudė.

Paaiškinimas

Pirmajai tuščiai vietai tinkamas variantas yra „buvo saulėta“.

Tomo teiginys turėtų būti teisingas visoms saulėtoms dienoms. Tomo teiginiui paneigti reikia rasti saulėtą dieną, kai jo teiginys nėra teisingas. Kadangi Tomas nekalba apie nesaulėtas dienas, jam negalima ir paprieštarauti teiginiu apie tokias dienas.

Antrajai tuščiai vietai teisingas variantas yra „tvenkinyje su kriekliu bebrai nesimaudė“.

Teiginys, kad tą saulėtą dieną joks bebras nesimaudė tvenkinyje su kriekliu, paneigia Tomo teiginį, kad visuose tvenkiniuose buvo besimaudančių bebrų.

Kiti atsakymo variantai Tomo teiginio nepaneigia.

„Praėjusį sekmadienį buvo saulėta ir visuose tvenkiniuose maudėsi bebrai“. Čia sakoma, kad kiekviename tvenkinyje maudėsi bebrai – tą patį teigia ir Tomas.

„Praėjusį sekmadienį buvo saulėta ir bebras Mykolas maudėsi visuose tvenkiniuose“. Pagal šį teiginį, kiekviename tvenkinyje maudėsi bent Mykolas, t. y. bent vienas bebras. Tai neprieštarauja Tomo teiginiui.

„Praėjusį sekmadienį buvo saulėta ir bebras Mykolas visai nesimaudė“. Šito nepakanka Tomo teiginiui paneigti. Kiekviename tvenkinyje galėjo maudytis kiti bebrai, o tai neprieštarauja tam, ką sako Tomas.

Tai informatika!

Kazys suabejoja Tomo teiginio teisingumu. Teiginius ir jų teisingumą nagrinėja teiginių logika. Teiginiai gali būti arba tiesa, arba netiesa. Tomo teiginys yra apie visas saulėtas dienas. Šis teiginys gali būti netiesa tik tuo atveju, jei atsiras bent viena saulėta diena, kurią bent viename tvenkinyje nesimaudys nė vienas bebras. Taigi, užtenka vieno prieštaraujančio pavyzdžio, kad teiginys būtų paneigtas.

Naudojant loginius operatorius IR, ARBA, NE, taip pat kiekinius žodžius KIEKVIENAS, BENT VIENAS galima dalinius teiginius sujungti į sudėtingus teiginius ir į juos įtraukti daugiau elementų.

Logika yra kompiuterių darbo principų pagrindas. Net ir patys kompiuteriai, galima sakyti, yra sudaryti iš loginių elementų, tokių kaip IR, ARBA ar NE. Tik šie elementai vietoj „tiesos“ ir „netiesos“ naudoja reikšmes 1 ir 0 (elektros impulsas yra arba ne). O programavimo kalbose, kuriomis rašomos komandos kompiuteriui, naudojami loginiai teiginiai, pavyzdžiui, „JEI dabar vasara IR šviečia saulė, TAI įjunk kondicionierius“.

Tai informatinis mąstymas:

Kadangi kompiuterių veikimas pagrįstas logika, dirbant kompiuteriu su logika susidurti tenka ypač dažnai. Pavyzdžiui, programuotojas realaus pasaulio situaciją turi aprašyti loginiais teiginiais, kad tinkamai pateiktų komandas kompiuteriui. Arba gali tekti nustatyti tikslų priešingą sąlygą duotajai, kad rašoma programa galėtų apdoroti visus įmanomus atvejus. Loginis mąstymas reikalingas duomenims analizuoti ir interpretuoti, o kartu yra ir vienas iš informatinio mąstymo elementų.

Taip pat galima teigti, kad šiai užduočiai išspręsti reikalingi ir dekomponavimo įgūdžiai, mat mokiniai kiekvieną teiginį turi naginėti atskirai.

Galiausiai mokiniai turi įvertinti, ar jų sudarytas teiginys yra teisingas, ar ne.

40. Bebrų karaliaus lobis

2019-VN-04

Bebrų karalius valdo septynias provincijas, kurių kontūrai pavaizduoti žemėlapyje. Vienoje iš provincijų karalius paslėpė savo lobį:



Karalius liepė pagaminti žemėlapij, kuriame provincijos būtų pažymėtos apskritimais. Žemėlapyje jis pažymėjo provinciją su lobiu. Du apskritimai yra sujungti tada ir tik tada, kai atitinkamos provincijos turi bendrą sieną. Kad sulaikytų plėšikus, karalius taip pat liepė pagaminti keturis netikrus lobių žemėlapius. Kuris lobių žemėlapis yra teisingas?

