

S
i
d
r
i
š
n
i
s



Informatikos ir informatinio mąstymo uždavinių rinkinys

Nr. 6



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

Dėkojame Daumilui Ardickui, Tolmantui Dagiui, Karoliui Jasučiui, dr. Eglei Jasutei, dr. Tatjanai Jevsikovai, dr. Eimučiui Karčiauskui, Audronei Klupšaitai, Alvidai Lozdienei, Modestui Rimkui, dr. Broniui Skūpui, dr. Gabrielei Stupurienei, Tomui Šiauliui, talkinusiems verčiant ir adaptuojant uždavinius. Taip pat dėkojame tarptautinei „Bebro“ bendruomenei ir uždavinių autoriams.

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



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

Įvadas

„Bebro“ konkurso tikslas – atskleisti mokiniams ir mokytojams informatikos mokslo subtilybes, patraukti mokinių dėmesį, supažindinti su pagrindinėmis informatikos ir informacinių technologijų sąvokomis, jų platesniu kontekstu – konceptais, motyvuoti gilintis į modernius informatikos ir inžinerijos sprendimų priėmimo metodus. „Bebras“ – tai ne tik konkursas. Tai daugybė įvairių veiklų, kurios vyksta ištisus metus. Susikūrė stiprus pasaulinis „Bebro“ konkurso mokslininkų ir mokytojų tinklas, kasmet kuriami nauji uždaviniai, ieškoma aktualių temų.

Mokiniai, sprenddami įdomius informatikos uždavinius, susipažįsta su informatikos sąvokomis, pagilina jų sampratą, išsiugdo gebėjimą taikyti jas praktiškai, uždavinių sprendimą aptaria su bendraamžiais ir mokytojais. Didelė dalis uždavinių yra iš algoritmų ir programavimo srities. Juos perpratus, skatinama mokytis praktinio programavimo.

Vykdamas konkursą kaupiami mokinių sprendimai – daugybė duomenų surinkta, galima atlikti tyrimus, stebėti ir pan. Įvairių šalių mokslininkai, remdamiesi sukauptais „Bebro“ konkurso duomenimis, rengia ir publikuoja mokslinius straipsnius. Rengiamos parodos, festivaliai, kuriuose aptariami „Bebro“ uždaviniai. Dauguma šalių leidžia knygeles, kuriose aiškinami uždavinių sprendimai ir, svarbiausia, kaip kiekvienas uždavinys susijęs su informatika, kokie konceptai paslėpti, parodoma, kaip galima eiti toliau ir giliau.

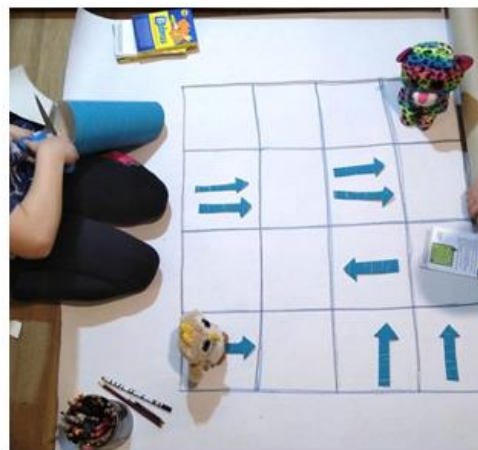
24 CAT AND MOUSE

Beaver created two robots: cat and mouse. Both of them can move from one square to another following the arrows. Cat wants to hunt the mouse.

- Cat starts first.
- Moves are made in alternately (cat, mouse, cat, mouse, etc.)
- The robots move in the direction indicated by the arrows as many squares as there are arrows (E.G. one square if there is one arrow, two squares if two arrows and etc.).
- When a robot is moving, it ignores the arrows on the squares.
- Mouse is eaten, when the cat is on the same square as the mouse.

Can the mouse avoid?

A playground is a simple program: start and finish is specified and there are rules that determine actions. Arrows in squares clearly and non-ambiguously determine where to go next. It is important that our mechanisms would understand these arrows. Cat and mouse are the mechanisms of this problem which can move according to the arrows.



Mokiniai sprendžia „Bebro“ kortelių uždavinius

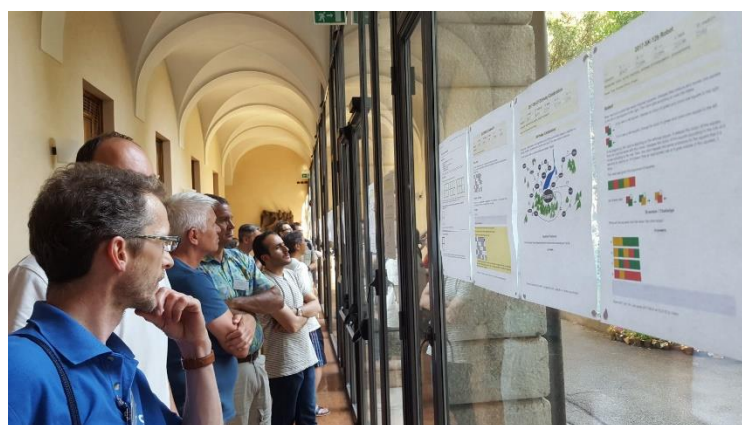


„Bebro“ uždavinių komplektai



„Bebro“ istorijos knygelė

Spręsdami „Bebro“ konkurso uždavinius mokiniai susipažįsta su pagrindiniais informatikos konceptais. Mokytojams uždaviniai – didaktiniai išteklių, idėjos savo pamokoms pajvairinti. Mokslininkai, informatikos tyrėjai kurdami uždavinius stengiasi perteikti esminius informatikos mokslo principus. Taigi „Bebro“ bendruomenę labiausiai vienija patrauklių, įdomių informatikos uždavinių kūrimas, klasifikavimas, idėjų paieška, informatikos konceptų išreiškimas žaidybinėmis užduotimis.



„Bebro“ uždavinių aptarimas



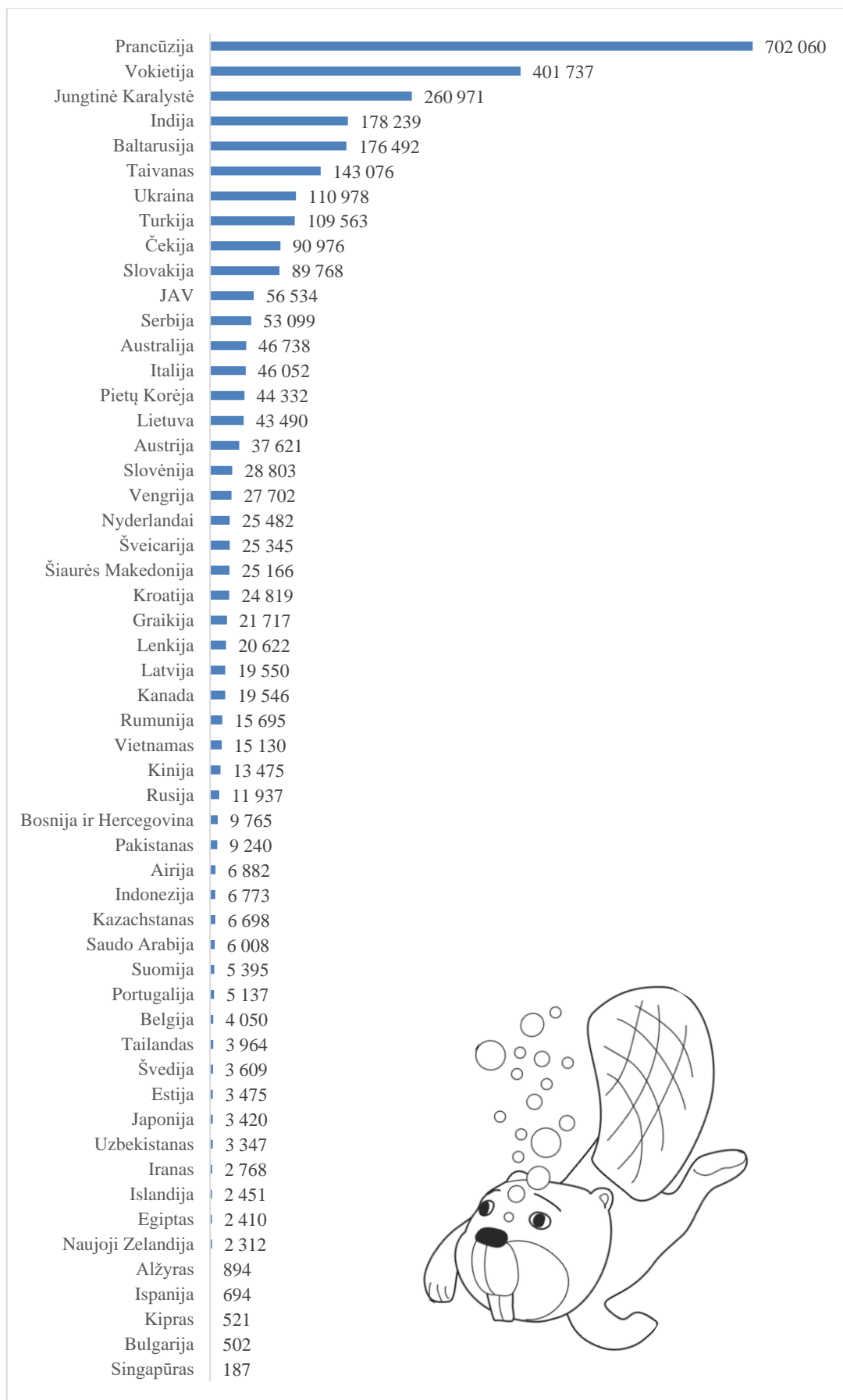
15-asis tarptautinis „Bebro“ uždavinių kūrimo simpoziumas Vengrijoje (2019 m.)

2019 metų „Bebro“ konkurso uždavinius pasaulyje sprendė per 2 977 000 mokinių iš 54 valstybių. Lietuvoje tais metais dalyvavo 43 555 mokiniai.

Nemažai šalių geriausiems dalyviams rengia akivaizdinį antrąjį etapą. Lietuva II etapą organizuoja nuo 2014 metų. Antrasis etapas rengiamas universitetuose ir kolegijose sausio mėnesio pabaigoje ar vasario pradžioje.

XVI informatikos ir informatinio mąstymo konkurso „Bebras“ II etapas vyko 2020 m. vasario 1 d. Jame dalyvavo 233 jaunieji (9–10 klasių mokiniai) ir 205 kolegos (11–12 klasių mokiniai).





2019 m. „Bebro“ dalyvių skaičius pagal šalis (54 šalių duomenys)

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

Nr.	Uždavinio pavadinimas	Uždavinio identifikatorius	Jauniai (9–10 kl.)	Kolegos (11–12 kl.)
1	Skaičių skirtumų automatas	2015-BG-02	6	6
2	Orų prognozė	2015-IT-02	6	6
3	Žodžių konvertavimas	2019-LT-05	6	6
4	Sūris	2015-CH-05	6	
5	Scenos šviesos	2015-SK-04	6	
6	Ūkininkų laukai	2019-CY-02	9	6
7	Automobiliu ar traukiniu?	2019-NL-03	9	6
8	Kartvelų patiekalas	2015-RU-02	9	9
9	Pelių vakarėlis	2019-LT-01	9	9
10	Korys	2019-PK-04	9	9
11	Slaptažodis	2017-PT-02b		9
12	Draugystės apyrankės	2019-BE-04	12	9
13	Gluosniai ir tuopos	2019-BE-03	12	12
14	Triguba bėda	2019-CA-06	12	12
15	Sukis, sukis, sukis!	2019-RU-05	12	12
16	Sandėliai	2019-RU-06	12	12
17	Robotas-siurblys	2017-LT-03		12

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

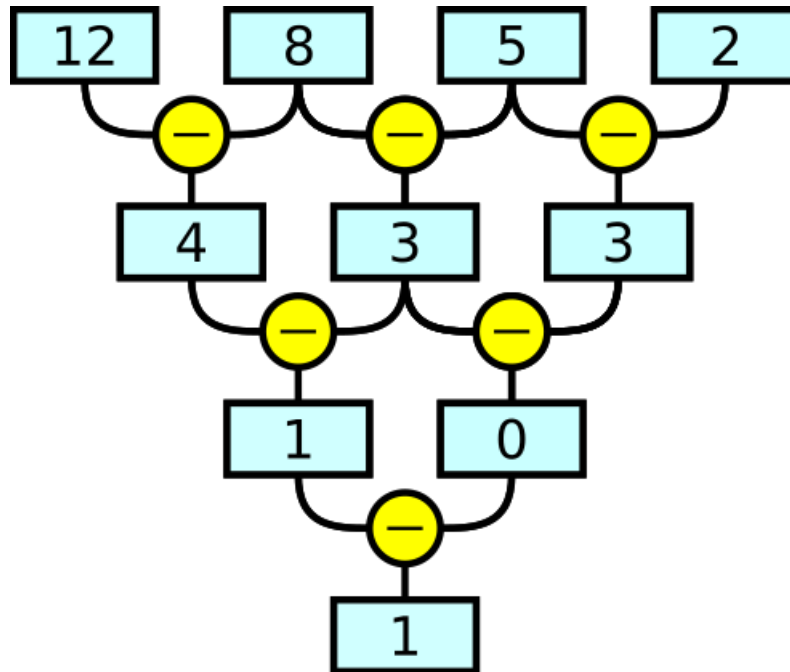
- lengvas – 6,
- vidutinis – 9,
- sunkus – 12.