Atsakymų variantai:

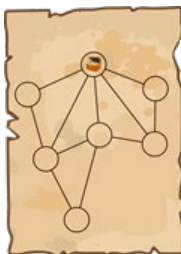
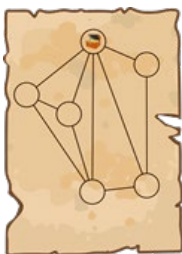
A

B

C

D

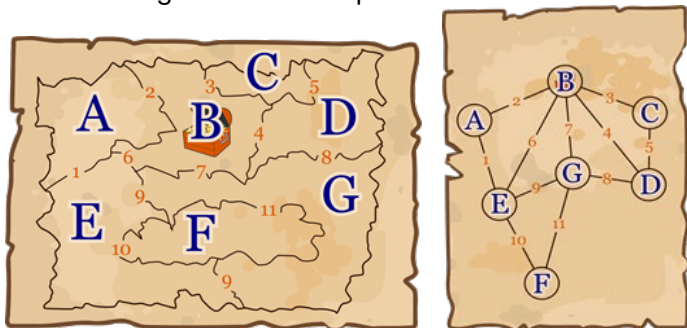
E



Teisingas atsakymas yra D.

Paaiškinimas

Paveikslėliuose provincijos pažymėtos raidėmis A, B, C, D, E, F ir G. Sienos tarp provincijų pažymėtos skaičiais nuo 1 iki 11. Nors tarp E ir G provincijų į bendrą sieną yra įsiterpusi F provincija, pagal sąlygą reikalaujama tik bendros sienos, todėl abi bendros sienos dalys pažymėtos skaičiumi 9. Naudojantis šiais žymėjimais įrodoma, kad tai teisingas lobio žemėlapis.



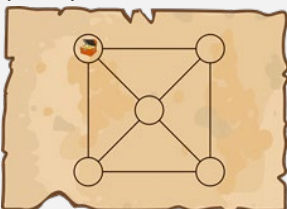
Atsakymas A yra klaidingas: trys provincijos A, C ir F turi tik po dvi kaimynes. Dėl to lobio žemėlapyje turėtų būti trys viršūnės (apskritimai) kiekviena jų turinčios tik po dvi briaunas. Tačiau A atveju tik apačioje esanti vienintelė viršūnė yra vaizduojama su dviem briaunomis.

Atsakymas B yra klaidingas, nes šio atvejo žemėlapyje yra tik šešios viršūnės. Tačiau karalius valdo septynias provincijas.

Atsakymai C ir E yra klaidingi, nes provincija su paslėptu lobiu turi penkias kaimynes, o C ir E atveju yra tik keturios briaunos – tik keturios kaimynės.

Tai informatika!

Ši užduotis žemėlapią atvaizduoja grafą. Toks grafas yra žemėlapijo abstrakcija (kaip ir žemėlapis yra tikrovės abstrakcija). Kaip ir kiekvienoje abstrakcijoje, prarandama nereikšminga informacija. Šiuo atveju tai yra geografinė provincijų padėtis viena kitos atžvilgiu: nors žemėlapyje provincijos B ir D yra tame pačiame aukštyje, grafe jos yra skirtinguose aukščiuose. Tačiau svarbi informacija vis tiek išlieka. Šiuo atveju tai atsakymas į klausimą, kurios provincijos yra viena kitos kaimynės. Tačiau tinkama abstrakcija kelia pavojų prarasti iš pirmo žvilgsnio nereikšmingą informaciją. Pateiktame pavyzdyje grafas yra simetriškas posūkiui, todėl vis tiek nebūtų aišku, kur galėtų būti paslėptas lobis:



Grafą sudaro viršūnės (šioje užduotyje apskritimai) ir briaunos (šioje užduotyje – atkarpos). Šioje užduotyje pavaizduotos grafo briaunos jungia tik dvi viršūnes. Taip pat neįmanoma sujungti dviejų viršūnių keliomis briaunomis, nors toks sujungimas būtų įmanomas provincijoms E ir G.

Informatikoje grafai dažnai naudojami informacijai abstrahuoti. Daugeliu atvejų ši abstrakcija jau yra uždavinio sprendimas. Kadangi grafai yra gerai ištirti, dažnai pakanka uždavinį redukuoti į tipinį grafo uždavinį ir naudoti jau žinomus sprendimo metodus.

4.1. Paveikslų šifravimas

2024-AU-01

Leonas sukūrė naują vaizdų šifravimo metodą, kuriame naudojamos dvi operacijos: H (horizontalioji) ir V (vertikalioji).

Šifruojamas vaizdas (paveikslas) yra stačiakampis, sudarytas iš spalvotų taškų, vadinamų pikseliais, išsidėsčiusių eilutėse ir stulpeliuose.

Atliekant operaciją H:

- kiekvienas pirmosios eilutės pikselis lieka toje pačioje padėtyje.
- kiekvienas antrosios eilutės pikselis pasislenka vienu stulpeliu į dešinę.
- kiekvienas trečiosios eilutės pikselis pasislenka dviem stulpeliais į dešinę.
- kiekvienas n -tosios eilutės pikselis pasislenka $n-1$ stulpeliu į dešinę.

Pikseliai, atsidūrę už paveikslo dešiniojo krašto, įterpiami į tą pačią eilutę kairiajame paveikslo krašte, nekeičiant jų eiliškumo.

Panašiai atliekama ir operacija V:

- kiekvienas n -ojo stulpelio pikselis pasislenka žemyn $n-1$ eilute. Pikseliai, kurie atsidūrė už apatinio paveikslo krašto, įterpiami į to paties stulpelio viršų.

Pavyzdys su 3×3 paveikslu, kuriame pikseliai sužymėti nuo 1 iki 9:

3	4	5
6	7	8
9	:	;

J →

3	4	5
8	6	7
:	;	9

X →

3	;	7
8	4	9
:	6	5

Čia 25×25 dydžio Monos Lizos paveikslas užšifruojamas HVHV seka:



Leonas užšifruoja tokį 1000×1000 dydžio paveikslą seka VH:



Kuriame paveikslėlyje pavaizduotas Leono užšifruotas vaizdas?

Atsakymų variantai:

A



B



C



D



E



Teisingas atsakymas: E.

Paiškinimas

Pavyzdyje pateiktas Monos Lizos paveikslo iškraipymas rodo, kad H operacija geometriškai apima horizontalią šlytį dešinėn ir kirpimą, o V operacija – vertikalią šlytį žemyn ir kirpimą.

Atvaizdas su keturiais spalvotais kvadratais pirmiausia nutempiamas žemyn (po pikselį kas stulpelį) atliekant operaciją V. Gaunamas lygiagretainis su iškreiptu vaizdu. Apačioje atsidūręs trikampis nutempiamas į viršų (lygiagrečiu poslinkiu), kad vėl būtų sukurtas stačiakampis atvaizdas. Naudojant operaciją H atvaizdas nutempiamas dešinėn (po pikselį kas eilutę) ir nukirptas iš dešinės trikampis nutempimas lygiagrečiai kairėn.



Tai informatika!

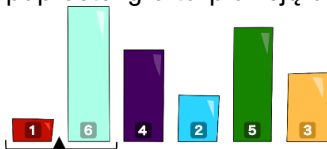
Saugiai perduodant informaciją svarbu užšifruoti duomenis, kad bet kas, perėmęs informaciją, negalėtų lengvai nustatyti pirminių duomenų. Vaizdų šifravimo metodo, kuris pagrįstas operacijomis H ir V, privalumas yra tas, kad jį galima taikyti paprastai ir lengvai. Jei gavėjas žino seką (pvz., HV), jis gali lengvai iššifruoti. Tačiau užšifruoto vaizdo nebegalima atpažinti iš akies. Kuo didesni šifravimo saugumo reikalavimai, tuo daugiau pastangų paprastai reikia užšifravimui ir iššifravimui (pvz., sudėtingo slaptojo rakto perdavimui tarp užšifruotos informacijos siuntėjo ir gavėjo).

Šią procedūrą lengva įgyvendinti, tačiau ją taip pat gali būti lengva „nulaužti“. Šio šifravimo ypatybė yra ta, kad jis grindžiamas geometriniais vaizdais.

42. Statybinių blokelių mašina

2024-BE-01b

Mašina gali perkelti šešis skirtingo aukščio blokelių. Tam ji naudoja žymeklį, kuris paprastai yra tarp dviejų blokelių. Pradžioje blokeliai ir žymeklis išdėstyti taip:

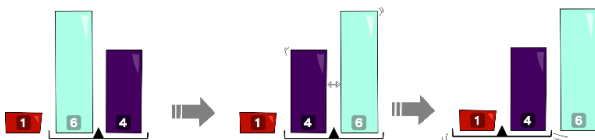


Mašina veikia pagal šias taisykles:

Jei į kairę nuo žymeklio esantis blokelis yra žemesnis už blokelį žymeklio dešinėje, žymeklis pasislenka į dešinę.



Jei į kairę nuo žymeklio esantis blokelis yra aukštesnis už blokelį žymeklio dešinėje, blokeliai sukeičiami vietomis. Atliekant šį veiksmą žymeklis pasislenka į kairę tik tuo atveju, jei jis vis dar yra tarp dviejų blokelių.

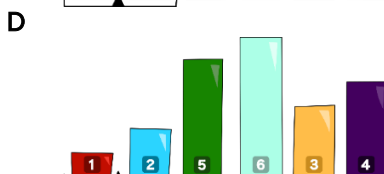
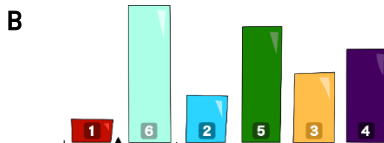
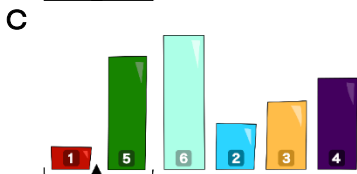
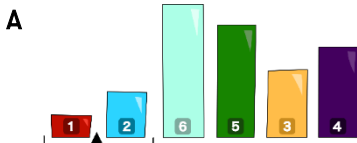


Mašina sustoja, kai žymeklis atsiduria šalia blokelių dešinėje pusėje.