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

1. Skaičių skirtumų automatas

Rūta suprogramavo automatą. Įvedama mažėjanti 4 teigiamų sveikųjų skaičių seka. Kiekvienoje tolesnėje eilutėje skaičiuojamas viršutinės eilutės dviejų gretimų skaičių skirtumas. Paskaičiavus visus galimus dviejų skaičių skirtumus, gaunamas vienas skaičius – rezultatas. Pavyzdžiui, įvedus skaičių seką 12 8 5 2, gauname:



Papildykite skaičių seką taip, kad ją įrašius į šį automatą, rezultatas būtų lygus nuliui.

18 5

Paaiškinimas

Teisingas atsakymas: 18-10-5-3.

Rezultatas yra nulis, jei trečiąją eilutę sudaro du vienodi skaičiai. Tai reiškia, kad antrosios eilutės skaičiai turi duoti tokį pat skirtumą kaip, pavyzdžiui, 7, 5, 3.

Tai informatika!

Šis skirtumų skaičiavimo automatas paaiškina veiksmų automatizavimą: atlieka paprastus aritmetinius skaičiavimus, kurie iš dalies panašūs į kompiuterio mikroschemose vykdomus procesus. Skaičiavimų apdorojimo procesas skaidomas į mažesnes dalis (žingsnius), kol galų gale nesunku apskaičiuoti rezultatą. Beje, šiuo metodu skaičiuoti skirtumus galima ir pritaikius skaičiuoklę.

2. Orų prognozė

Jonas turi failą, kurio 24 eilutėse surašytos rytojaus kiekvienos valandos orų prognozės nuo 00.00–01.00 iki 23.00–00.00. Kiekvienoje eilutėje parašytas vienas iš žodžių: saulė, debesis, lietus, sniegas. Norėdamas sužinoti, ar verta eiti į paplūdimį, Jonas gali naudoti šias komandas:

- **YRA a** – išrenka eilutes, kuriose yra žodis a
- **PIRMOSIOS n** – išrenka pirmąsias n eilučių
- **PASKUTINIOSIOS m** – išrenka paskutiniąsias m eilučių
- **SKAIČIUOTI** – suskaičiuoja ir parodo įvesties eilučių skaičių

Atskirdamas komandas ženklu „|“, Jonas gali sudaryti seką iš šių komandų. Pirmosios komandos įvestis visuomet yra orų prognozės failo turinys, o kiekvienos kitos komandos įvestis yra prieš tai esančios komandos rezultatas.

Sudėliokite komandų seką, kuri padėtų bebrui Jonui apsispręsti, ar eiti į paplūdimį.

YRA	PIRMOSIOS	PASKUTINIOSIOS	LIETUS	SAULĖ	6	11	13
							SKAIČIUOTI

Paaiškinimas

Teisingas atsakymas:

PASKUTINIOSIOS	11	PIRMOSIOS	6	YRA	SAULĖ	SKAIČIUOTI
----------------	----	-----------	---	-----	-------	------------

Tai informatika!

Šioje užduotyje naudojama keletas informatikos konceptų: algoritmas, komanda, seka, duomenų filtravimas, duomenų srautas. Kiekviena algoritmo komanda atrenka dalį įvedamų duomenų.

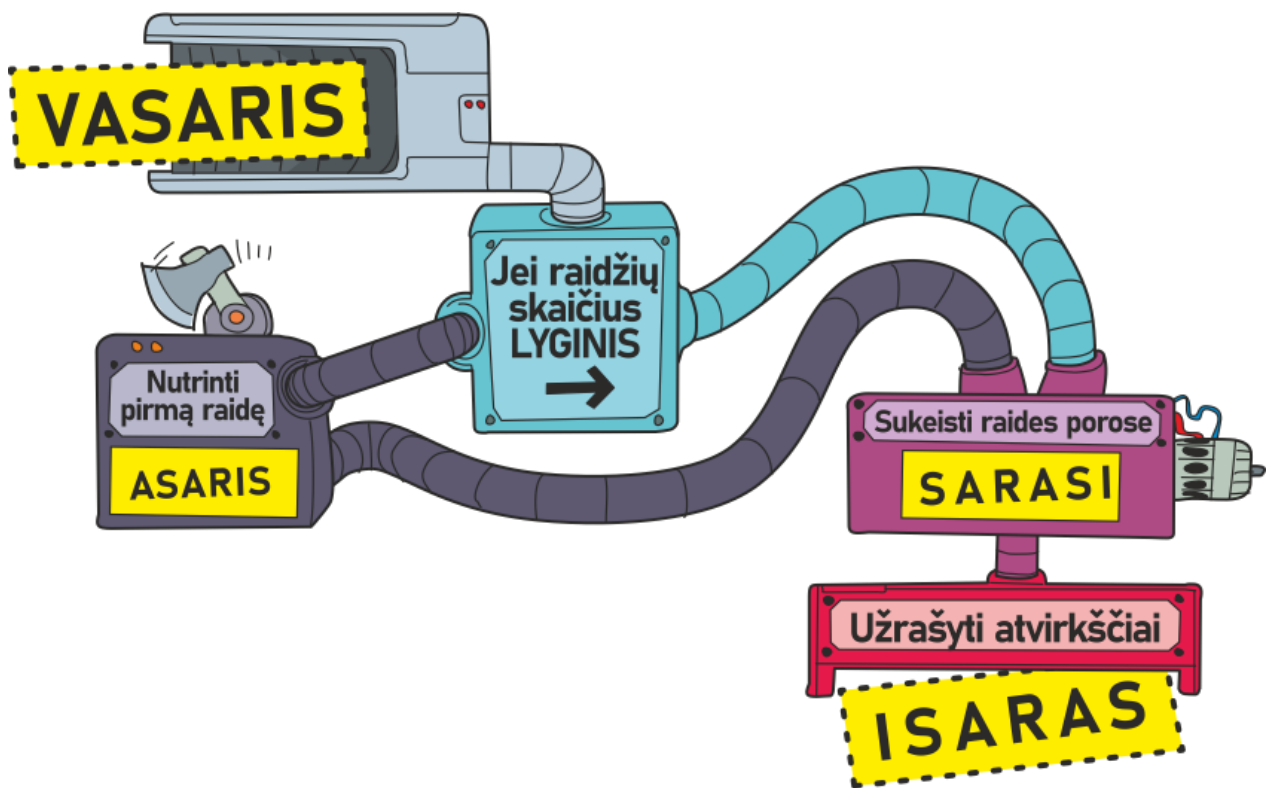
Šia užduotimi siekiama parodyti, kaip apdorojami kasdieniai duomenys, kaip taikomos komandos, sudaromos komandų sekos, kaip vienu komandų gauti rezultatai tampa kitų komandų pradiniais duomenimis.

3. Žodžių konvertavimas

Bebrai sukūrė mašiną, kuri konvertuoja žodžius tokiu būdu:

- Jei raidžių skaičius yra lyginis, žodis perduodamas nepakitęs.
- Priešingu atveju pirmoji žodžio raidė pašalinama.
- Toliau keičiama kiekviena gretimų raidžių pora: pirma ir antra raidės, po to trečia ir ketvirta raidės ir t. t.
- Paskutiniame etape gautas žodis pakeičiamas perrašytu atbulai.

Pavyzdžiui, žodis VASARIS konvertuojamas į ISARAS.



Jei mes turime žodį MEGABAITAS, tai gausime...

Paaiškinimas

Teisingas atsakymas: ASITBAGAME.

Žodyje MEGABAITAS raidžių skaičius yra lyginis, todėl žodis perduodamas nepakitęs. Kitame žingsnyje sukeitę gretimų raidžių poras gauname EMAGABTISA. Paskutiniame žingsnyje žodį perrašome atbulai: ASITBAGAME.

Tai informatika!

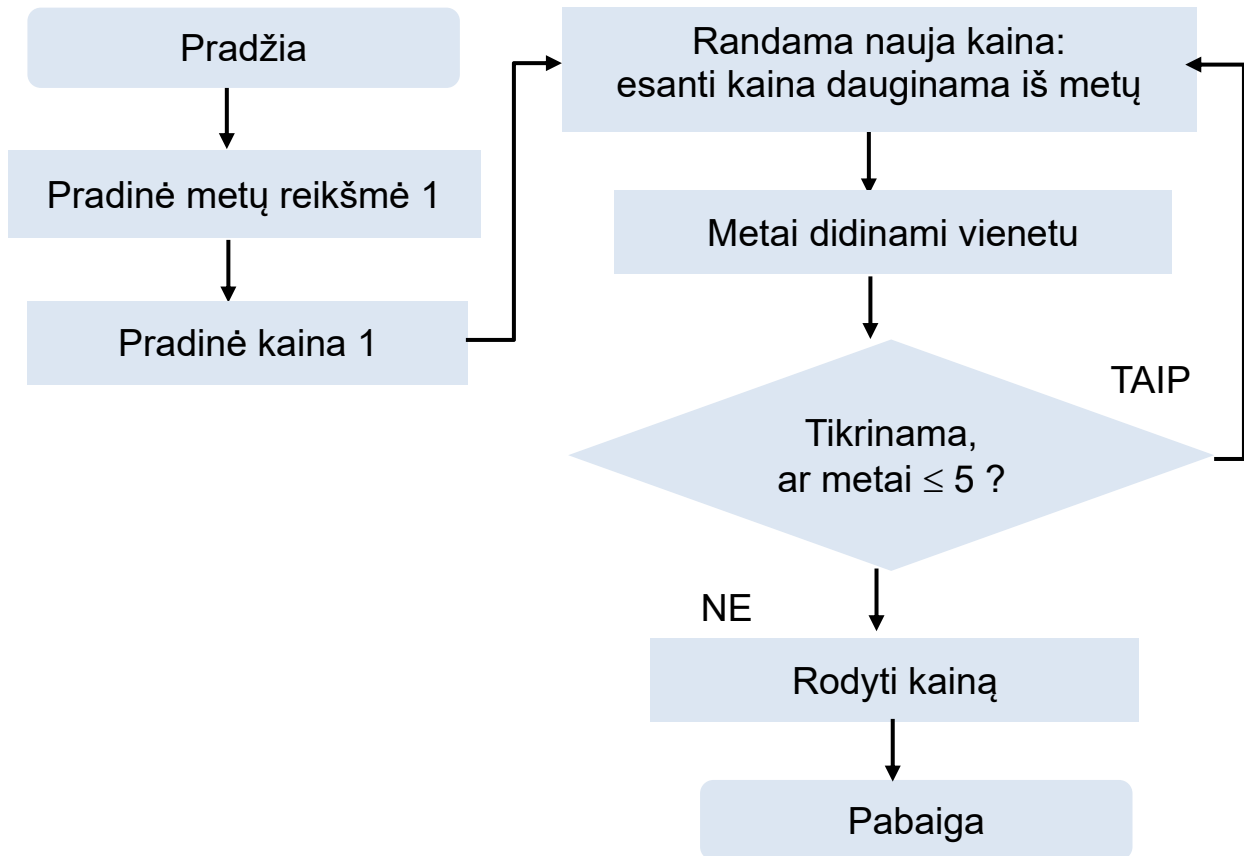
Šiame uždavinyje pateiktas paveikslėlis yra dalis paprastos struktūrinės schemos, kurioje paaiškinta, kaip žingsnis po žingsnio pakeisti žodį. Struktūrinės schemos – vaizdus būdas aprašyti algoritmus: atkreipkite dėmesį, kad struktūrinėse schemose galima užfiksuoti sprendimų idėjų (kaip mašinos dalyje „Jei...“) arba atlikti veiksmus (kaip mašinos dalyse „Nutrinti...“ arba „Sukeisti...“).

Struktūrinėmis schemomis galima pavaizduoti ciklus (pasikartojimą), kurie būtų vaizduojami rodykle „atgal“ pereiti į ankstesnę mašinos dalį.

Šiame uždavinyje algoritmas gauna žodį (simbolių seką) ir keisdamas raides gauna kitą žodį.