Priklausomai nuo pradinio blokelių išsidėstymo, mašina atlieka skirtingą skaičių veiksmų.

Kurio pradinio blokelių išsidėstymo atveju mašina atlieka mažiausiai veiksmų?

Atsakymų variantai:



Teisingas atsakymas: D.

Paiškinimas

Norint nustatyti teisingą atsakymą, galima imituoti mašinos darbą, kol ji sustos, ir kiekvienam atvejui suskaičiuoti žymeklio judesius.

Lengviau palyginti keturių pradinėjų situacijų žymeklio judesių skaičių jų neskaičiuojant.

Kad būtų paprasčiau, šios pradinės situacijos vaizduojamos tik skaičiais (blokelių aukščiau), o žymeklis yra tarp dviejų skaičių. Taip pavaizduotos kiekvienam atvejui pradinės situacijos atrodo taip:

A	B	C	D
1 ▲ 2 6 5 3 4	1 ▲ 6 2 5 3 4	1 ▲ 5 6 2 3 4	1 ▲ 2 5 6 3 4

Jei mašina pradeda nuo B atvejo, tai A konfigūracija pasiekama po dviejų veiksmų:
 $1 \blacktriangle 6 2 5 3 4 \rightarrow 1 6 \blacktriangle 2 5 3 4 \rightarrow 1 \blacktriangle 2 6 5 3 4$

Todėl B atveju mašina atlieka daugiau veiksmų nei A atveju ir atsakymas B negali būti teisingas.

Jei mašina pradeda nuo A atvejo, ji pirmiausia atlieka šiuos tris veiksmus:

$1 \blacktriangle 2 6 5 3 4 \rightarrow 1 2 \blacktriangle 6 5 3 4 \rightarrow 1 2 6 \blacktriangle 5 3 4 \rightarrow 1 2 \blacktriangle 5 6 3 4$

Tokia konfigūracija $1 2 \blacktriangle 5 6 3 4$ iš D konfigūracijos gaunama vienu veiksmu:

$1 \blacktriangle 2 5 6 3 4 \rightarrow 1 2 \blacktriangle 5 6 3 4$.

Todėl A atveju mašina atlieka daugiau veiksmų nei D atveju, o tai reiškia, kad atsakymas A negali būti teisingas.

Jei mašina pradeda nuo C, D atvejis pasiekiamas po keturių veiksmų.

$1 \blacktriangle 5 6 2 3 4 \rightarrow 1 5 \blacktriangle 6 2 3 4 \rightarrow 1 5 6 \blacktriangle 2 3 4 \rightarrow 1 5 \blacktriangle 2 6 3 4 \rightarrow 1 \blacktriangle 2 5 6 3 4$

Todėl C atveju mašina taip pat atlieka daugiau veiksmų (žymeklio judesių) nei D atveju.

Išrodėme, kad mažiausiai veiksmų mašina atlieka D atveju.

Tai informatika!

Duomenų rikiavimas yra įprastas procesas informatikoje – rikiuojama pagal dydį, kainą ar abėcėlę. Todėl informatikos mokslininkai ieško kuo efektyvesnių rikiavimo metodų.

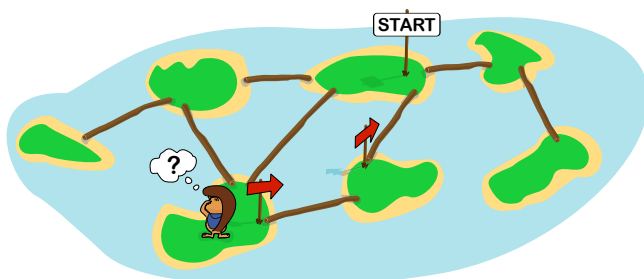
Šioje užduotyje naudojamas metodas vadinamas Gnomu rikiavimu. Jis panašus į rikiavimą didėjimo tvarka (Burbulo rikiavimo algoritmas). Jis nėra itin efektyvus – gerokai prastesnis, negu efektyvūs rikiavimo metodai, pavyzdžiui, greito rikiavimo algoritmas ([angl. quicksort](#)) ar sąlaivos rikiavimas ([angl. mergesort](#)).

Kita vertus, Gnomu rikiavimas yra lengviau suprantamas.

4.3. Tyrinėjimas

2024-DE-05

Haris lanko salyną, kurio salos sujungtos medžių kamienais. Jis nori bent kartą aplankyti kiekvieną salą. Stovėdamas saloje jis gali matyti visas kaimynines salas ir nustatyti, ar jose yra ženklas.



Paveiksle yra situacija, kai jis jau įkėlė koją į tris salas. Haris juda pagal šias instrukcijas:

Bet kurioje saloje pastato pradžios ženklą.

Kiekvieną kartą, kai atvyksta į salą, atlieka tokius veiksmus:

Jei ši sala jau turi ženklą:

palikite ženklą tokį, koks jis yra.

Kitu atveju:

pastatykite rodyklės ženklą, rodantį į salą, iš kurios ką tik atvykote.

Jei yra kaimyninė sala, kurioje , tada:

Eikite į tą salą.

Kitu atveju:

pažiūrėkite į ženklą saloje, kurioje esate.

Jei tai yra rodyklės ženklas, tada:

eikite į salą .

Kitu atveju:

nuėjote į visas tinklo salas, kelionė baigėsi.

Tuščioms vietoms parinkite tinkamus teksto fragmentus, kad algoritmas būtų teisingas.

Atsakymų variantai:

nėra jokio ženklo

yra rodyklė

yra pradžios ženklas

iš kur ką tik atėjote

kur rodo rodyklė

Teisingas atsakymas:

něra jokio ženkle

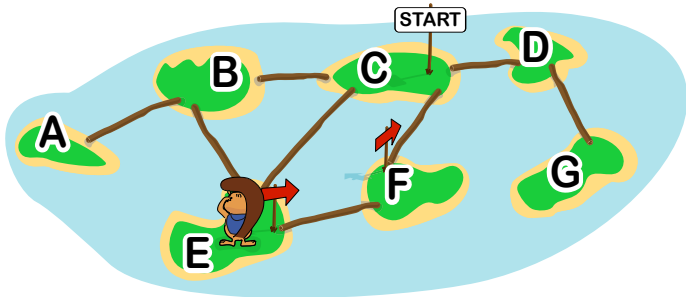
kur rodo rodyklę

Paaiškinimas

Pirmiausia suvokime, kad Haris uždeda ženklą, kai įeina į salą be ženklų. Todėl visose salose, į kurias Haris jau įžengė, yra ženklas, o visose salose, į kurias jis dar nepateko, dar nėra ženklų.

Todėl teksto fragmentas (a) yra teisingas: **nėra jokio ženklo**. Tai ivers jį, jei įmanoma, patekti į salą, kurioje dar nebuvo.

Štai paveikslėlyje jis turi vykti į B salą, nes tai vienintelė kaimyninė sala be ženklo.



Teksto fragmentas (b) **yra rodyklė** neteisingas. Esant šiam tekstui Haris patektų tik į salas, kuriose jis jau buvo.

Teksto fragmentas (c) **yra pradžios ženklas** taip pat neteisingas. Atsitiktų taip: Hariui įžengus į antrąją salą, jis grįžtų į salą su pradžios ženklu ir ten sustotų. Tačiau jis dar neaplankė visų salų ir savo užduoties neįvykdė.

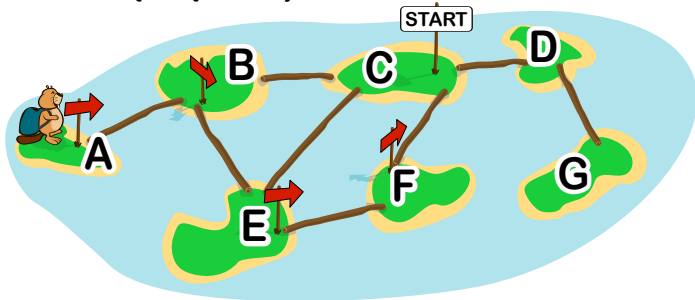
Antrojo teksto (žaliai vietai) parinkimas

Svarbu suvokti: Haris naudoja rodykles, kad atsektų vietą, kur pradėjo paiešką.

Teksto fragmentas (b) **kur rodo rodyklė** teisingas. Tai užtikrina, kad Haris sektų savo ženklų pėdsakais, kol atvyks į salą, kuri dar nebuvo aplankyta. Tai vadinama grįžimo (angl. *backtracking*) metodu.

Teksto fragmentas (a) **iš kurio ką tik atėjai** neteisingas. Jam esant kad Haris amžinai keliautų pirmyn ir atgal tarp dviejų salų, neaplangydamas jokių kitų kaimyninių salų. Pavyzdžiui, Haris yra A saloje, kuri neturi gretimos salos be ženklo. Naudodamas teksto fragmentą (a) kitu žingsniu jis eity į B, tada atgal į A, tada vėl į B ir t. t.

Naudodami teisingą teksto fragmentą iš salos A sekite rodyklėmis atgal į salą C ir tada eisite į salą D, nes ji dar neturi ženklo.



Tai informatika!

Sujungtų salų tinklą galima modeliuoti grafu. Grafas yra duomenų struktūra. Jį sudaro mazgai (šioje užduotyje salos) ir briaunos (čia – medžių kamienai), jungiančios mazgus. Daugelį realių sistemų galima pavaizduoti grafais.