4. Sūris

Bebrijoje sūrio kaina priklauso nuo jo brandinimo laiko. Kainai apskaičiuoti naudojamos struktūrinės schemos. Štai ši pateikta schema skiriama 5 metus brandinto sūrio kainai sužinoti. Mažiausia sūrio kaina – 2 pinigai.



Kiek kainuoja 5 metus brandintas sūris?

Paaiškinimas

Pradinė kaina = 2, metai = 1.

Tada nauja kaina = $2 \times 1 = 2$, metai = $1 + 1 = 2$. Metų skaičius yra mažesnis už 5, todėl vėl skaičiuojama iš naujo.

Po dvejų metų kaina = $2 \times 2 = 4$, metai = $2 + 1 = 3$.

Po trejų metų kaina = $4 \times 3 = 12$, metai = $3 + 1 = 4$.

Po ketverių metų kaina = $12 \times 4 = 48$, metai = $4 + 1 = 5$.

Po penkerių metų kaina = $48 \times 5 = 240$, metai = $5 + 1 = 6$.

Metų skaičius jau didesnis už 5, todėl toliau nebeskaičiuojama ir rodoma galutinė kaina.

Tai informatika!

Šioje užduotyje programa išreiškiama struktūrine schema. Tai įprastas būdas, kai nevirtojama jokia programavimo kalba. Pradedama nuo **pradžios**, toliau einama pagal rodykles, kol pasiekiamas **pabaiga**. Programoje pateikiamas ciklas ir veiksmų, kuriuos kompiuteris turi atlikti keletą kartų, seka. Šiuo atveju veiksmas – tai sąlyginis ciklas, t. y., ciklas veiksmų, kartojamų tol, kol tenkinama nurodyta sąlyga.

Keblumų kelia tai, kad kintamieji (**metai** ir **kaina**) keičia savo reikšmes ir rezultatas priklauso nuo skirtingos metų reikšmės.

Šiuo atveju programa skaičiuoja faktorialą, kuris naudojamas ne tik įvairiose matematikos srityse, bet ir informatikoje.

5. Scenos šviesos

Trys prožektoriai apšviečia sceną raudona, žalia ir mėlyna šviesa. Scenos šviesos spalva priklauso nuo įjungtų prožektorių.

Lentelėje parodyti visi galimi spalvų deriniai.

Raudonas	Žalias	Mėlynas	Scenos spalva
Išjungtas	Išjungtas	Išjungtas	Juoda
Išjungtas	Išjungtas	Įjungtas	Mėlyna
Išjungtas	Įjungtas	Išjungtas	Žalia
Išjungtas	Įjungtas	Įjungtas	Žalsvai mėlyna
Įjungtas	Išjungtas	Išjungtas	Raudona
Įjungtas	Išjungtas	Įjungtas	Purpurinė
Įjungtas	Įjungtas	Išjungtas	Geltona
Įjungtas	Įjungtas	Įjungtas	Balta

Spektaklio metu prožektoriai įjungiami ir išjungiami laikantis tam tikros tvarkos:

- Raudonas prožektorius tris minutes būna įjungtas, tris minutes išjungtas.
- Žalias prožektorius dvi minutes būna įjungtas, vieną minutę išjungtas.
- Mėlynas prožektorius vieną minutę būna išjungtas, vieną minutę įjungtas.

Kokiomis spalvomis bus apšviesta scena per pirmąsias keturias spektaklio minutes? Nurodyk scenos šviesos spalvą kas minutę:

Juoda	Mėlyna	Žalia	Žalsvai mėlyna	Raudona	Purpurinė	Geltona	Balta
-------	--------	-------	----------------	---------	-----------	---------	-------



I minutė



II minutė



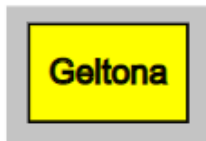
III minutė



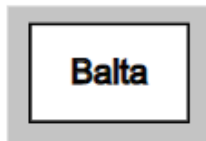
IV minutė

Paaiškinimas

Teisingas atsakymas:



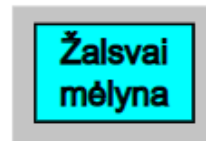
I minutė



II minutė



III minutė



IV minutė

	1 minutė	2 minutė	3 minutė	4 minutė
Raudonas prožektorius	Įjungtas	Įjungtas	Įjungtas	Išjungtas
Žalias prožektorius	Įjungtas	Įjungtas	Išjungtas	Įjungtas
Mėlynas prožektorius	Išjungtas	Įjungtas	Išjungtas	Įjungtas
Scenos spalva	Geltona	Balta	Raudona	Žalsvai mėlyna

Tai informatika!

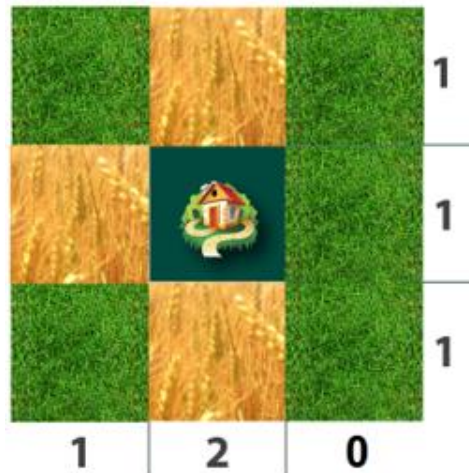
Spalvų sudarymas iš trijų pagrindinių spalvų – raudonos, žalios ir mėlynos – vadinamas RŽM (angl. RGB) spalvų modeliu. Šis modelis naudojamas kompiuterių monitoriuose, televizorių ekranuose. Paveikslėliai yra sudaryti iš pikselių, kurie nuspalvinti naudojant šių trijų spalvų derinius. Tai pagrindinis kompiuterinės grafikos principas.

Užduotyje iliustruojama ir veiksmų seka – atitinkamų spalvų prožektorių įjungimas ir išjungimas. Tai vienas iš svarbiausių algoritminio mąstymo elementų. Be to, užduotyje naudojamas ir lygiagretusis trijų veiksmų sekų atlikimas.

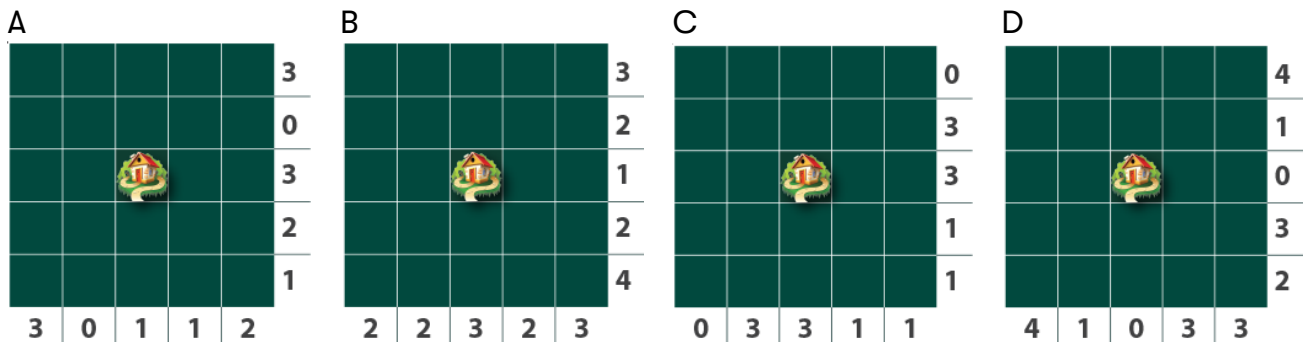
6. Ūkininkų laukai

Bebrijos ūkiai yra kvadrato formos, kiekvieno ūkio plotas taip pat sudalintas į kvadratinius laukus. Ūkininko namas visada stovi viso ūkio centre. Kiekvienais metais ūkininkai turi nuspręsti, kuriuose kvadratiniuose laukuose augs kviečiai ir kuriuose – žolė. Valstybinėje ataskaitoje kiekvienoje eilutėje ir kiekviename stulpelyje surašoma, kiek juose yra kviečių laukų. Palydovas patikrina ataskaitos duomenis.

Pavyzdys – paveikslėlyje pateikta vieno ūkininko ataskaita.

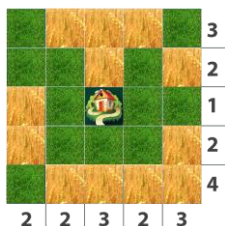


Kuri ataskaita be klaidų?



Paaiškinimas

Teisingas atsakymas: B.



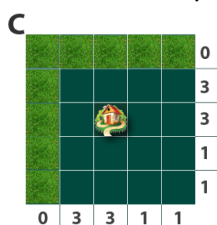
Atsakymuose A ir D nesutampa eilutėse ir stulpeliuose esančių kviečių laukų sumos:

A: eilutėse yra 7 kviečių laukai, stulpeliuose – 9,

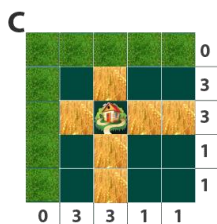
D: eilutėse yra 11 kviečių laukai, stulpeliuose – 10.

Kadangi C atsakymo eilučių ir stulpelių sumos sutampa, tai turime manyti kitaip:

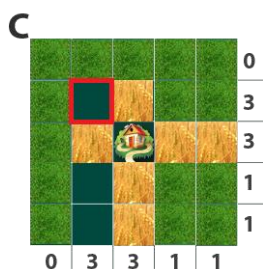
Pirma eilutė ir pirmas stulpelis turi būti tušti:



Trečioje eilutėje ir trečiame stulpelyje esančius kviečių laukus galima užsėti tik vienu būdu:



Tada visi 4 ir 5 stulpelių laukai turi būti padengti žole:



Po to neįmanoma pasėti kviečių dar dviejuose antros eilės laukuose, nes yra tik vienas laukas (pažymėtas raudonai).

Tai informatika!

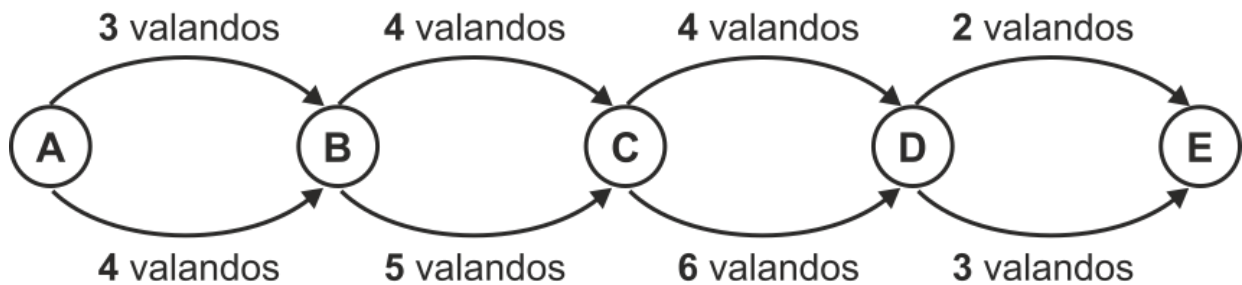
Masyvas yra viena svarbiausių duomenų struktūrų, naudojamų programavimo kalbose. Masyvą sudaro panašūs objektai, mūsų atveju kviečių laukai, išdėstyti eilutėmis ir stulpeliais. Šiame uždavinyje eilučių ir stulpelių sumos yra labai svarbūs duomenys.

Grįžimo algoritmas (angl. *backtracking algorithm*): sprendžiama palaipsniui, imant po vieną reikiamą dalį, šalinant sprendinius, kurie netenkina uždavinio ribojimų.

7. Automobiliu at traukiniu?

Jonas turi nukeliauti iš taško A į tašką E per taškus B, C ir D. Jis gali keliauti traukiniu arba automobiliu. Važiavimo trukmė tiek traukiniu, tiek automobiliu yra pateikta paveiksle.

Kelionės laikas automobiliu



Kelionės laikas traukiniu

Jonui labai rūpi aplinkosaugos problemos, jis stengiasi mažinti įtaką klimato kaitai. Jonas nori keliauti automobiliu kaip įmanoma mažiau. Deja, jis turi pasiekti E tašką per 15 valandų, todėl dalį kelio privalo važiuoti automobiliu.

Kiek mažiausiai laiko Jonas turės važiuoti automobiliu?

Paaiškinimas

Teisingas atsakymas: 6.

Jonas mieliau keliautų traukiniu. Visa kelionė jam užtruktų 18 valandų. Taigi jis turi pakeisti bent tris valandas kelionės traukiniu važiuodamas automobiliu.

	Sutaupytas laikas keliaujant automobiliu, o ne traukiniu	Važiavimo laikas
Iš taško A į tašką B	1 valanda	3 valandos
Iš taško B į tašką C	1 valanda	4 valandos
Iš taško C į tašką D	2 valandos	4 valandos
Iš taško D į tašką E	1 valanda	2 valandos

Mūsų uždavinį galima suformuluoti taip: minimizuoti bendrą važiavimo laiką su sąlyga, kad sutaupytas laikas sudarytų ne mažiau kaip 3 valandas. Lentelėse pateiktos keturios mainų galimybės.