Pavyzdžiui, žmonės sieja draugystę, traukinių stotis jungia geležinkelio linijos, galimos pozicijos šachmatų partijoje yra sujungtos galiojančiais ėjimais. Kartais svarbu išnagrinėti visą grafą iki galo, pavyzdžiui, ieškant mazgo su tam tikromis savybėmis. Ypač dideliems grafams su daug mazgų ir briaunų reikia strategijos, užtikrinančios, kad tikrai apžvelgsite kiekvieną mazgą ir nepraleisite nė vieno mazgo.

Šioje užduotyje naudojama strategija kompiuterių moksle vadinama gilumine paieška. Visiškai įvykdęs algoritmą bebras grįžta į salą, kurioje jis pradėjo savo kelionę.

Algoritmas, kurį reikėjo atlikti, buvo užrašytas pseudokodu. Pseudokodas yra natūralios kalbos ir programavimo kalbos susitarimų mišinys. Nurodymai suformuluoti taip, kad būtų iš esmės suprantami. Įtrauka nuo eilutės pradžios (kaip ir programavimo kalboje *Python*) parodo, kurie teiginiai priklauso tai pačiai grupei.

4. Plytų tvora

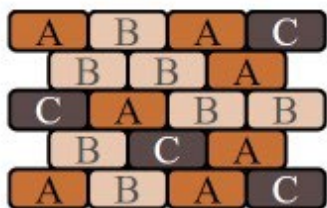
2024-FI-03

Benas sudėjo tvorą naudodamas trijų rūšių plytas: A, B ir C. Dabar jis turi perstatyti tvorą kitoje vietoje. Benas ima ir deda plytą po plytos, kol perstato tvorą taikydamas tokią taisyklę:

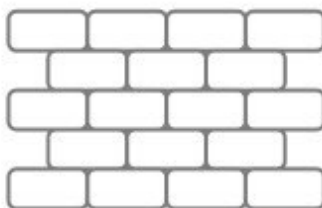
- ima iš ardamos tvoros bet kurią plytą, ant kurios nėra kitos plytos, ir
- deda ją arba ant grindinio arba ant vienos ar dviejų plytų naujoje tvoroje.

Benas laikosi šios taisyklės ir jam nesvarbu, kokia tvarka skirtingų rūšių plytos perkeliamos į naująją tvorą.

Kairėje matote pradinę Beno tvorą. Dešinėje numatyta vieta naujai tvorai. Galite išbandyti, kaip Benas perkrauna plytas. Vilkite plytas iš kairės į dešinę į naujos tvoros vietą. Deja, netikrinama, ar perkraunate plytas pagal Beno taisyklę.



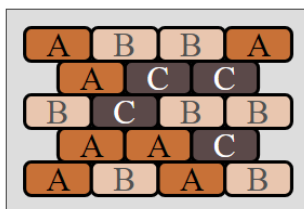
Pradinė tvora



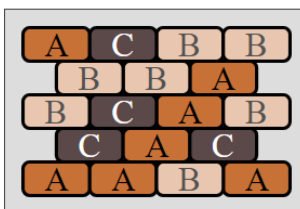
Nauja tvora

Kaip galėtų atrodyti naujoji Beno tvora, kai ji bus baigta? Pažymėkite visus tinkamus atvejus.

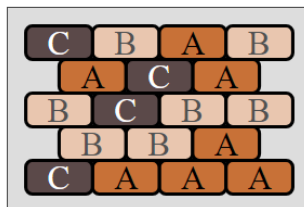
A



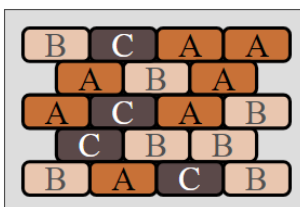
B



C



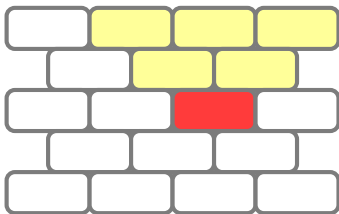
D



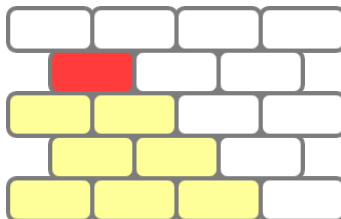
Paiškinimas

Teisingi atsakymai yra A ir C, nes tik juose pavaizduotas tvoras galima pastatyti iš pradinės tvoros laikantis duotos taisyklės.

Pradėkime nuo pagrindinės aplinkybės. Kairiajame paveikslėlyje raudona plyta yra apversto geltono trikampio viršūnė. Būtent, šias geltonas plytas reikia sudėti iš naujo prieš dedant raudonąją plytą. Kiekviena pirmosios krūvos plyta turi „anksčiau pašalintą“ trikampį. Dešinėje pusėje esančiame paveikslėlyje geltonos plytos yra tos, kurias reikia sudėti į naują krūvą prieš raudonąją plytą, tai yra trikampis „prieš“, kuris egzistuoja kiekvienai plytai.