	Sutaupytas laikas	Važiavimo laikas
Iš A į B	1	3
Iš B į C	1	4
Iš D į E	1	2
Iš viso:	3	9

	Sutaupytas laikas	Važiavimo laikas
Iš B į C	1	4
Iš C į D	2	4
Iš viso:	3	8

	Sutaupytas laikas	Važiavimo laikas
Iš C į D	2	4
Iš D į E	1	2
Iš viso:	3	6

	Sutaupytas laikas	Važiavimo laikas
Iš A į B	1	3
Iš C į D	2	4
Iš viso:	3	7

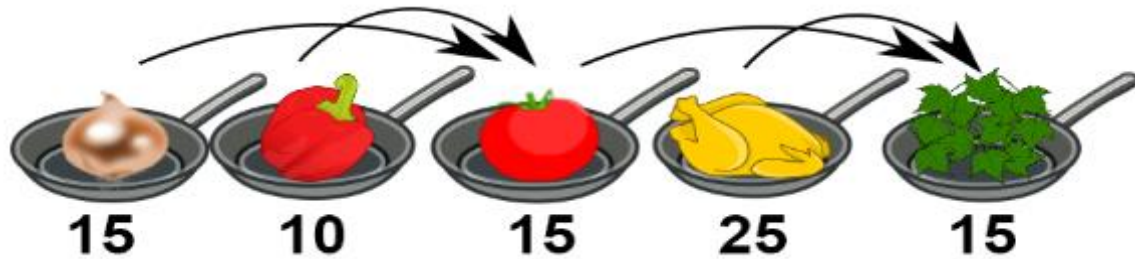
Geriausias būdas sutaupyti 3 valandas – automobiliu važiuoti iš C į D ir iš D į E (trečioji lentelė). Taip sutaupomos 3 valandos, o vairuoti reikia tik 6 valandas. Visi alternatyvūs keitimo variantai reikalauja ilgesnio vairavimo laiko.

Tai informatika!

Šis uždavinys susijęs su informatiniu mąstymu. Uždavinys aprašomas remiantis schema, sudaryta iš rodyklių ir apskritimų. Tokia schema vadinama grafu. Spręsdami pirmiausia sudarėme lentelę, remdamiesi informacija, pateikta grafe, ir apskaičiavome naujus skaičius (taupydami laiką). Uždavinys performuluotas į naują uždavinį, kurį išspręsti kur kas paprasčiau.

Toks uždavinių performulavimas yra svarbus informatinio mąstymo pavyzdys. Tokio tipo uždaviniai apima objektus, kurių būseną turi tenkinti tam tikrus ribojimus. Ribojimų uždaviniai, be kitų būdų, gali būti sprendžiami performuluojant uždavinį į paprasčiau sprendžiamą uždavinį, kaip matėme šiuo atveju.

8. Kartvelų patiekalas



Ilja gamina tradicinį kartvelų patiekalą, vadinamąjį čachochbilį (kartv. ჩახოჩბილი). Gaminant reikia atlikti šiuos žingsnius (nurodyta, kiek trunka kiekvienas žingsnis):

1	Pakepinti susmulkintą svogūną	15 minučių
2	Pakepinti supjaustytą papriką	10 minučių
3	Kepti 1-ojo ir 2-ojo žingsnio rezultatus drauge su pjaustytu pomidoru	15 minučių
4	Kepti viščiuką	25 minutes
5	Kepti 3-iojo ir 4-ojo žingsnio rezultatus drauge su prieskoniais	15 minučių

Kai Ilja gamina sode, turi tik vieną kaitvietę, todėl veiksmus (žingsnius) turi atlikti paeiliui. Kad pasigamintų čachochbilį sode, Iljai reikia 80 minučių.

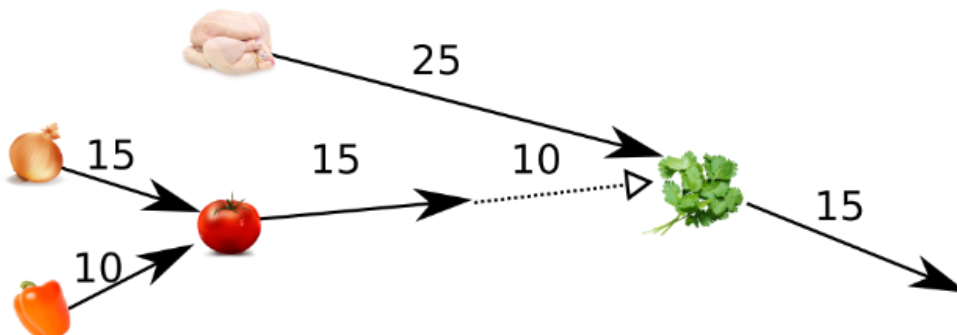
Gamindamas namie, Ilja gali naudotis šešiomis kaitvietėmis. Kai kuriuos žingsnius jis gali daryti tuo pačiu metu ir taip sutaupyti laiko.

Kiek mažiausiai minučių prireiks Iljai, kad pasigamintų čachochbilį namie?

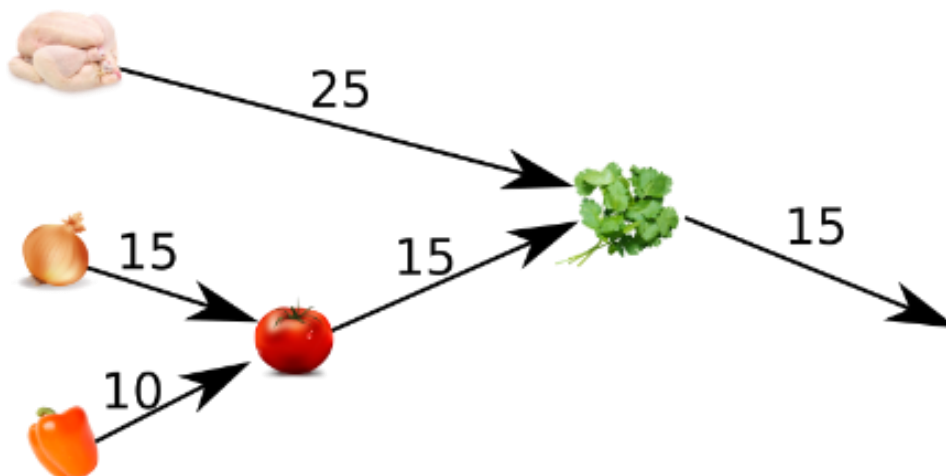
Paaiškinimas

Teisingas atsakymas: 45.

Su dviem kaitvietėmis IĮjai reikia mažiausiai 50 minučių:



Su trimis kaitvietėmis IĮjai reikia mažiausiai 45 minučių:



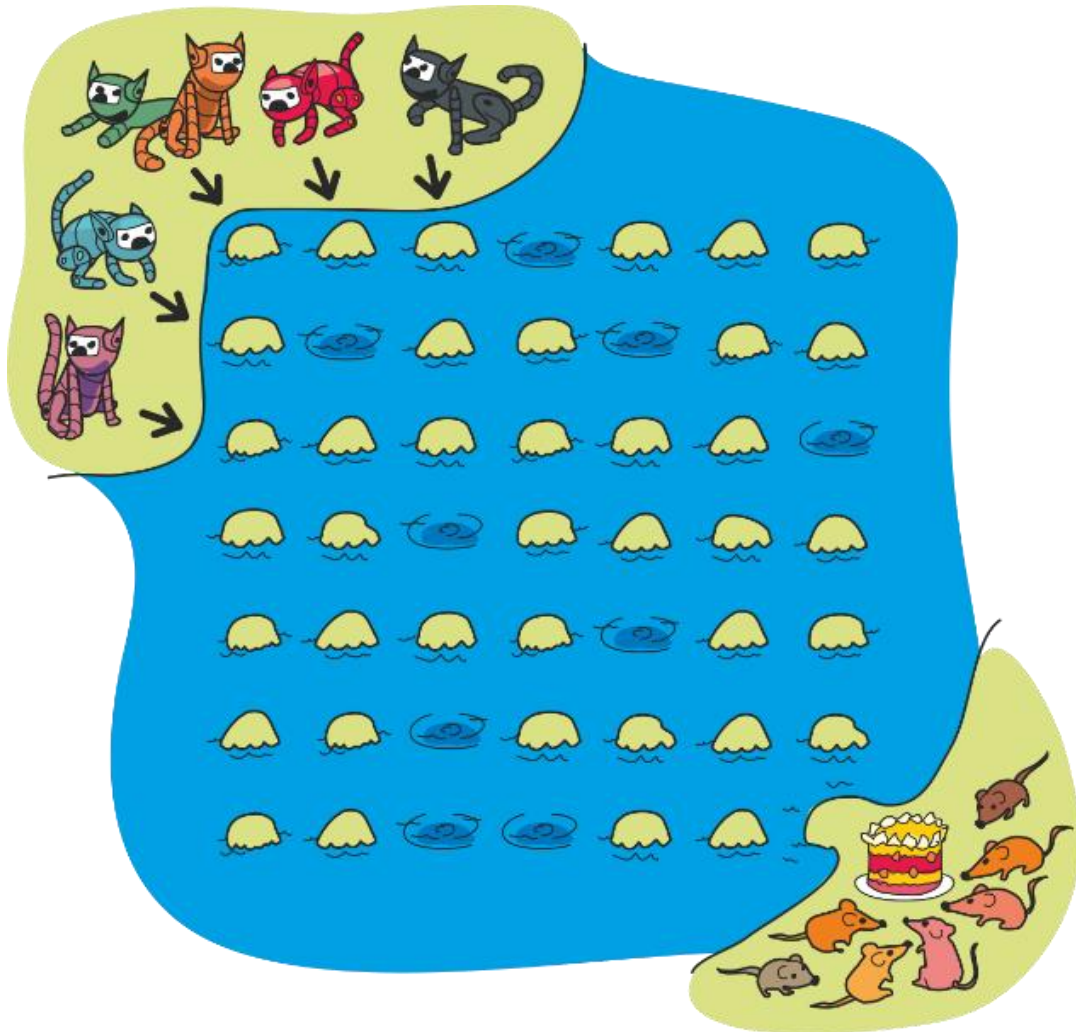
Tai informatika!

Užduotyje naudojamos kaitvietės galima laikyti kompiuterio procesoriais. Jei turite tik vieną procesorių, reikia visus žingsnius atlikti nuosekliai vieną po kito. Jei turite daugiau – tai kai kuriuos žingsnius galima atlikti lygiagrečiai.

Laiko mažinimas yra panašus į struktūrinio programavimo tekstą, vykdomą kiek įmanoma greičiau naudojant tam tikrą skaičių procesorių. Tai programos optimizavimo pavyzdys. Lygiagretusis skaičiavimas yra informatikos mokslo tyrinėjimo sritis.

9. Pelių vakarėlis

Šešios katės eina į pelių vakarėlį šokinėdamos iš salos į salą. Katės gali nušokti tik į kaimyninę salą. Jos negali šokinėti įstrižai ir į salas, esančias po vandeniu. Katė gali šokinėti labai greitai, todėl laikas, reikalingas šuoliui, yra lygus nuliui. Šokdama katė išsenka, todėl ji vieną minutę praleidžia saloje ir tik tada šoka ant kitos salos, jei ta sala nėra užimta. Priešingu atveju katė dar minutę išbūna toje pat saloje. Katės gali pradėti šokinėti nuo bet kurio įėjimo, pažymėto rodyklėmis.



Kiek mažiausiai minučių truks, kol visos katės nušokinės iš pradinės vietos į vakarėlio vietą?