Pradinė tvora



Nauja tvora

Jei plytos sunumeruotos ta tvarka, kuria jos perkraunamos, taikomos šios dvi perrinkimo sąlygos:

1. Kiekviena pirmosios tvoros plyta turi didesnį numerį nei kitos plytos, esančios „anksčiau išjungtame“ trikampyje;
2. Taip pat kiekviena naujos tvoros plyta turi didesnį numerį nei kitos plytos, esančios jos trikampyje „anksčiau įjungta“.
- 3.

Atvejis A

Pradinė

3	4	5	7
6	9	8	
;	:	32	33
37	34	35	
38	39	3:	36

Nauja

38	37	39	3:
35	34	36	
33	;	9	32
:	8	7	
3	4	5	6

Atvejis C

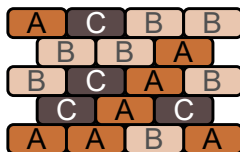
3	5	4	6
7	8	9	
;	:	32	33
38	34	35	
39	3:	37	36

36	38	39	3:
35	34	37	
8	;	32	33
7	5	:	
6	3	4	9

Įrodykime, kad atvejis B negalimas. Pirmiausia atvejo B tvoroje bet kurią ketvirtą B plytą, kurią ketinama paimti iš pradinės tvoros pavadinkime „4-oja B plyta“. Norint paimti bet kurią ketvirtąją plytą B, galima iš pradinės tvoros išimti daugiausia 4 plytas A ir 2 plytas C, esančias viršutinėse trijose eilutėse. Dabar išimkime ketvirtąją B plytą ir įdėkime ją į sieną B. Matome, kad į naująją sieną jau perkeltos bent 5 A plytos arba bent 3 C plytos, o tai prieštarauja ankstesniam pastebėjimui. Taigi siena B neįmanoma.

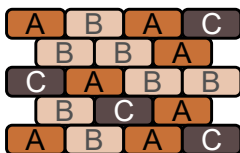


Atvejis B

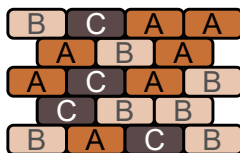


Panagrinėkime tvorą atvejo D. Tarkime, kad nuo pradinės tvoros nuimama trečioji plyta B. Tuomet jau turi būti pašalintos bent 2 plytos A. Antroji plyta A turi būti padėta arba kairiajame D tvoros vidurinės eilės krašte, arba trečia nuo šios eilės kairiojo krašto. Tačiau tam reikia, kad jau būtų padėta mažiausiai trečioji plyta B. Tarkime, kad 3-ioji plyta A būtų tik padėta. Tai gali būti arba antroji A plyta trečioje eilėje, arba kairiausia plyta aukščiau esančioje eilėje, bet tam reikia, kad nuo pradinės sienos jau būtų perkeltos 3 plytos C, o tai negali įvykti, kol dar nėra pašalintos 4 plytos A. Taigi 2-oji ir 3-ioji A plytos dedamos į vidurinę eilę, todėl dviejose apatinėse eilėse jau turėjo būti 4 plytos B. Tačiau 4 plytos B gali būti pašalintos iš pradinės tvoros tik po to, kai buvo pašalintos bent 3 plytos A.

Šis prieštaravimas reiškia, kad atvejo D tvora neįmanoma.



Atvejis D



Tai informatika!

Užduotis susijusi su priklausomybių (arba pirmenybės apribojimų) tarp objektų analize. Tai klasikinis informatikos uždavinys. Priklausomybės tarp objektų – šiame uždavinyje – plytų paprastai išreiškiamos priklausomybių grafo pavidalu. Teisėta apdorojimo tvarka šioje užduotyje – tai tvarka, kuria plytos gali būti išimamos ir dedamos – atitinka topologinę objektų tvarką.

45. Palago

2024-HU-04

Palago yra dviejų žaidėjų žaidimas, žaidžiamas su identiškais šešiakampiais. Šešiakampyje yra mėlynos ir baltos spalvos dalys. Kiekvienas žaidėjas pasirenka spalvą: baltą ar mėlyną. Žaidimo tikslas yra sudėti uždarytą savo spalvos figūrą.



Pirmasis žaidėjas padeda du šešiakampius vieną šalia kito taip, kad besiliečiančių briaunų spalvos sutaptų. Toliau kiekvienas žaidėjas vėl deda po du šešiakampius laikydamasis šių taisyklių:

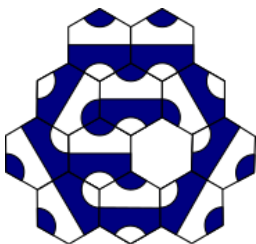
1. bent vienas dedamas šešiakampis turi būti prijungtas prie jau padėtų šešiakampių;
2. abu padėti šešiakampiai turi liesti vienas kitą briauna;
3. susilietusių šešiakampių spalvos turi sutapti.