Paaiškinimas

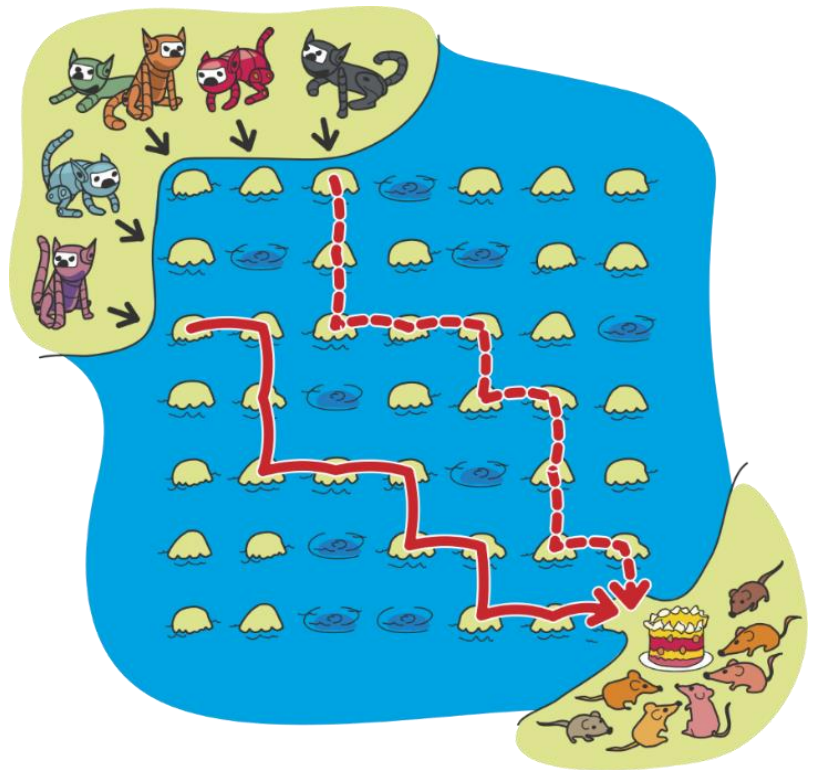
Teisingas atsakymas: 12.

Iš pradžių turime rasti trumpiausius kelius nuo kiekvieno galimo ėjimo: tai 10, 11 ir 12 minučių. Tada svarbu atkreipti dėmesį, kad yra lygiai du skirtingi trumpiausieji keliai (einant dviem lygiagrečiais keliais), kaip parodyta paveikslėlyje.

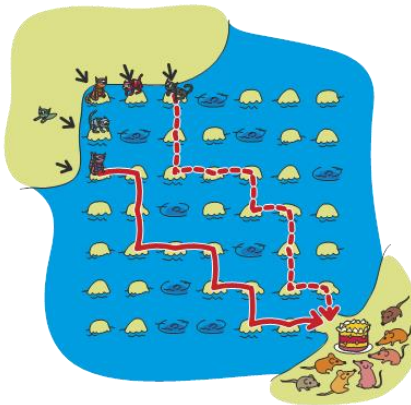
Taigi 5 katės gali atvykti per 12 minučių. Paskutinė 6-oji katė gali pradėti eiti tuo pačiu metu kaip ir 5-oji katė, užšokusi ant vieno iš jėimų, esančių netoli kampo.

Katė keliauja ir atvyksta į vakarėlio salą per 10 minučių. Jei katės yra dvi, laikas vis tiek yra 10 minučių. Jei katės yra trys, visos jos negali eiti trumpiausiais keliais lygiagrečiai, nes yra tik du trumpiausieji keliai. Taigi trečioji katė turi keliauti paskui pirmąją arba antrąją katę vienu iš trumpiausių kelių, o tai užtruks 11 minučių.

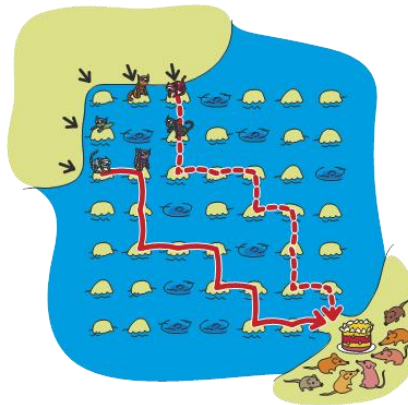
Panašiai keturios katės užtruks tiek pat laiko, kiek ir trys katės. Vadinasi, penkioms arba šešioms katėms prireiks 12 minučių.



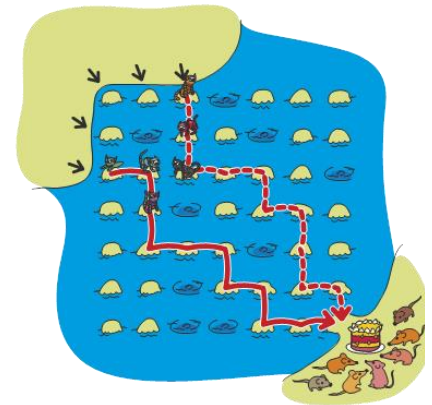
Pirmas žingsnis



Antras žingsnis



Trečias žingsnis



Kodėl jos negali to padaryti greičiau?

Kiekviena katė turi praeiti pro pirmąją jėjimo salą arba pro paskutinę jėjimo salą. Tada, jei katė pradeda nuo kurios nors kitos salos ir praeina pro vieną iš šių dviejų, galime manyti, kad ši katė ką tik pradėjo nuo čia, ir pamiršti visas ankstesnes kelio dalis. Taigi dabar šį uždavinį sumažinome iki uždavinio, kuriame yra tik du jėjimai.

Kiekvienam iš šių dviejų jėjimų yra akivaizdus trumpiausias 10 minučių kelias į pelių vakarėlį, parodytas paveikslėlyje. Taigi, kates galime padalyti į du srautus. Kadangi viename iš srautų turėtų būti bent 3 katės, turime pridėti bent po 2 minutes antrai ir trečiai katei kiekviename sraute, taigi mums reikia bent 12 minučių.

Jei trys katės iš vieno iš mūsų dviejų jėgimų pradeda eiti viena po kitos, o kitos trys katės tokiu pat būdu pradeda eiti iš kito jėgimo, turime lygiai 12 minučių.

Tai informatika!


















Tai situacijų modeliavimo ir vertinimo uždavinys, susijęs su grafais. Viena iš pagrindinių užduočių – nustatyti, kaip gali elgtis matematiškai modeliuojamos sistemos ir kiek laiko gali trukti matematiškai modeliuojamas procesas. Optimizavimo uždavinys – tai uždavinys, kai reikia rasti geriausią sprendinį iš visų galimų.

Informatikoje svarbu realaus pasaulio situacijas paversti kompiuteriams suprantamomis duomenų struktūromis. Duomenų vieta arba padėtis dažnai vaizduojama tinkleliu. Reikia rasti trumpiausią kelią keliaujant tik vertikaliai ir horizontaliai tinklelyje, išvengiant salų po vandeniu.

Be to, šį uždavinį galima laikyti grafų teorijos uždaviniu. Salų aibę (įskaitant jėgimus ir pelių vakarėlio vieta) galime pavaizduoti kaip grafo viršūnes, o gretimos viršūnės atitinka kaimynines salas. Tokiu atveju tiesiog ignoruotume po vandeniu esančias salas. Taigi mūsų uždavinys susijęs su didžiausio srauto uždaviniu: galime pastebėti, kad yra dviejų salų „pjūvio aibė“, todėl vienu metu per mūsų grafą gali praeiti ne daugiau kaip dvi katės, o likusioms katėms palaukti reikia papildomo laiko.

10. Korys

Bitininkas turi avilį, kurį nori pastatyti į tokią vietą, nuo kurios suma atstumų iki visų gėlių būtų mažiausia. Gėlių laukas pavaizduotas gardele, kurios eilutės sunumeruotos nuo 1 iki 9, o stulpeliai nuo A iki I. Bitės šiame lauke skrenda tik vertikaliai arba horizontaliai, taigi atstumas tarp langelių yra horizontalaus ir vertikalio atstumų suma (pavyzdžiui, atstumas tarp C4 ir D7 yra 4 (3 langeliai vertikaliai ir 1 horizontaliai)).

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									



Kuriame langelyje bitininkas turėtų pastatyti avilį, kad suma atstumų nuo avilio iki kiekvienos gėlės būtų mažiausia? Nutempkite avilį į šį langelį.

Paaiškinimas

Teisingas atsakymas: D5.

Šį uždavinį galima sutrumpinti iki vidurinės gėlės eilutės vertikaliai ir vidurinės gėlės stulpelio horizontaliai radimo. Kadangi lauke yra 17 gėlių, vidurinė gėlė yra 9-oji vertikaliai ir horizontaliai (vienoje jos pusėje yra 8 gėlės, o kitoje – irgi 8 gėlės). Vertikaliai surūšiuotas vidurinis žiedas yra A5 eilutėje, todėl avilys turėtų būti 5 eilutėje. Horizontaliai išrikiuota vidurinė gėlė yra arba D3, arba D9, nes jos abi yra tame pačiame stulpelyje, todėl avilys turėtų būti D stulpelyje.

Kadangi atstumas laikomas vertikalojo ir horizontaliojo atstumo suma (o ne euklidinis, tiesiaiegis, atstumas), uždavinį galima padalyti į nepriklausomą eilutės ir stulpelio radimą, nes avilio perkėlimas pagal vieną koordinatę neturi įtakos atstumams pagal kitą koordinatę. Taigi tas pats algoritmas, kuris naudojamas optimaliam stulpeliui rasti, tinka ir optimaliai eilutei rasti, o optimalių vietų nurodys optimalus stulpelis ir optimali eilutė.

Tai informatika!

Pasirinktam šio uždavinio sprendimui (kai neskaičiuojami visų kelių atstumai) naudojama algoritmo projektavimo „skaldyk ir valdyk“ metodas, kuris pagrindinį uždavinį padalina į du uždavinius: optimalios eilutės ir optimalaus stulpelio paiešką. Išsprendus šiuos du uždavinius, rezultatai sujungiami.

11. Slaptažodis

Bebras būsimoms statyboms kaupia tvirtus rąstus. Juos sudeda į savo buveinę ir užrakina. Spyna užrakinama 8 dvejetainių (tik 0 ir 1) skaitmenų ilgio slaptažodžiu. Kartą bebras atvilko puikių klevų ir gluosnių rąstų, užrakino ir... pamiršo slaptažodį.

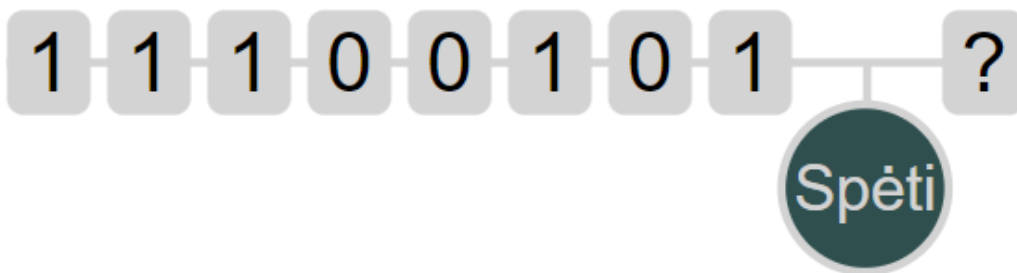
Bebro spyna – išmanioji: ekrane rodomi įvedami 8 slaptažodžio skaitmenys, taip pat yra spėjimo mygtukas ir rezultato langelis. Įvedus bet kokį slaptažodį ir paspaudus spėjimo mygtuką, rezultato langelis rodo štai ką:

- jei teisingai atspėti lygiai 4 skaitmenys, rodomas skaičius 4;
- jei teisingai atspėti visi 8 skaitmenys, rodomas skaičius 8;
- bet kuriuo kitu atveju rodomas brūkšnelis „-“.

Pavyzdžiui, jei bebras spėjo 10110111, o teisingas slaptažodis yra 11010101, tai rezultato langelyje rodoma „-“.

Tačiau jei bebras įveda 10110110, rezultato langelyje matomas skaičius 4, o jei įveda 11010101, tai matomas skaičius 8.

Padėk bebrui atspėti slaptažodį.



Paaiškinimas

Kadangi rezultatas reikšmingos informacijos pateikia tik tada, kai atspėjame 4 skaitmenis, tai turime bandyti atspėti tuos skaitmenis. Pirmiausia spėjame bet kurių skaitmenų eilutę. Jei joje nėra 4 skaitmenų, tai, tarkime, kad yra X teisingų skaitmenų. Invertuojame savo spėjimą (kiekvieną 0 keičiame 1, o kiekvieną 1 keičiame 0) ir galų gale gauname eilutę su $8 - X$ teisingais skaitmenimis. Jei X yra mažiau už 4, tada $8 - X$ yra daugiau už 4, o jei X yra daugiau už 4, tada $8 - X$ yra mažiau už 4. Taigi invertuojame pirmąjį skaitmenį ir spėjame, o jei nepavyksta, invertuojame antrąjį skaitmenį ir spėjame, tada trečiąjį ir taip toliau, kol atspėjame 4 teisingus slaptažodžio skaitmenis.

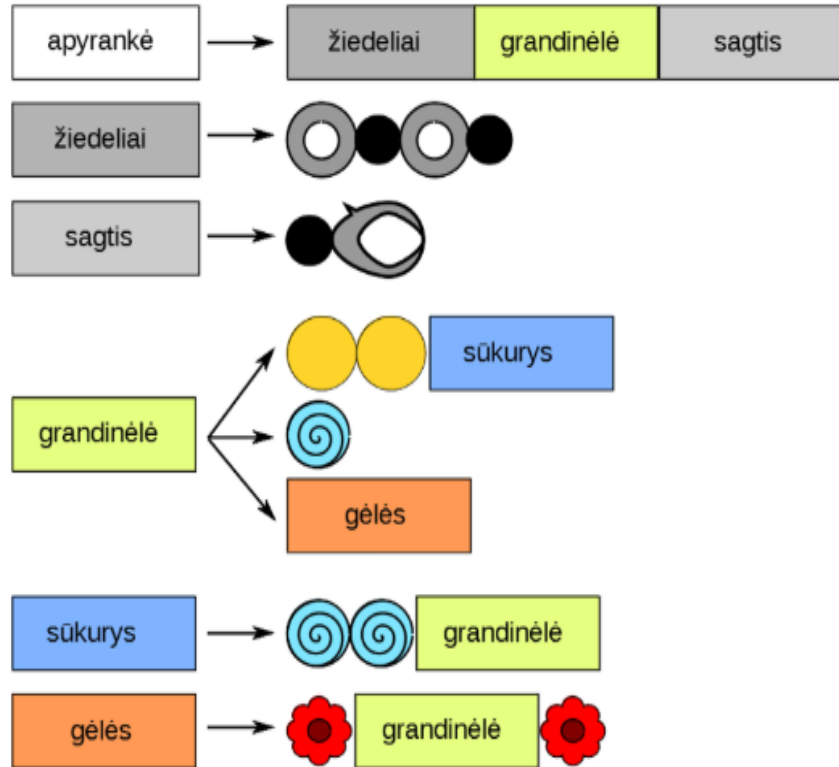
Atspėję 4 teisingus skaitmenis turime nustatyti, kurie skaitmenys yra neteisingi, ir atspėti visą slaptažodį. Tai darome, paeiliui invertuodami pirmąjį ir antrąjį skaitmenis ir stebėdami rezultatą: gali būti 2 arba 6 teisingi skaitmenys (rezultatas lygus „-“) arba lieka 4 atspėti skaitmenis. Šitaip sužinosime, kad pirmi du skaitmenys yra arba abu teisingi, arba abu neteisingi (pastaruoju atveju rodomas rezultatas 4). Tada analogiškus veiksmus atliekame su pirmuoju ir trečiuoju skaitmenimis, toliau – su pirmuoju ir ketvirtuoju skaitmenimis ir t. t. Šitaip galų gale gausime visų teisingų skaitmenų eilutę arba visų neteisingų skaitmenų eilutę – pastarosios visus skaitmenis invertuotume. Pagaliau atspėjame slaptažodį.

Tai informatika!

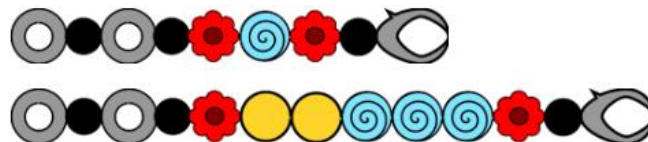
Dvejetainė skaičiavimo sistema yra kompiuterio veikimo ir skaičiavimų pagrindas. Kompiuteris bet kokią informaciją išreiškia dviem būsenomis. Šiame uždavinyje svarbu pastebėti, kaip pateikiamas grįžtamasis ryšys (rezultatas). Pateikę pradinį duomenį ir gavę rezultatą, galime geriau pasirinkti tolesnius spėjimus.

12. Draugystės apyrankės

Stepas gamina apyrankes pagal šias taisykles:







Kiekvienas kairėje esantis simbolis pakeičiamas į dešinėje pateiktą simbolių seką. Pavyzdžiui, Stepas pagamino šias dvi apyrankes laikydamasis nurodytų taisyklių:



Savo draugams Stepas pagamino keturias apyrankes. Vienas iš draugų nutraukė savo apyrankę ir padarė klaidą bandydamas ją sutaisyti.

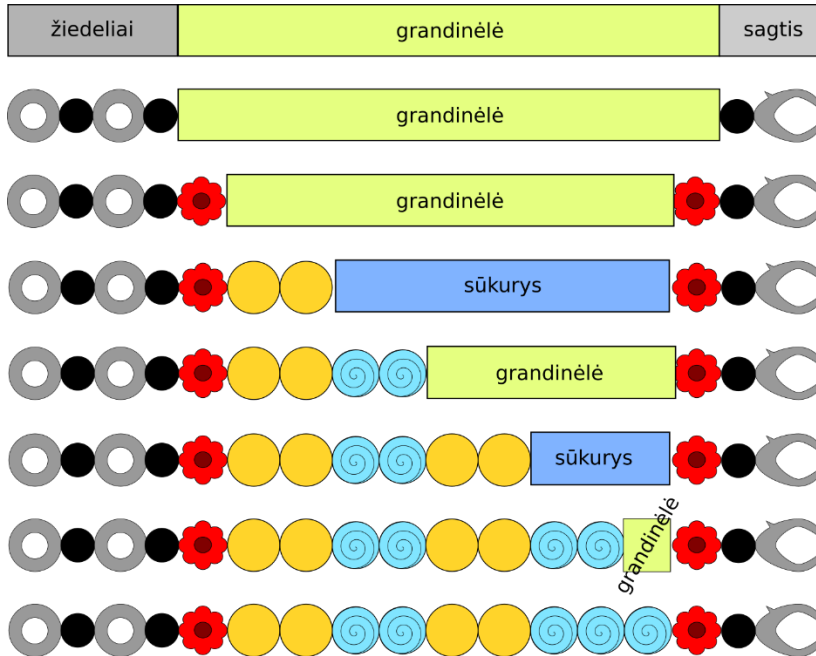
Kurioje apyrankėje yra padaryta klaida?

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

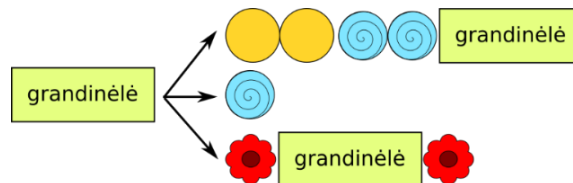
Paaiškinimas



Teisingas atsakymas: C.

Vietoj to, kad įrodytume, jog C apyrankėje yra klaida, būtų lengviau įrodyti, kad kitas tris apyranke iš tiesų galima pagaminti pagal šias taisykles. D apyrankė parodyta paveikslėlyje:



Nesudėtinga panašiai pavaizduoti A ir B. Kad įrodytume, jog C niekada negalima sukurti pagal taisykles, apžvelgsime „grandinėlių“, „sūkurio“ ir „gėlių“ taisykles. Jas kartu galima sujungti į naują taisyklę, kuri nekeičia apyranke, kurią galime pagaminti, šablono:



Jei nekreipsime dėmesio į gėlių karoliukus, kiekvieną apyrankę sudarys kelis kartus pakartotas raštas  (galbūt tik vieną kartą arba net 0 kartų), po kurio eina vienas .

A, B ir D apyranke aiškiai atitinka šį šabloną, tačiau C apyrankė neatitinka šio šablono – apyranke viduryje yra trys šviesiai mėlynos spalvos sūkuriai, o tai neleistina.

Tai informatika!

Remdamasis užduotyje pateiktomis taisyklėmis, kompiuteris galėtų nuspręsti, ar apyrankę pagamino Stepas, ar ne jis. Panašios taisyklės yra naudojamos patikrinti, ar programuotojai programose padarė rašymo klaidų, ar paieškos lauke įvestas tekstas yra taisyklingas ir pan. Pavyzdžiui, šios taisyklės:

- skaičius → be ženklo arba + be ženklo, arba - be ženklo
- be ženklo → skaičius arba skaičius be ženklo
- skaičius → 0 arba 1, arba 2, arba 3, arba 4, arba 5, arba 6, arba 7, arba 8, arba 9 gali būti naudojamos norint pasakyti kompiuteriui, koks skaičius yra leidžiamas.

Tokios taisyklės informatikoje vadinamos gramatika.

13. Gluosniai ir tuopos



Gluosnis



Tuopa

Bebras norėtų pasisodinti sode medžių, kad karštomis vasaros dienomis galėtų pailsėti pavėsyje. Jis nori išsiaiškinti, kuris medis – gluosnis ar tuopa – auga greičiau. Informacijai apie medžius ieškoti bebras naudoja paieškos programėlę planšetiniame kompiuteryje. Programėlėje galima naudoti du specialiuosius ženklus sudėtingesnėms paieškos užklausoms užrašyti:

... #	Abipus ženklo # užrašytos užklauskos sujungiamos. Tokiu būdu randami tik tokie tinklalapiai, kurių kiekvienas tenkina abi užklauskas.
[...]	Skliaustuose užrašyta užklausa invertuojama – gaunamas užklauskos neiginys. Randami tik tokie tinklalapiai, kurie netenkina užklauskos.

Pavyzdžiai:

Užklausa	Rezultatas
gluosnis	Tinklalapiai apie gluosnius.
[gluosnis]	Tinklalapiai NE apie gluosnius.
gluosnis # tuopa	Tinklalapiai, kurių kiekviename yra informacijos IR apie gluosnius, IR apie tuopas.
[[tuopa]]	Tinklalapiai apie tuopas (= tinklalapiai, kurie NĖRA NE apie tuopas).
[gluosnis # tuopa]	Tinklalapiai, kuriuose nėra informacijos NEI apie gluosnius, NEI apie tuopas.
gluosnis # [tuopa]	Tinklalapiai apie gluosnius, bet NE apie tuopas.

Kurios iš šių užklauskų rezultatas bus visi (ir tik tie) tinklalapiai, kurių kiekviename yra informacijos arba apie gluosnius, arba apie tuopas, arba ir apie gluosnius, ir apie tuopas?

- A. [gluosnis] # [tuopa]
- B. [[gluosnis] # tuopa] # [gluosnis # [tuopa]]
- C. [[gluosnis] # [tuopa]]
- D. [[gluosnis] # [tuopa]] # [gluosnis # tuopa]

Paaiškinimas

Teisingas atsakymas: [[gluosnis] # [tuopa]].

Norėdami suprasti, kodėl šis atsakymas teisingas, pirmiausia pažvelkime į atsakymą A, kuris yra šiek tiek paprastesnis: [gluosnis] # [tuopa]. Šia užklausa prašoma surasti visus tinklalapius, kurie atitinka ir „[gluosnis]“, ir „[tuopa]“, kitaip tariant, visus, kuriuose NĖRA informacijos apie gluosnius ir kartu NĖRA informacijos apie tuopas, t. y., tinklalapius, kuriuose neminimi nei gluosniai, nei tuopos. Tai tikrai nėra teisinga. Tai visiškai priešinga tam, ko ieško bebras. Tačiau „priešingai“ reiškia „[...]“. Taigi atsakymas A laužtiniuose skliaustuose yra būtent tai, ko ieškome, ir tai yra atsakymas C.

D užklausa prasideda nuo C užklauskos, tačiau pridedamas papildomas reikalavimas: [gluosnis # tuopa]. Užklausa [gluosnis # tuopa] nurodo, kad tinklalapiai, kuriuose yra informacijos ir apie gluosnius, ir apie tuopas, NEGALI būti pateikiami. Kitaip tariant: atsakymas D rodo tai, ko ieško bebras, išskyrus tinklalapius, kuriuose yra informacijos ir apie gluosnius, ir apie tuopas. Taigi D užklausa atves į tinklalapius, kuriuose yra informacijos arba apie gluosnius, arba apie tuopas, bet ne apie abu.

Kad suprastume, ko ieško užklausa B, padalijame ją į dvi dalis. Pirmoji dalis [[gluosnis] # tuopa] nurodo, kad nenorime tinklalapių, kuriuose rašoma apie tuopas, bet ne apie gluosnius, kitaip tariant, jei tinklalapis yra apie tuopas, jis turi būti ir apie gluosnius. Panašiai ir antroji dalis [gluosnis # [tuopa]] nurodo, kad jei tinklalapyje yra informacijos apie gluosnius, jame taip pat turi būti informacijos apie tuopas. Sujungus abu reikalavimus (= užklausa B), gaunami tinklalapiai, kuriuose turi būti informacija ir apie gluosnius, ir apie tuopas, arba nė apie vieną iš jų.

Tai informatika!

Kaip ir užduotyje minimoje paieškos programėlėje, taip ir šiuolaikinėse paieškos sistemose užklauskas galima jungti naudojant logines operacijas NE, IR, ARBA. Bebras iš tiesų norėtų ieškoti pagal užklauską „gluosnis ARBA tuopa“, deja, jo programėlėje nėra loginę operaciją ARBA atitinkančio ženklo. Šiame uždavinyje parodoma, kad operaciją ARBA galima pakeisti loginių operacijų NE ir IR deriniu:

$$a \text{ ARBA } b = \text{NE} (\text{NE}(a) \text{ IR } \text{NE}(b))$$

Logikoje tokią taisyklę aprašo dualioji funkcija, dar žinoma De Morgano dėsnio vardu. Programavime naudojama daug loginių reiškinų. Išmanant keletą pagrindinių logikos taisyklių (tokių, kaip dualioji funkcija), kur kas lengviau užrašyti ir suprasti sudėtingas programas. Loginę operaciją ARBA reikėtų vertinti atsargiai – kartais ji gali turėti skirtingas reikšmes.

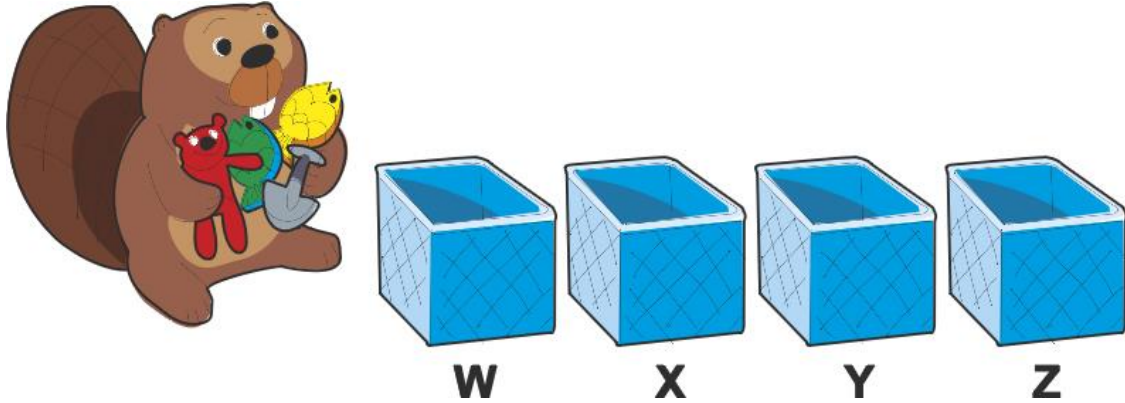
Viename iš neteisingų atsakymo variantų – D – aprašomas vadinamasis griežtasis ARBA: tokia užklausa randami tinklalapiai, kurių kiekvienas yra arba apie gluosnius, arba apie tuopas, bet ne apie abu medžius. Tokių operaciją ARBA dažniausiai naudojame kasdieniame gyvenime: kai kas nors siūlo mums picos arba mėsinį, paprastai juk nesakome, kad norime abiejų.

C variantas aprašo įprastą loginę operaciją ARBA – šios užklauskos rezultatas yra tinklalapiai, kurių kiekviename gali būti informacijos arba apie gluosnius, arba apie tuopas, arba apie abu medžius.

Gali pasirodyti keista, kad paieškos programėlėje operacijai IR naudojamas ženklas #. Galima būtų tikėtis žodelio IR arba ženklo &. Lygiai tas pats ir dėl laužtinių skliaustų neigimui žymėti. Vis dėlto, kompiuteriui nėra jokios reikšmės, kokie ženklai naudojami operacijoms žymėti. Kompiuteris tiesiog daro tai, ką yra užprogramuotas daryti. Dažniausiai programuotojai žymėjimui parenka tokius ženklus, kuriuos žmogui lengva perskaityti ir suprasti. O kompiuteriui nėra jokio skirtumo – # yra visiškai toks pat ženklas, kaip ir bet kuris kitas.

14. Triguba bėda

Mama bebrė turi keturis žaislus ir norėtų juos sudėti į keturias dėžes, pažymėtas raidėmis W, X, Y ir Z. Į kiekvieną dėžę ji gali įdėti ne daugiau nei vieną žaislą.



Mama bebrė nori patikrinti visas šias sąlygas:

1. Bent vienas iš teiginių yra teisingas:
dėžėje X yra žaislas arba dėžėje Y nėra žaislo, arba dėžėje Z nėra žaislo.
2. Bent vienas iš teiginių yra teisingas:
dėžėje W yra žaislas arba dėžėje X yra žaislas, arba dėžėje Z nėra žaislo.
3. Bent vienas iš teiginių yra teisingas:
dėžėje X nėra žaislo arba dėžėje Y nėra žaislo, arba dėžėje Z yra žaislas.
4. Bent vienas iš teiginių yra teisingas:
dėžėje W nėra žaislo arba dėžėje X nėra žaislo, arba dėžėje Y nėra žaislo.
5. Bent vienas iš teiginių yra teisingas:
dėžėje X nėra žaislo arba dėžėje Y yra žaislas, arba dėžėje Z nėra žaislo.

Kiek daugiausiai žaislų mama bebrė gali sudėti į dėžes taip, kad visos penkios išvardintos sąlygos būtų tenkinamos?

- A. 1
- B. 2
- C. 3
- D. 4

Paaiškinimas

Teisingas atsakymas: 3.

Į dėžes galima įdėti ne daugiau kaip 3 žaislus. Atkreipkite dėmesį, kad jei mama bebrė įdės žaislą į visas dėžes, 4 sąlyga bus neįvykdyta.

Visose sąlygose yra trys taisyklės, sujungtos žodžiu ARBA. Vadinasi, turi būti tenkinama bent viena iš taisyklių. Sistemingai patikrinti visas sąlygas lengviau, jei jas surašysime į lentelę, kurioje 1 reiškia, kad „dėžutėje yra žaislas“, o 0 – „dėžutėje nėra žaislo“.

Sąlyga	W	X	Y	Z
1	-	1	0	0
2	1	1	-	0
3	-	0	0	1
4	0	0	0	-
5	-	0	1	0

Tada galime patikrinti visas galimas kombinacijas, pradedant nuo tų, kuriose bendras žaislų skaičius yra didžiausias (ieškome didžiausio žaislų skaičiaus), ir patikrinti, ar jos atitinka visas sąlygas.

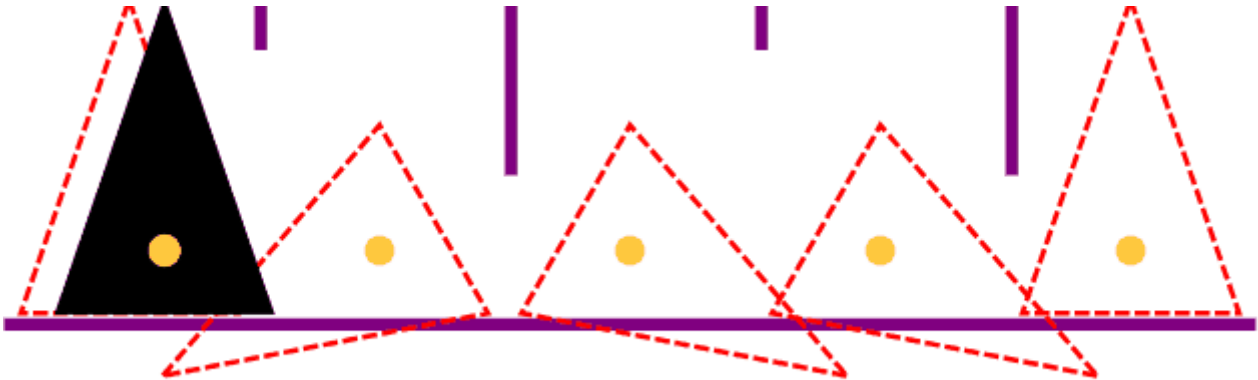
Tam tikra dėžučių kombinacija tenkina sąlygą, kai kiekvienos sąlygos atveju bent vienas skaičius atitinka vieną iš taisyklių (kadangi kiekvienos sąlygos trys taisyklės sujungiamos loginiu ARBA, sąlyga yra klaidinga tik tada, kai pažeidžiamos visos taisyklės), o priešingu atveju ji pažeidžiama. Toliau pateiktoje lentelėje rasime pirmąją neįvykdytą sąlygą, raudonai pažymėti skaičiai, neatitinkantys pažeistos taisyklės. Pavyzdžiui, kombinacija *b* pažeidžia 1 taisyklę, nes visi X, Y, Z langeliai neatitinka taisyklės. Kombinacija *e* nepažeidžia nė vienos taisyklės, todėl didžiausias žaislų skaičius, kurį mama bebrė gali įdėti į dėžes, yra trys.

Kombinacija	W	X	Y	Z	Pirmoji pažeista sąlyga	Dėžės su žaislais
a	1	1	1	1	4	(4)
b	1	0	1	1	1	(3)
c	1	1	0	1	5	(3)
d	1	1	1	0	3	(3)
e	0	1	1	1	-	(3)

Tai informatika!

Loginiai teiginiai informatikoje dažnai naudojami. Tokio tipo uždavinys, kai reikėtų rasti rinkinį, tenkinantį ribojimus, deja, bendru atveju neturi efektyvaus sprendimo. Šis uždavinys žinomas kaip konjunktyvios normaliosios formos uždavinys (angl. *Boolean satisfiability problem*, vartojama santrumpa SAT). Trys teiginiai kiekvienoje sąlygoje (3-SAT) komplikuoja uždavinio sprendimą, kai sąlygų skaičius didėja. Čia turėjome vos penkias sąlygas, tačiau naudota strategija reikalavo patikrinti jas visas kaskart, kai tik pakeisdavome rinkinį. Informatikoje yra geresnių strategijų, tačiau jos visos gana neefektyvios. Netgi organizuojamos SAT varžybos, kuriose programuotojai mėgina parašyti greičiausių SAT sprendimo programą. Kadangi daugybė problemų gali būti transformuotos į SAT uždavinius, greitos SAT sprendimo programos gali būti naudingos įvairiose situacijose.

15. Sukis, sukis, sukis!



Trikampis juda konvejerio juosta ir turi savo sukimosi centrą. Dvi trikampio viršūnės yra arčiau centro, o trečioji viršūnė kiek nutolusi. Sukimosi centras pasirenkamas taip, kad pasukus trikampį 120° , viena iš viršūnių visada būtų tiesiai virš sukimosi centro.

Virš konvejerio juostos yra strypelių pavidalo rodyklės. Kai trikampis praeina rodyklę, jis gali pasisukti 120° prieš laikrodžio rodyklę. Turimi du tipai rodyklių:

- ilgoji rodyklė, kuri visada pasuka trikampį;
- trumpoji rodyklė, kuri pasuka trikampį tik tada, jeigu viršūnė, esanti toliausiai nuo centro, yra viršuje.

Bobas kuria procesą, kuriam naudoja trikampius, kaip pavaizduota paveikslėlyje. Jis prašo jam padėti surasti rodyklių seką, kuri trikampius iš bet kokios pradinės padėties pasuktų taip, kad proceso pabaigoje jie visada būtų vienodoje padėtyje. Kuri tai padėtis Bobui nėra svarbu.

Spustelėkite ir pakeiskite rodyklių ilgius taip, kad jų seka atitiktų Bobo norus.



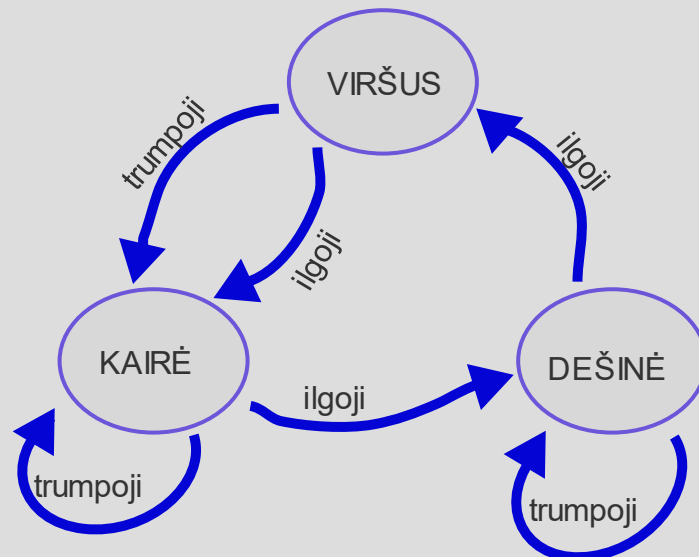
Paaiškinimas

Teisingas atsakymas:



Tai informatika!

Šis uždavinys yra susijęs su baigtiniu automatu, pavaizduotu tokia schema:



Baigtinio automato būsenos vaizduojamos grafo viršūnėmis, o perėjimai tarp jų – lankais.

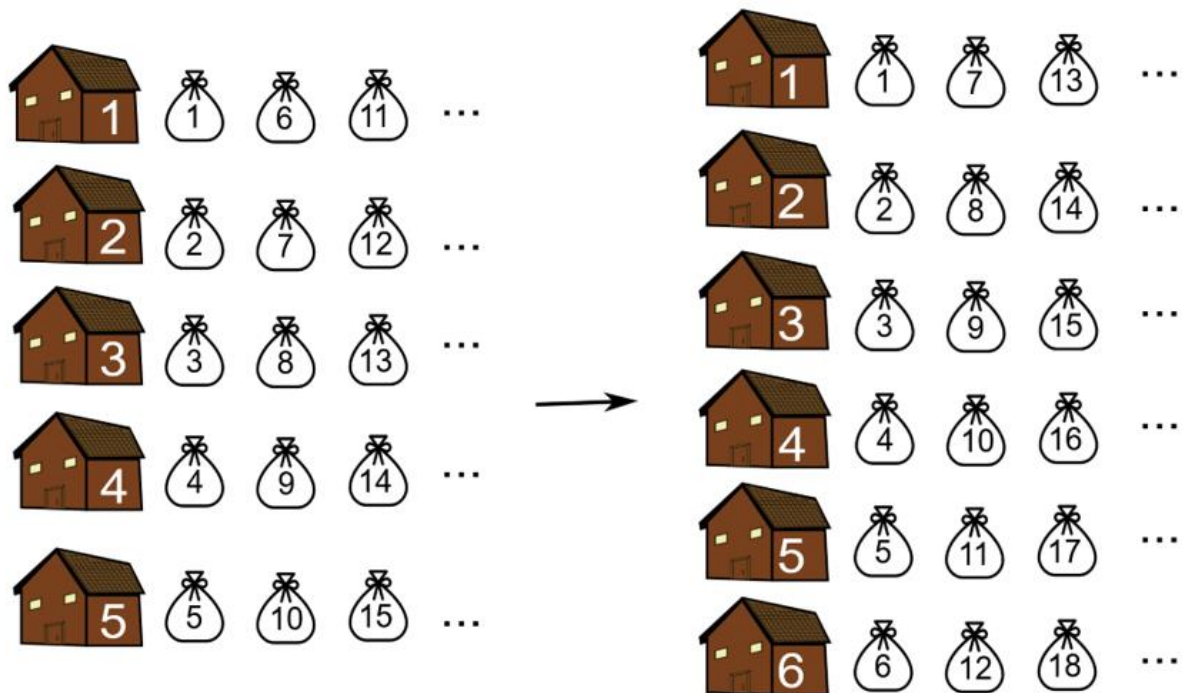
Plačiau: https://en.wikipedia.org/wiki/Synchronizing_word

16. Sandėliai

42 ežiai laiko savo atsargas 5 sandėliuose. Pirmas ežys savo atsargas deda į pirmą sandėlį, antras – į antrą, ..., šeštas – į pirmą ir taip toliau.

Vieną dieną ežiai įrengė dar vieną – šeštą – sandėlį ir nusprendė sudėti atsargas taip, kad pasiskirstymas sandėliais vėl būtų labai paprastas: pirmas ežys laiko atsargas pirmame sandėlyje, antras – antrame sandėlyje, ..., septintas – pirmame sandėlyje ir taip toliau.

Dalis ežių turės pernešti savo atsargas į kitą sandėlį. Pavyzdžiui, pirmas ežys visada laiko atsargas pirmame sandėlyje, taigi jis neturi nieko perkelti. Tačiau šeštas ežys anksčiau laikė atsargas pirmame sandėlyje, o dabar turi perkelti į šeštą.



Keliems ežiams **neriekės** perkelti savo atsargų į kitus sandėlius?

Paaiškinimas

Teisingas atsakymas: 10.

1, 2, 3, 4, 5 ir 31, 32, 33, 34, 35 ežiukai neturi pernešti savo atsargų.

Norėdami tuo įsitikinti, pirmiausia galime supaprastinti numeraciją: tegul ežiai ir sandėliai būna sunumeruoti nuo 0. Taigi turime ežius nuo 0 iki 41. Ežys, kurio numeris x , neturėtų judėti, jei $x \bmod 5 = x \bmod 6$. Tai įmanoma, jei $30 \mid x - r$, čia $r = x \bmod 5 = x \bmod 6$. Taigi tai įmanoma tik esant $x = 0 + 0, 1, 2, 3, 4$ ir $x = 30 + 0, 1, 2, 3, 4$.

Tai informatika!

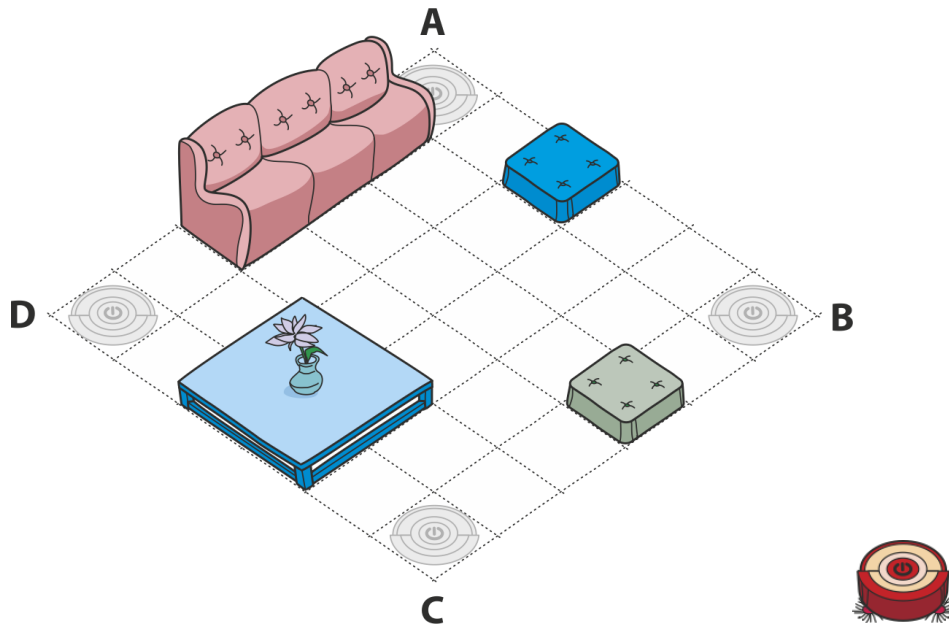
Ši užduotis susijusi su paskirstytu išteklių saugojimu: norint juos atkurti, galima naudoti strategiją, vadinamą maiša (angl. *hash*). Taikant šį metodą kiekvienas išteklius susiejamas su maišos raktu, kurį galima efektyviai apskaičiuoti; tada ištekliai ieškomi apskaičiuojant maišos raktą ir ieškant tik tarp tų, kurių maišos raktai sutampa. Pavyzdžiui, jei ištekliai yra asmenys ir jų vardai, taikant maišos strategiją kaip maišos raktą galima naudoti pirmąją vardo raidę: jei reikia rasti asmenį „Matas“, maišos raktas bus „M“; norint jį rasti, galima ieškoti tik tarp mažesnės grupės asmenų, turinčių tą patį maišos raktą (pavyzdžiui, „Matas“, „Matilda“ ir „Marta“). Užduotyje maišos raktas apskaičiuojamas naudojant mod funkciją (sveikąjo skaičiaus dalybos likutį). Padidėjus sandėlių skaičiui raktas pasikeičia. Sakoma, kad geros maišos funkcijos, turinčios savybę nekeisti rakto, kai kas nors pasikeičia duomenų saugyklos struktūroje, turi nuoseklios maišos savybę.

17. Robotas-siurblys

Robotas valo kvadratinėmis plytelėmis išklotas grindis pagal šias komandas:

- P – paeina viena plytele į priekį (sugaišta 1 minutę),
- D – pasisuka 90° dešinėn (tai daro akimirksniu),
- V – išvalo plytelę (sugaišta 1 minutę).

Robotas pradeda ir baigia vienoje iš kampinių plytelių (A, B, C, D), bet nebūtinai toje pačioje.



Kiek mažiausiai minučių sugaišta robotas, kol išvalo visas prieinamas grindų plyteles?

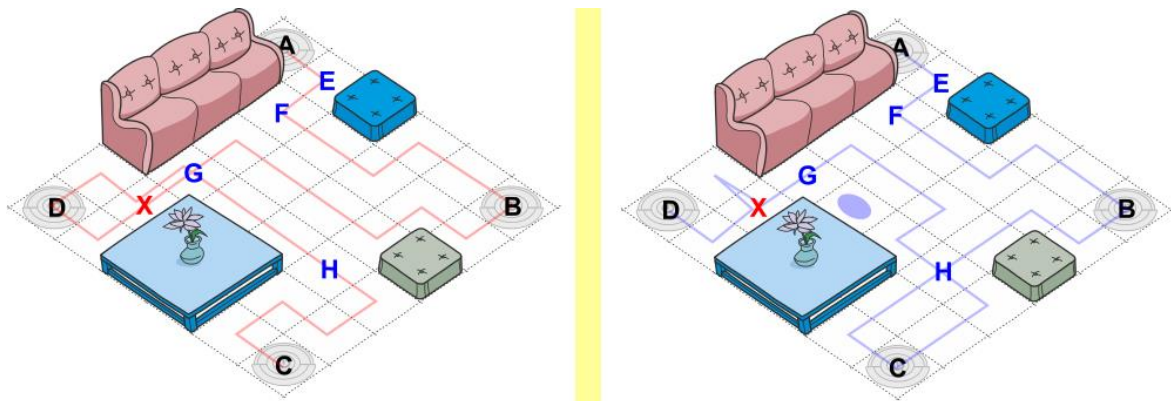
Paaiškinimas

Teisingas atsakymas: 55.

Iš viso yra $36 - 9 = 27$ grindų plytelės, taigi joms išvalyti reikia 27 minučių. Jei kiekvieną plytelę pavyktų pereiti tik vieną kartą, tai siurblys eidamas sugaištų 26 minutes (pirma plytelė neskaičiuojama, nes joje jau stovima). Taigi reikia mažinti skaičių plytelių, per kurias reikėtų eiti antrą kartą.

Panagrinėkime kairįjį paveikslą. Plytelė X yra išskirtinė – ją teks pereiti du kartus, kad ir kurį kelią pasirinktume. Plytelės, per kurias patenkama į A, B, C ir D kampus, pažymėtos atitinkamai raidėmis E, F, G ir H. Per E ir F plyteles įeinama į A kampą, per G plytelę – į D kampą, o per H plytelę – į C kampą. B kampas neturi tokios „įleidžiančios“ plytelės. Jei kuris kampas nėra nei pradžia, nei pabaiga, robotui tenka į šiuos kampus „įleidžiančias“ plyteles pereiti du kartus. Jei robotas pradeda ir baigia tame pačiame kampe, tai abi plyteles, per kurias patenka į kampus, turi pereiti du kartus. Jei robotas pradeda viename kampe, o baigia kitame, tai į juos „įleidžiančias“ plyteles gali pereiti po vieną kartą. Visa tai žinodami, galime suplanuoti roboto kelią taip, kad jam reikėtų pereiti kuo mažiau plytelių, per kurias patenkama į kampus.

Kadangi A kampe yra dvi tokios plytelės – E ir F, tai nuo šio kampe ir geriausia pradėti (arba jame baigti). Toliau tenka pasirinkti antrą – pabaigos – kampą – C arba D (abiem atvejais vieną iš plytelių, per kurias įeinama, t. y., G arba H, pereitume tik vieną kartą).



Kairiajame paveiksle parodytas kelias iš A į C, kurį sudaro 28 ėjimai į priekį (jame yra išskirtinės plytelės X ir G). Keliui iš A į D reikėtų nelyginio skaičiaus ėjimų – padarius 28 ėjimus į priekį viena plytelė liktų nevalyta (tai būtų plytelė, pažymėta mėlyna dėme dešiniajame paveiksle). Taigi šis kelias yra ilgesnis nei iš A į C.

Mažiausias plytelių valymo laikas yra: 27 (valyti) + 28 (judėti) min. = 55 min.

Tai informatika!

Ši užduotis yra vienas iš keliaujančio pirklio uždavinio atvejų: norint aplankyti visas grafo viršūnes, reikia rasti trumpiausią kelią. Šį uždavinį galima traktuoti ir kitaip: tai – išplėsta Hamiltono kelio problema, kai reikia nustatyti, ar galimas kelias per visas grafo viršūnes, aplankant jas po vieną kartą. Deja, nė vienam iš šių uždavinių išspręsti nėra jokio optimalaus algoritmo. Jei plytelių būtų daug, tai net galingi kompiuteriai nesusidorotų su skaičiavimais per žmogui priimtina laiką.

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

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



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



**Vilniaus
universitetas**

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

„Bebro“ konkurso uždavinių rinkinys tinka Mokyklos pedagogikos studijų programos moduliui „Informatikos didaktika“. Studentai, būsimi mokytojai, nagrinėdami „Bebro“ uždavinius susipažįsta su įvairiais informatikos konceptais, gilinasi į sudėtingesnius informatikos konceptus ir išmoksta juos paaiškinti.