Žaidėjas laimi, jei sudeda savo spalvos uždarytą figūrą, kurioje yra bent viena atkarpa. Tačiau jis pralaimi, jei tame pačiame ėjime sudeda ir vienos, ir kitos spalvos tokią uždarytą figūrą. Jeigu uždaryta figūra sukurti reikia tik vieno šešiakampio, antro žaidėjas gali nebedėti.



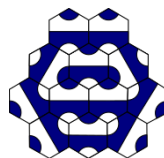
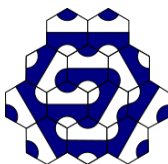
Šiame paveiksle sudėta mėlyna uždaryta figūra, tačiau joje nėra nė vienos atkarpos, todėl laimėjimo nėra. Iš baltos spalvos dar nėra sudėtos jokios uždarytos figūros, todėl irgi laimėjimo nėra.

Esama žaidimo situacija:

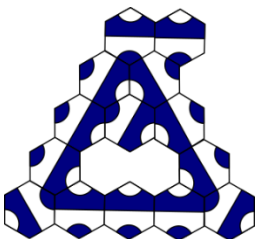


Šešiakampį įdėjo baltasis žaidėjas:

1. Baltasis laimi, nes suformavo uždarytą baltą figūrą, kurioje tikrai yra atkarpa.
2. Baltasis pralaimi, nes suformavo ne tik baltą figūrą su atkarpa, bet ir mėlyną.



Kas laimės šioje žaidimo situacijoje, jei abu žaidėjai žais optimaliai, kad laimėtų?

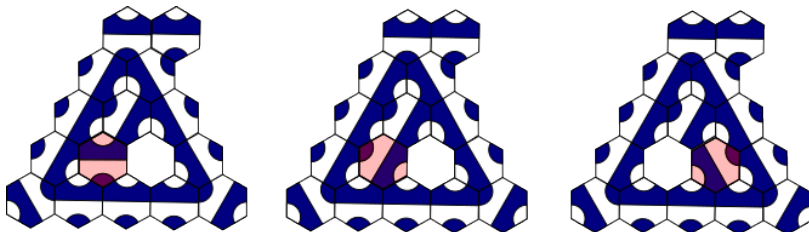
**Atsakymų variantai:**

- A) Laimės mėlynasis.
- B) Laimės baltasis.
- C) Neįmanoma atspėti.
- D) Tai priklauso nuo to, kurio dabar ėjimas.

Teisingas atsakymas: D.

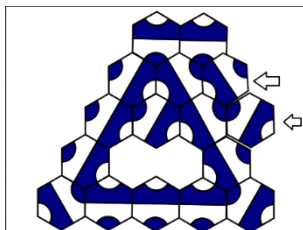
Paaiškinimas

Jeigu kitas ėjimas baltojo, jis gali užbaigti baltą figūrą padėdamas tik vieną šešiakampį. Tai jis gali padaryti trimis būdais:



Pagal taisyklės antrojo šešiakampio jam dėti nebereikia.

Jeigu ėjimas yra mėlynojo, jis gali laimėti padėdamas du šešiakampius ir jo sudarytoje uždaroje figūroje bus viena briauna:



Tai informatika!

Tokiame daugelio žaidėjų strateginiame žaidime kaip šis („Palago“) reikia strategijos, kad laimėtumėte arba išvengtumėte pralaimėjimo. Svarbu matyti kelis žingsnius į priekį ir pagal tai priimti sprendimą.

Daugumoje strateginių žaidimų prognozuoti ėjimus galima naudojant žaidimo medį. Žaidimo medžio šaknis yra pradinė pozicija, o visi galimi ėjimai iš kiekvienos pozicijos yra šakos. Taip sudaromas medis, kuris atspindi visas galimas tokio žaidimo būsenas. Tokiame medyje tiesiog pasirenkamas trumpiausias kelias link laimėjimo.

Kai kurių žaidimų (šachmatų, Go ir Palago) žaidimo medis yra didžiulis ar net begalinis, todėl tenka dirbti su daliniu žaidimo medžiu ir analizuoti tik dalį viso medžio, pavyzdžiui, tik 5 artimiausius ėjimus. Arba galite įvertinti žaidimo situaciją ir tikrinti tik tuos kelius, kurie žada sėkmę.

Palago žaidimas ugdo erdvinį mąstymą, kuris labai svarbus sėkmingai sprendžiant daugelį informatikos problemų.

Palago yra kūrybinis meninis galvosūkis, kurį sukūrė Cameronas Browne'as. Nebūtina žaisti „Palago“ tik tam, kad laimėtumėte. Žaidėjai bando iš detalių sukurti palagonietį (Palagonijos pilietį Palago pasaulyje) arba gyvūnus:

