ISSN 2783-7599

b·e·b·r·a·s

Informatikos ir informatinio mąstymo uždavinių rinkinys

Nr. 1

ŠVIETIMO, MOKSLO IR SPORTO MINISTERIJA

VILNIAUS UNIVERSITETAS · 1579 · UNIVERSITAS VILNENSIS

Bebras
International Challenge on Informatics and Computational Thinking

Šiame rinkinyje pateikiami XII 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.

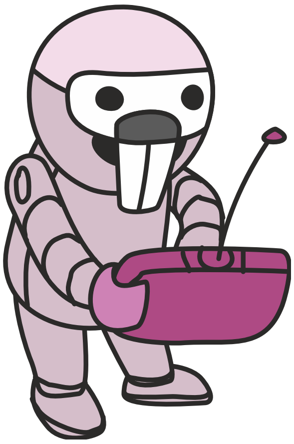
Dėkojame Daumilui Ardickui, dr. Gintautui Grigui, Audrai Ivanauskienei, dr. Eglei Jasutei, dr. Tatjanai Jevsikovai, Alvidai Lozdienei, dr. Gabrielei Stupurienei, talkinusiems verčiant ir adaptuojant uždavinius. Taip pat dėkojame tarptautinei „Bebro“ bendruomenei ir uždavinių autoriams.

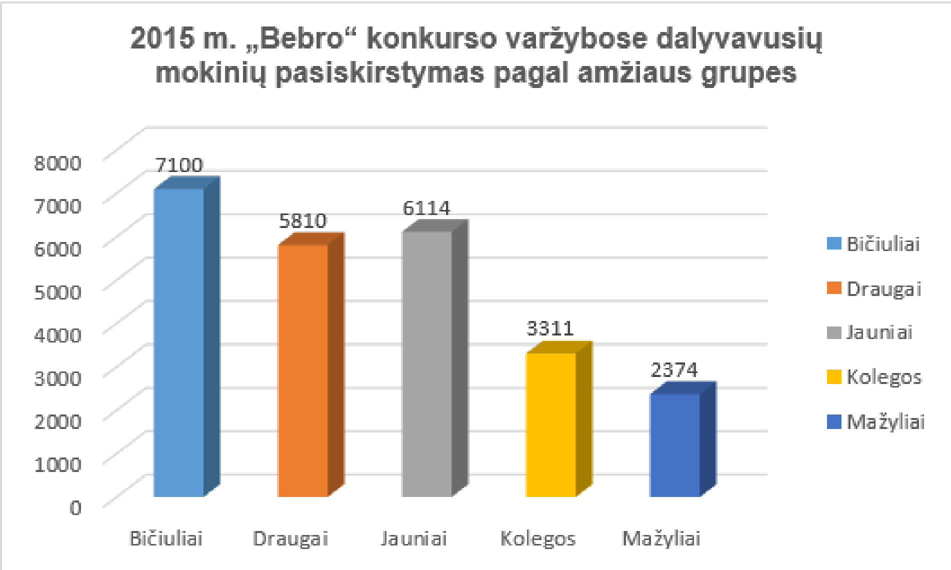
Parengė Lina Vinikienė
Konsultavo Valentina Dagienė
Redagavo Viktoras Dagys
Viršelį kūrė Vaidotas Kinčius

Įvadas

„Bebro“ konkurso tikslas – sudominti bendrojo ugdymo mokyklų mokinius informatikos mokslo fundamentinėmis idėjomis, sumaniau naudotis informacinėmis technologijomis, ugdyti mokinių kūrybiškumą, informacinę kultūrą, algoritminį, loginį, kritinį ir informatinį mąstymą. Konkurso uždaviniai trumpi, iliustruoti įdomiais paveikslėliais ar schemomis, interaktyvūs, vidutiniškai išsprendžiami per kelias minutes. Sprendžiant uždavinius mokiniai skatinami įžvelgti problemą, kelti naujas idėjas, ieškoti originalių sprendimo būdų.

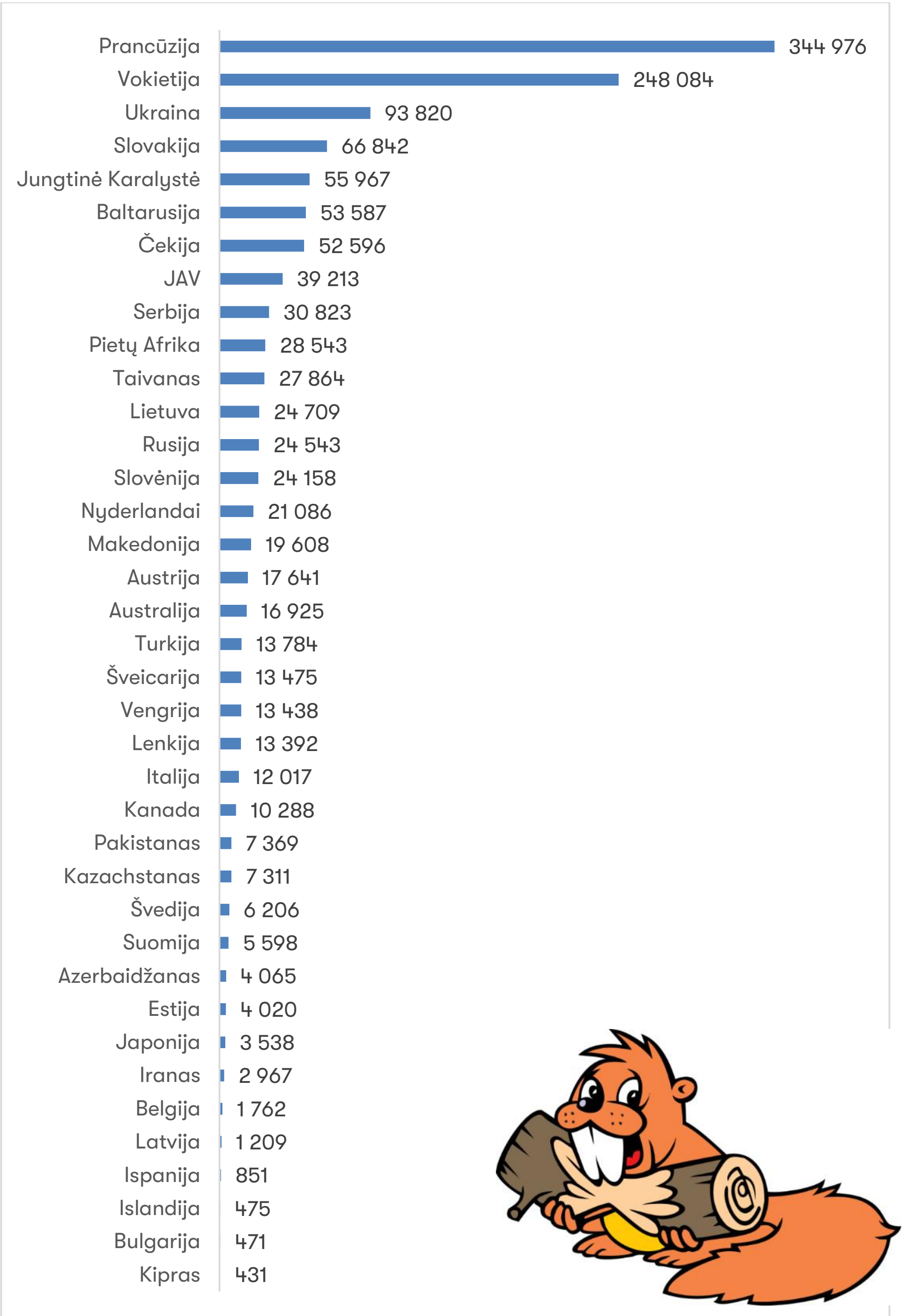
2015 metų XII „Bebro“ konkurso I etapo uždavinius pasaulyje sprendė virš 1 313 650 mokinių (duomenis pateikė 38 valstybės). Lietuvoje tais metais dalyvavo 24 709 mokiniai.





2015–2016 mokslo metais antrasis etapas vyko 2016 m. sausio 31 d. Į konkursą galėjo registruotis tik 9–12 klasių mokiniai, pirmajame etape surinkę ne mažiau kaip 116 taškų. Iš viso užsiregistravo 704 mokiniai. Konkursas vyko Lietuvos aukštosiose mokyklose: Kauno technologijos, Klaipėdos, Vilniaus universitetuose, Alytaus, Panevėžio, Utenos kolegijose. Į renginį atvyko ir uždavinius sprendė 556 mokiniai: 9–10 klasių – 269, 11–12 klasių – 287 mokiniai. Dalyviai turėjo išspręsti 15 uždavinių, tam buvo skirta 30 min. Pradėdami spręsti mokiniai turėjo po 45 taškus.

Kiekvienoje aukštojoje mokykloje mokiniai buvo pakviesti ne tik dalyvauti konkurse (spręsti uždavinius), bet ir dalyvauti veiklose, kurių tikslas supažindinti mokinius su aukštųjų mokyklų veikla, sudominti informatikos srities studijomis. Moksleiviai bei juos atlydėję mokytojai ar tėveliai stebėjo jaunųjų informatikų akademijos pristatymą, lazerių šou (Alytaus kolegijoje), lankėsi KTU santakos slėnyje (KTU Informatikos fakultete), konstravo prototipinius robotukus (KTU Panevėžio technologijų ir verslo fakultete), dalyvavo robotų sumo ir krepšinio varžybose su „Lego Mindstorms EV3“ komplektais (Klaipėdos universitete), lankėsi praktinio mokymo laboratorijose (Panevėžio kolegijoje), klausėsi paskaitų apie Miobijaus juostos ypatumus, stebėjo trimačio spausdintuvo galimybes (Vilniaus universiteto Šiaulių akademijoje). Taigi, kol vieni mokiniai dalyvavo konkurse, kitiems neteko nuobodžiauti.



2015 m. „Bebro“ dalyvių skaičius pagal šalis (38 šalių duomenys)

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

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

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

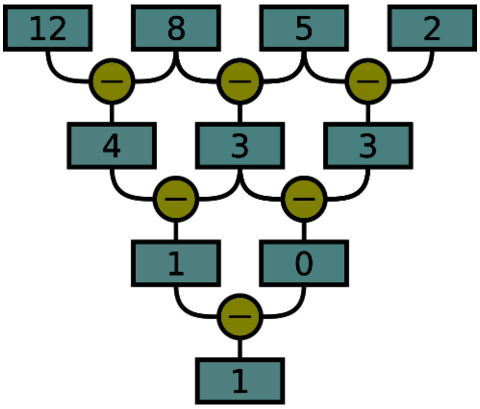
Taip pat pateikiamas uždavinio atsakymas ir paaiškinimas, kaip uždavinys susijęs su informatika.

Nr.	Pavadinimas	Uždavinio identifikatorius	Jauniai (9–10 kl.)	Kolegos (11–12 kl.)
1	Skaičių skirtumų automatas	2015-BG-02	6	
2	Šuoliai	2015-CA-05	6	
3	Kelionė	2015-CH-15	6	
4	Orų prognozė	2015-IT-02	6	
5	Telefonų knyga	2015-AT-07	6	
6	Rąstai	2015-TW-02	9	6
7	Sūris	2015-CH-05	9	
8	Labirintas	2015-CH-09	9	
9	Keleivių tarpusavio ryšiai	2015-CZ-02	9	
10	Robotas spalvintojas	2015-NL-01	9	
11	Akmenukų žaidimas	2015-IT-01 A	12	9
12	Tobulasis grafas	2015-PL-04	12	9
13	Svarbiausioji sekos dalis	2015-DE-01	12	
14	Sunumeruotos kortelės	2015-NL-05	12	
15	Paštas	2015-US-01	12	
16	Pasivyk	2015-CA-03		6
17	Akrobatai	2015-HU-05		6
18	Medžio dirbiniai	2015-JP-01		6
19	Vitražas	2015-AZ-02		6
20	Bebro trobelių jungimas	2015-CH-08		9
21	Klaidų paieška	2015-FR-08		9
22	Draugystė	2014-TW-04		9
23	Akmenukų žaidimas	2015-IT-01 C		12
24	Šokinėjanti kengūra	2015-RU-08		12
25	Akmenukai	2015-TW-04		12
26	Stebuklingas įrenginys	2013-BE-07		12
27	Šaudymas į taikinį	2015-BE-05		12

1. Skaičių skirtumų automatas

Rūta suprogramavo automatą, į kurio pirmąją eilutę įvedami 4 skaičiai, o kiekvienoje tolesnėje eilutėje skaičiuojami virš jos esančios eilutės dviejų gretimų skaičių skirtumai.

Pavyzdžiui, įvedame skaičius 12, 8, 5, 2 ir gauname:



Kurią skaičių seką įvedus į šį automatą, rezultatas, esantis ketvirtoje eilutėje, lygus nuliui?

A. 13 9 7 6

B. 16 9 4 1

C. 13 8 4 2

D. 5 5 5 1

Paaiškinimas

Teisingas atsakymas – B.

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 akivaizdu tik B atveju).

Tai informatika!

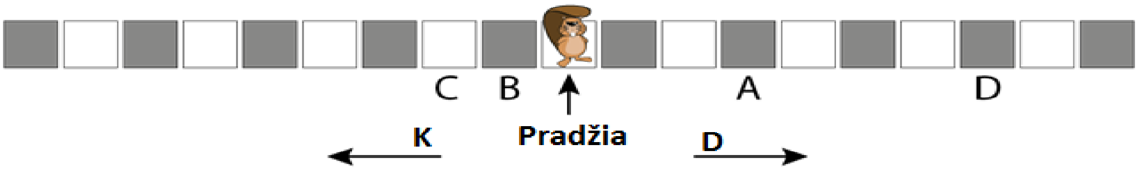
Šis skirtumų skaičiavimo automatas atlieka paprastus skaičiavimus, šiek tiek panašius į vykdomus kompiuterio mikroschemoje. Jame gaunamos informacijos apdorojimo procesas išskaidomas į mažesnius žingsnius ir tik tada gaunamas rezultatas. Beje, tokiu metodu skaičiuoti skirtumus nesunku ir su skaičiuokle.

Reikšminiai žodžiai

Algoritmai, modeliavimas, skaičiuoklė, skaitiniai metodai

2. Šuoliai

Bebras Faustas keistai šokinėja. Jis pradeda nuo vidurinio langelio ir šoka penkis kartus, kaskart keisdamas kryptį iš dešinės (D) į kairę (K) arba iš kairės į dešinę: pirmiausia šoka į dešinę, tada į kairę, vėl į dešinę, tada į kairę ir galų gale į dešinę.



Bebras gali pasirinkti, į kelintą langelį šokti: 1, 2, 3, 4 ar 5 – kaskart vis skirtingą skaičių. Pavyzdžiui, gali šuoliuoti taip: dešinėn 2, kairėn 1, dešinėn 5, kairėn 4 ir dešinėn 3 ir atsidurti 2 – 1 + 5–4 + 3 = 5 langelyje dešiniau nuo langelio, kuriame pradėjo.

Kas antras langelis nuspalvintas pilkai – lengviau skaičiuoti.

Keturi langeliai pažymėti raidėmis A, B, C ir D. **Į kurį iš šių langelių bebras niekada nepateks, šokinėdamas pagal aprašytą taisyklę?**

(A) Nepateks į A

(B) Nepateks į B

(C) Nepateks į C

(D) Nepateks į D

Paaiškinimas

Teisingas atsakymas –C.

Pažymėkime ėjimus į dešinę teigiamais skaičiais, o į kairę – neigiamais. Tada:

į A langelį galima ateiti einant 5–4 + 3–2 + 1 = 3,

į B langelį galima ateiti einant 1–5 + 2–3 + 4 = -1,

į D langelį galima ateiti einant 5–3 + 4–1 + 2 = 7.

Negalima patekti į C langelį, nes bebras gali eiti 3 kartus nelyginį skaičių langelių ir 2 kartus lyginį skaičių langelių. Kai su šiais skaičiais atliekamos atimties ir sudėties operacijos, rezultatas visada yra nelyginis skaičius, o langelis C yra antrasis langelis nuo bebro ėjimo pradžios (lyginis). Todėl bebras, eidamas pagal aprašytą taisyklę, niekada į jį neįšoks.

Tai informatika!

Ši užduotis remiasi abstrakcija ir algoritmu. Šiuo atveju ėjimų seka yra algoritmas, kuriame pirmojo žingsnio pasirinkimas nulemia kitą žingsnį (-ius) ir šie ėjimai yra atliekami tam tikra tvarka.

Abstrakcijos konceptas pasireiškia, kai, užuot galvojus apie ėjimą kaip patekimą į tam tikrą vietą, problema supaprastinama, ėjimus susiejant su teigiamais ir neigiamais skaičiais ir iš to nustatant įmanomą ar neįmanomą rezultatą.

Nors užduotis iš pirmo žvilgsnio atrodo labai lengva, ji tokia nėra. Įsivaizduokite šiek tiek kitą užduotį, kurioje bebrui yra leidžiama rinktis langelių skaičių (iš tam tikros skaičių aibės) ir kryptį (nebūtinai iš eilės į dešinę ir kairę). Ši problema yra žinoma kaip labai sunki (informatikoje vadinama „NP sudėtingumo“) ir praktiškai neišsprendžiama šiuolaikiniais kompiuteriais, kai galimas didelis ėjimų skaičius.

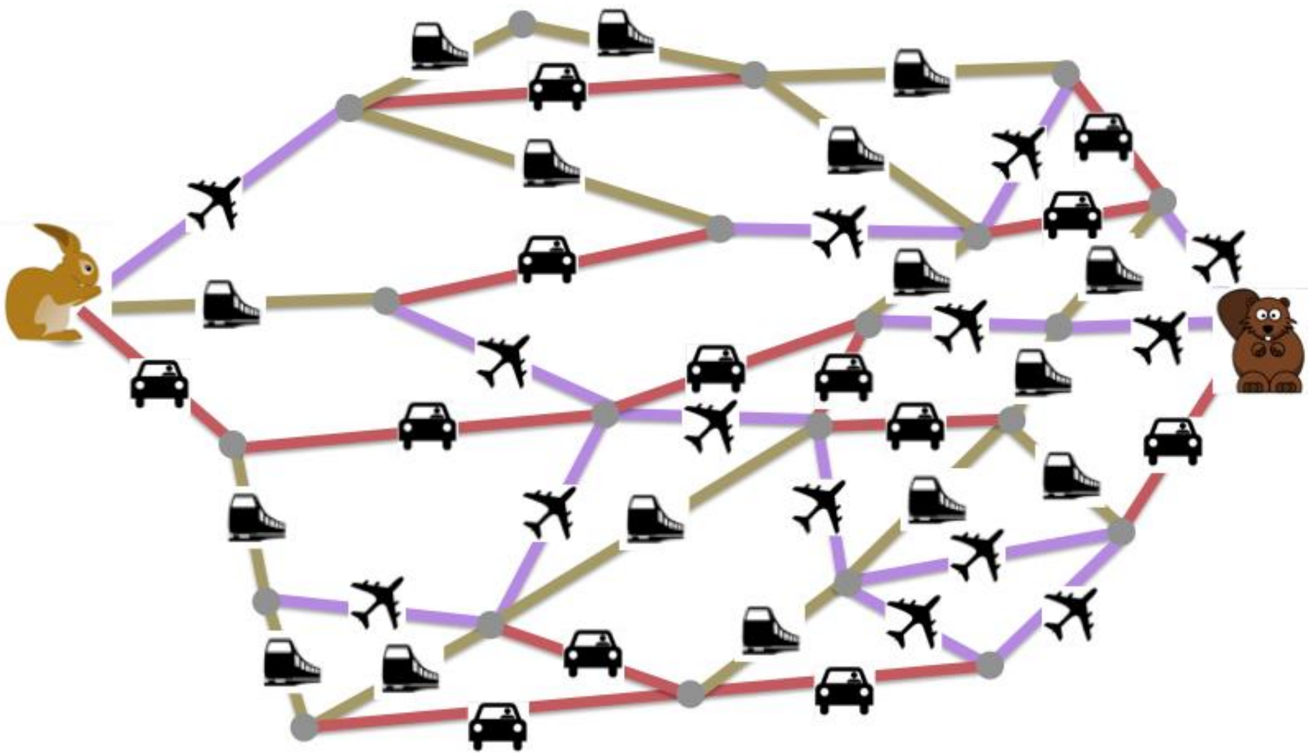
Reikšminiai žodžiai

Algoritmas, abstrakcija

3. Kelionė

Triušis Robertas nori aplankyti savo draugą bebrą Bronių. Žemėlapyje pavaizduoti keliai, kuriais gali keliauti triušis Robertas.

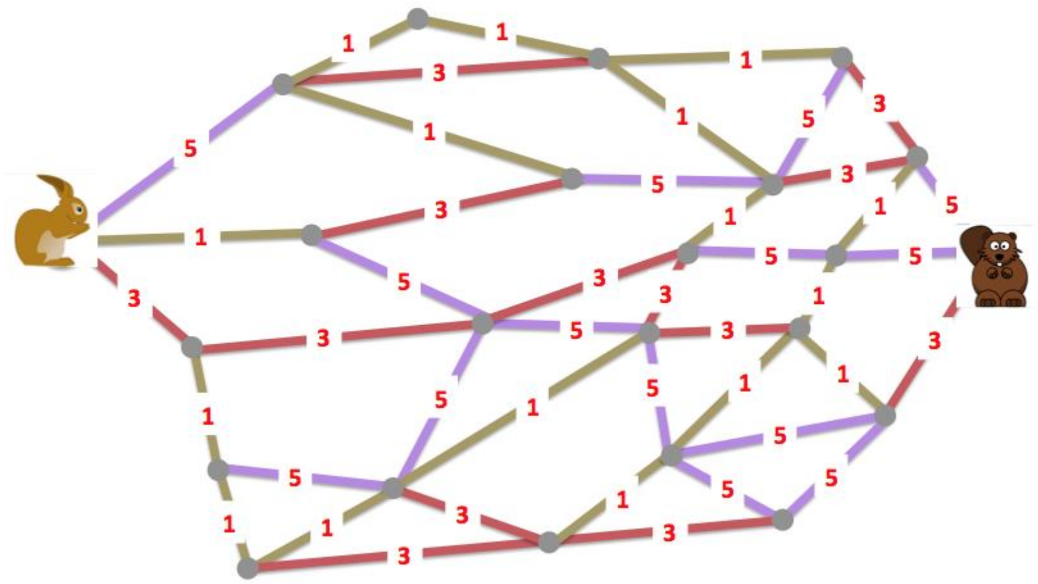
Transporto priemonės pažymėtos skirtingais simboliais. Kelionė matuojama ekologiniais taškais. Skrendant lėktuvu, kelionė nuo vieno taško iki kito yra 5 ekologiniai taškai, važiuojant automobiliu – 3 ekologiniai taškai, važiuojant traukiniu – 1 ekologinis taškas.



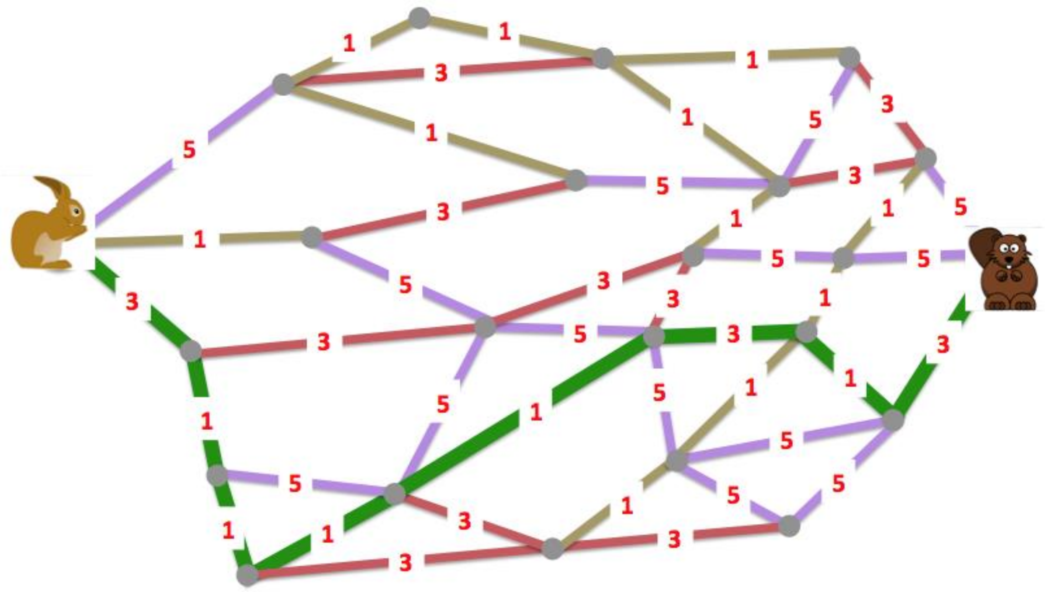
Pažymėk mažiausiai ekologinių taškų kainuojantį kelią, kurį Robertas turi pasirinkti, norėdamas aplankyti Bronių.

Paaiškinimas

Paveikslėlyje surašyti ekologiniai taškai, reikalingi, norint pasiekti tam tikrą kelionės tašką.



Antrame paveikslėlyje paryškintas kelias, kainuojantis mažiausiai ekologinių taškų. Optimalus kelias kainuoja 14 taškų.



Tai informatika!

Šis uždavinys yra trumpiausio kelio nuo vieno taško iki kitų radimas. Tačiau kelias matuojamas ne laiku ar atstumu, bet ekologiniais taškais. Žinomi keli metodai, kuriais sprendžiamas šis uždavinys, labiausiai žinomas Dijkstros algoritmas ir paieškos algoritmas.

Reikšminiai žodžiai

Trumpiausias kelias, Dijkstros algoritmas, kelionės kaina

4. Orų prognozė

Bebras Jonas planuoja rytoj keliauti į paplūdimį. Jis eis tik tada, jei bent tris valandas tarp 13.00 ir 19.00 švies saulė. 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, debesys, lietus arba 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 surašyti šias komandas į seką. Pirmosios komandos įvestis visada yra orų prognozės failo turinys, o kiekvienos kitos komandos įvestis yra prieš tai esančios komandos rezultatas.

Kuri komandų seka gali padėti bebrui Jonui apsispręsti, ar eiti į paplūdimį?

A. PIRMOSIOS 19 | PASKUTINIOSIOS 6 | YRA saulė | SKAIČIUOTI

B. YRA saulė | PIRMOSIOS 19 | PASKUTINIOSIOS 6 | SKAIČIUOTI

C. PIRMOSIOS 20 | PASKUTINIOSIOS 6 | YRA saulė | SKAIČIUOTI

D. PASKUTINIOSIOS 20 | PIRMOSIOS 6 | YRA saulė | SKAIČIUOTI

Paaiškinimas

Teisingas atsakymas – A.

Klausimas turi du galimus atsakymus:

1) PIRMOSIOS 19 | PASKUTINIOSIOS 6 | YRA saulė | SKAIČIUOTI

2) PASKUTINIOSIOS 11 | PIRMOSIOS 6 | YRA saulė | SKAIČIUOTI

Aptarkime atsakymą A.

PIRMOSIOS 19 | PASKUTINIOSIOS 6 | YRA saulė | SKAIČIUOTI

Pirmoji komanda išrenka failo eilutes nuo 00.00 – 01.00 iki 18.00 – 19.00. Antroji komanda iš šių eilučių išrenka eilutes nuo 13.00 – 14.00 iki 18.00 – 19.00. Trečioji komanda išrenka tik tas eilutes, kuriose yra žodis „saulė“, o paskutinė komanda suskaičiuoja, kiek tokių eilučių yra. Jei komandų sekos rezultatas yra trys arba daugiau, bebras eina į paplūdimį, priešingu atveju – neina.

Tai informatika!

Šioje užduotyje naudojama keletas informatikos konceptų: algoritmas, filtravimas, duomenų srautas. Kiekviena algoritmo komanda atrenka dalį įvedamų duomenų. Šioje užduotyje labiau sutelkiama į keletą paprastų veiksmų (atliekamų tam tikrus duomenis apdorojančiais žingsniais), nei į sudėtingesnę bendro pobūdžio problemą.

Reikšminiai žodžiai

Algoritmas, filtravimas, duomenų srautas

5. Telefonų knyga

Bebrė Lina ieško draugės britės telefono numerio labai ilgame tinklalapyje. Ji neprisimena tikslaus vardo ir pavardės, todėl paieškoje vartoja specialiuosius ženklus:

? – kai tiksliai nežinomas vienas simbolis,

& – kai tiksliai nežinomi du gretimi simboliai,

% – kai nežinoma žodžio pabaiga.

Pavyzdžiui, įvedus **The%**, paieškos rezultatai gali būti **Theresa**, **Theodor** ir pan.

Lina įveda **S?rah B&cht%**.

Kurį iš asmenvardžių pateiks paieškos sistema pagal šią Linos užklausą?

1. Sirah Birchman
2. Sara Bilchdrain
3. Sarah Birchtree
4. Sahrah Beachtram

Paaiškinimas

Teisingas atsakymas – 3. S**a**rah B**ir**cht**ree**.

Pirmasis neteisingas, nes pavardėje Birchman nėra raidės „t“.

Antrasis neteisingas, nes varde Sara trūksta raidės „h“.

Ketvirtasis neteisingas, nes varde Sahrah yra perteklinė antroji arba trečioji raidė (specialusis simbolis ? keičia tiksliai vieną raidę).

Tai informatika!

Dažnai tenka ieškoti duomenų fragmentų dideliuose duomenų masyvuose. Kartais negalime tiksliai pasakyti, ką norime rasti. Kaip matėme iš pateikto uždavinio, nežinomoms duomenų dalims galime vartoti pakaitos simbolius. Tuomet galime rasti mums reikalingus duomenis.

Galima ieškoti atskirame faile (pavyzdžiui, viename tinklalapyje), viename kompiuteryje (pavyzdžiui, visuose standžiojo disko failuose) arba daugelyje kompiuterių (pavyzdžiui, visame kompiuterių tinkle). Paieškos rezultatai sunkiau gaunami, kai vartotojui suteikiama daugiau galimybių formuojant užklausą ir kai ieškoma labai dideliuose failuose. Labai dideliems informacijoms kiekiams apdoroti reikalingi sudėtingi paieškos algoritmai, geriausia išeitis – duomenų bazė, kurioje ir pateikiamos užklausos.

Reikšminiai žodžiai

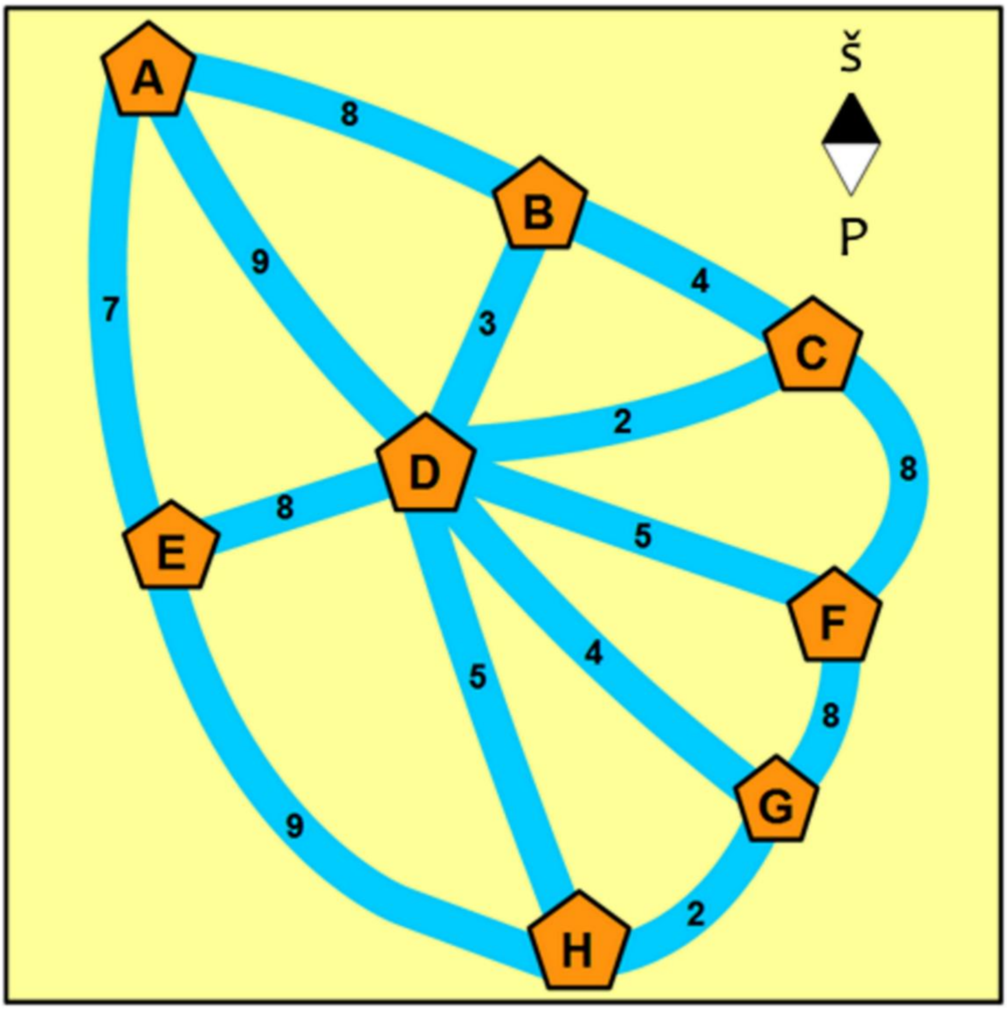
Pakaitos simbolis, duomenų bazė, paieška

6. Rąstai

Užtvankų statyba yra svarbiausias bebrų metų darbas. Miške jie prisirenka labai daug sunkių rąstų. Juos atsiplukdo upėmis pagal šias taisykles:

1. Bebrai surenka rąstus ant upės kranto ir sukrauna ant plausto.
2. Kiekvienoje upėje į plaustą galima krauti tik tam tikrą skaičių naujų rąstų.
3. Plaustai gali plaukti bet kuria upe tik pietų kryptimi.

Bebrams rūpi rasti geriausią maršrutą, kuriuo plaukdami surinktų daugiausiai rąstų.



Žemėlapyje pavaizduota upių sistema ir nurodytas galimas rąstų skaičius kiekvienoje upėje. Kompasas rodo šiaurės ir pietų kryptis.

Jei bebrai pradeda taške A ir nori pasiekti tašką H, kiek daugiausiai rąstų jie gali plukdyti vienu plaustu?

(A) 32

(B) 31

(C) 19

(D) 40

Paaiškinimas

Teisingas atsakymas – B (31).

Geriausias maršrutas A → B → C → D → E → H. Bebrai gali surinkti 8 + 4 + 2 + 8 + 9 = 31 rąstą.

Šiame uždavinyje nesunku įžvelgti grafą: plaustai – tai viršūnes, upės – briaunos.

Lengviausias būdas rasti geriausią maršrutą yra peržiūrėti grafą nuo pradinės viršūnės, plukdyti rąstus ir skaičiuoti iš eilės, iki kurios viršūnės bus daugiausiai rąstų, o tada analogiškai tęsti nuo tos viršūnės kelionę toliau.

Viršūnė gali turėti dvi būsenas: įjungta arba išjungta (ant plausto galima krauti rąstus arba ne). Pradžioje visos viršūnės išjungtos.

Geriausia skaičiuoti šitaip:

- įjungti pirmąją viršūnę,
- kiekvienai įjungtai viršūnei pakartoti šiuos veiksmus:
  - kiekvienai likusiai briaunai
    - briaunos reikšmę pridėti prie viršūnės reikšmės,
    - jei ši nauja reikšmė didesnė negu viršūnės, į kurią veda briauna, tai reikšmė pakeičiama ir ši viršūnė įjungiama,
    - prieš tai buvusi viršūnė išjungiama.

Lentelėje rodomi atliekami veiksmai (raudonai pažymėtos įjungtos viršūnės):

A	B	C	D	E	F	G	H
0	0	0	0	0	0	0	0
0	8	0	9	7	0	0	0
0	8	12	11	17	14	13	14
0	8	12	14	19	16	15	16
0	8	12	14	22	19	18	19
0	8	12	14	22	19	18	31
0	8	12	14	22	19	18	31

Tai informatika!

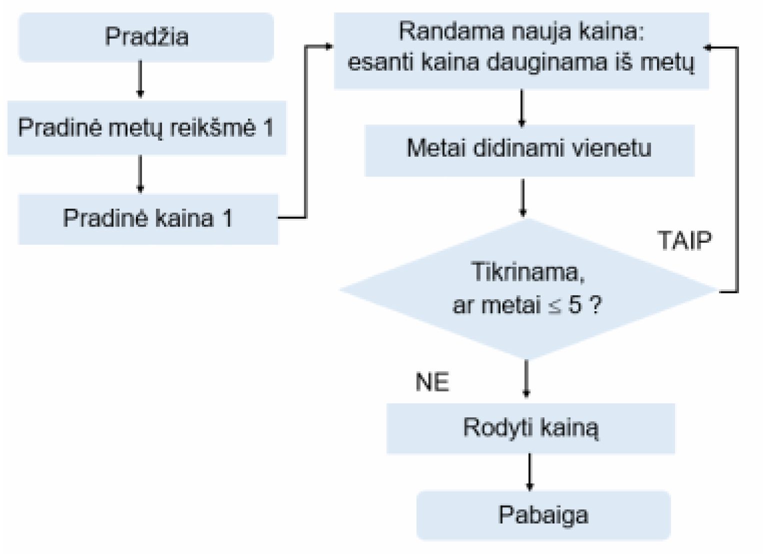
Ši užduotis remiasi grafais, kai reikia apskaičiuoti ilgiausią kelią. Grafai naudojami sudarant tvarkaraščius. Ilgiausias kelias gali būti apskaičiuojamas, atliekant topologinį rikiavimą orientuotame grafe.

Reikšminiai žodžiai

Ilgiausio kelio problema, topologinis rikiavimas

7. 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 – 1 pinigas.



Kiek kainuoja 5 metus brandintas sūris?

Paaiškinimas

Pradinė kaina = 1, metai = 1.

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

Po dvejų metų kaina = 1 x 2 = 2, metai = 2 + 1 = 3.

Po trejų metų kaina = 2 x 3 = 6, metai = 3 + 1 = 4.

Po ketverių metų kaina = 6 x 4 = 24, metai = 4 + 1 = 5.

Po penkerių metų kaina = 24 x 5 = 120, metai = 5 + 1 = 6.

Šitaip suskaičiavus, metų skaičius didesnis už 5, todėl toliau nebeskaičiuojama ir rodoma galutinė kaina.

Tai informatika!

Programa yra pateikiama kaip struktūrinė schema. Tai įprastas būdas, jei nevartojama jokia programavimo kalba. Pradedama nuo pradžios, toliau, sekant rodyklėmis, pasiekiama pabaiga. Programoje pateikiamas ciklas ir veiksmų seka, kurias kompiuteris turi atlikti keletą kartų. Šiuo atveju 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.

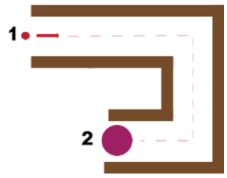
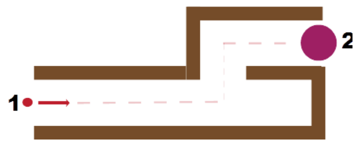
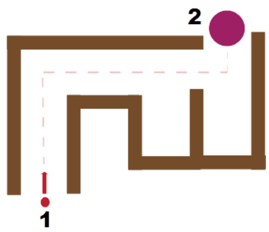
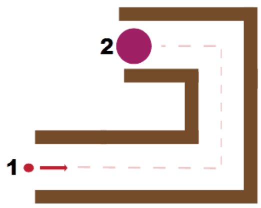
Šiuo atveju programa skaičiuoja faktorialą, kuris naudojamas ne tik įvairiose matematikos srityse, bet ir informatikoje.

Reikšminiai žodžiai

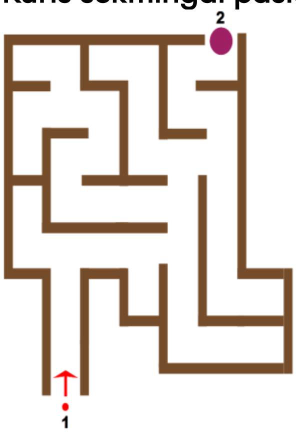
Struktūrinė schema, kintamieji, priskyrimas, sąlyginis ciklas

8. Labirintas

Eglė, Ieva, Lukas ir Matas nori pereiti labirintą nuo pradžios (1) iki pabaigos (2). Kiekvienas jų turi savas kelio pasirinkimo taisykles ir jas išbandė, mėgindami pereiti nedidelį savo pačių pasidarytą labirintą. Taisyklės ir pagal jas išbandyti labirintai pateikti lentelėje.

Eglė	*Jei nesusiduriu su siena, einu tiesiai.* *Jei susiduriu su siena, suku į dešinę.*	
Ieva	*Jei kairėje nėra sienos, suku į kairę ir einu tiesiai.* *Antraip, jei susiduriu su siena, suku į dešinę.* *Antraip, jei nesusiduriu su siena, einu tiesiai.*	
Lukas	*Jei nesusiduriu su siena, einu tiesiai.* *Jei susiduriu su siena, pasisuku į dešinę ir einu tiesiai, kol susiduriu su kita siena, kur pasisuku į kairę ir einu tiesiai.*	
Matas	*Jei nesusiduriu su siena, einu tiesiai.* *Jei susiduriu su siena, suku į kairę.*	

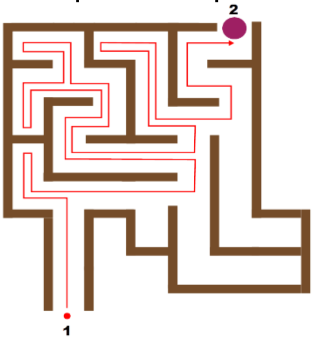
Kuris sėkmingai pasieks labirinto pabaigą?

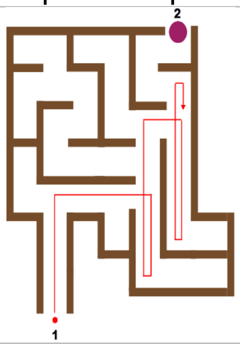
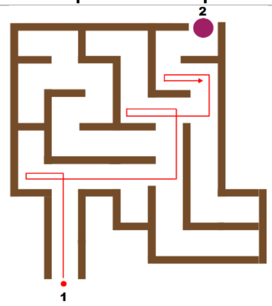


A. Eglė
B. Ieva
C. Lukas
D. Mantas

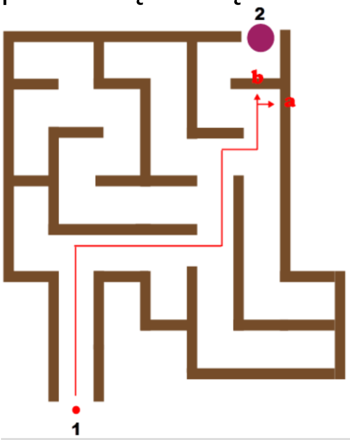
Paaiškinimas

Pabaigą pasiekia tik Ieva. Jos taisyklės tinka bet kokiam labirintui. Tačiau jos kelias labirintu nėra pats trumpiausias.



Eglė nepasiekia pabaigos	Matas nepasiekia pabaigos
	

Lukas nepasiekia pabaigos, nes, kai taške *b* susiduria su siena, jo taisyklės sako, kad reikia pasisukti į dešinę ir eiti tiesiai, o ten – siena (*a*).



Tai informatika!

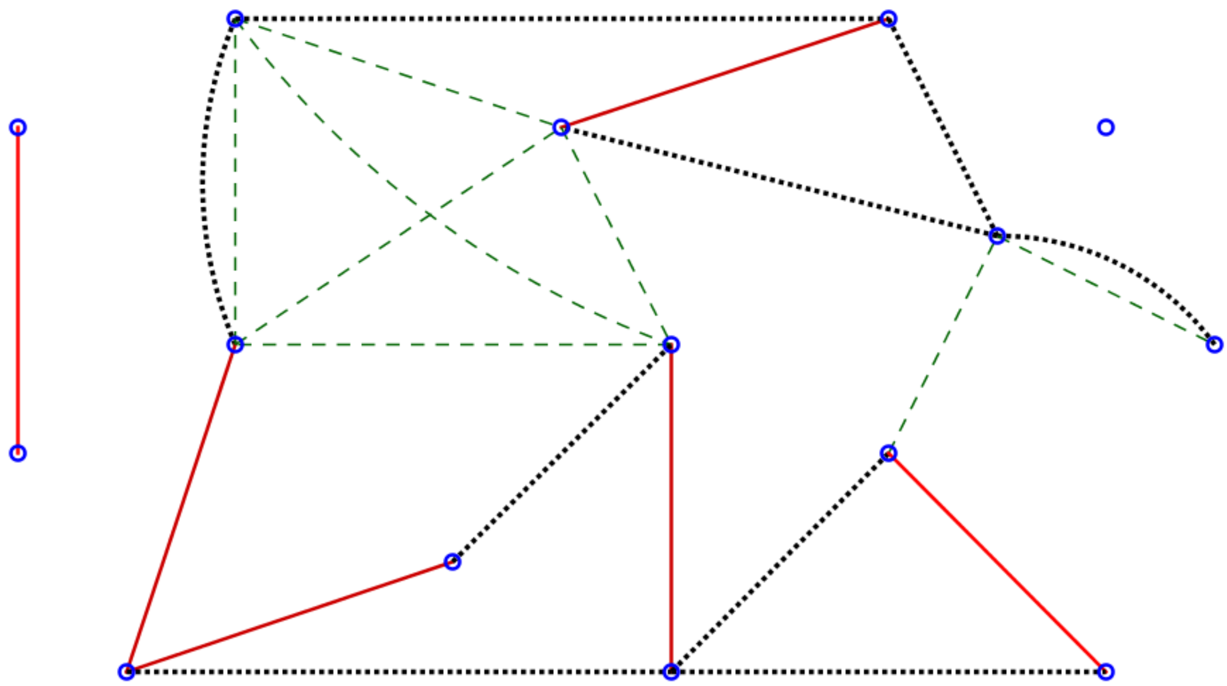
Informatikoje įprasta rasti bendrus sprendimus, tinkančius panašių uždavinių grupei. Šiuo atveju tokią grupę sudaro kelio paieška bet kuriame labirinte. Tokie sprendimai išreiškiami aiškių komandų arba taisyklių sekomis ir vadinami algoritmais. Algoritmai naudojami įvairiose informatikos srityse sudėtingiems kompleksiniams uždaviniams spręsti.

Reikšminiai žodžiai

Algoritmas, kelio paieška, labirintas, uždavinių sprendimas

9. Keleivių tarpusavio ryšiai

Petras, Jurgis ir Rima darė mokyklinį projektą apie ryšius tarp žmonių. Jie apklausė važiuojančius autobusu žmones apie tarpusavio ryšius ir nustatė trijų tipų ryšius tarp keleivių: dvyniai (broliai arba seserys), draugai ir klasės draugai. Petras sujungė dvynius raudona ištisine linija, Jurgis sujungė draugus juoda taškine linija, o Rima klasės draugams panaudojo žalią punktyrinę liniją.



Mokytojas, peržiūrėjęs sudarytą schemą, pasakė, kad tik vienas iš trijų mokinių teisingai atliko savo projekto dalį.

Kuris ryšio tipas schemoje pateiktas teisingai?

A) dvyniai ____________

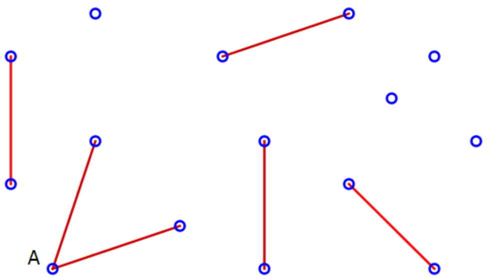
B) draugai ....

C) klasės draugai ------------

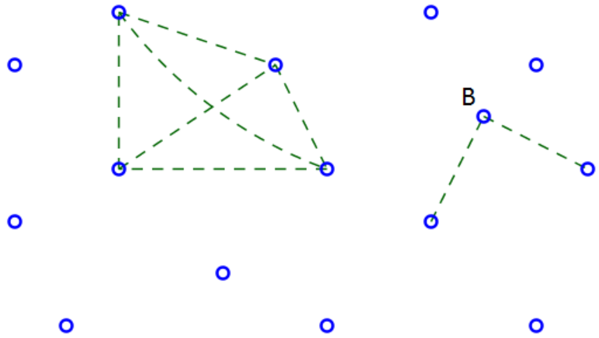
D) Mokytojas suklydo: daugiau kaip vienas mokinys pavaizdavo savo projekto dalį teisingai.

Paaiškinimas

Išskirkime iš grafo minėtus tris ryšių tinklus. Dvynių (brolių ir seserų) tinklas yra neteisingas. Yra dvi briaunos, išeinančios iš viršūnės A. Neįmanoma būti dvyniais su dviem skirtingais žmonėmis, kurie nėra dvyniai.

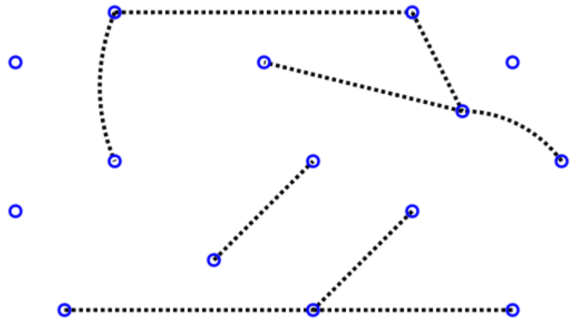


Klasės draugų tinklas yra neteisingas. Jei vienas asmuo yra kelių žmonių klasės draugas, tai ir kiekvienas iš tų žmonių grupės taip pat yra visų kitų klasės draugas. Jei viršūnė sujungta su kita keliomis briaunomis, einančiomis per kitas viršūnes, tai turi būti ir tiesioginė briauna, jungianti tas dvi viršūnes. Viršūnė B sujungta su dviem kitomis viršūnėmis, kurios nėra sujungtos tarpusavyje. Tai būtų įmanoma tik tada, jei asmuo B mokytųsi dviejose klasėse vienu metu, o tai nėra įprasta.



Draugų tinklas yra teisingas. Nėra taisyklių, kas turėtų būti kieno draugu. Draugams netaikomos tokios ryšių taisyklės, kaip, pavyzdžiui, darbo kolegoms ar klasės draugams.

Teisingas atsakymas – B (draugai).



Tai informatika!

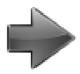
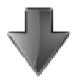
Grafas yra nuostabi priemonė pasauliui apibūdinti. Žmonių ryšiai , geležinkelio tinklai, darbų procesai ir t. t. gali būti vaizduojami grafais. Kompiuteriai gali labai greitai ir efektyviai spręsti uždavinius, aprašytus grafais. Informatikai bando grafais apibūdinti realias gyvenimo situacijas, o tada išversti tai į kompiuteriui suprantamą kalbą. Darbo kolegų grafas yra specialusis grafas, kuriame kiekviena viršūnė sujungta briaunomis su visomis kitomis viršūnėmis. Toks grafas vadinamas pilnuoju ir turi keletą įdomių savybių. Pavyzdžiui, jei sunumeruotume pilnojo grafo visas N viršūnes skaičiais 0, 1, 2, ... N-1, tai briaunų skaičius būtų lygus šių skaičių sumai.

Reikšminiai žodžiai

Grafai

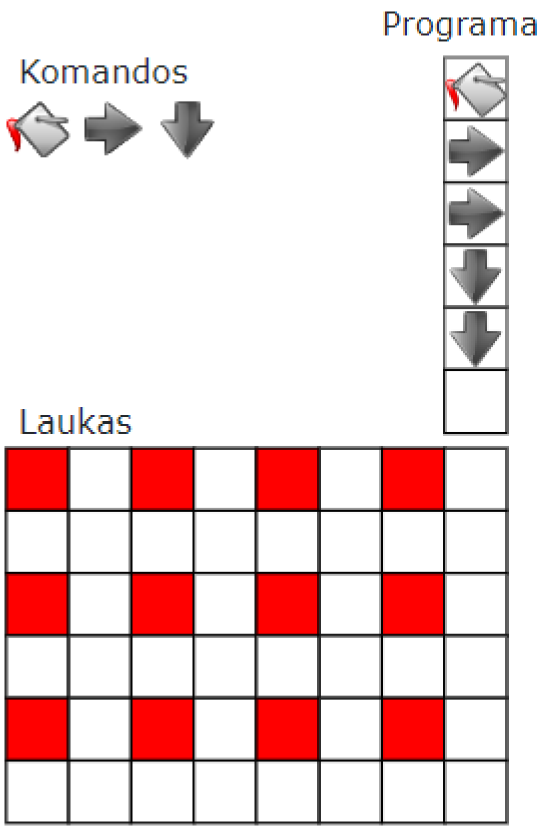
10. Robotas spalvintojas

Kompiuterinis robotas juda dvispalviame 8x6 dydžio lauke. Robotas supranta tris komandas:

	Robotas spalvina langelį, kuriame stovi. Jei langelis jau nuspalvintas, tada programa baigia darbą.
	Robotas paeina vienu langeliu į dešinę. Jei jis išeitų už lauko ribų, tai tęstų ėjimą tos pačios eilės kairiajame langelyje.
	Robotas paeina vienu langeliu žemyn. Jei jis išeitų už lauko ribų, tai tęstų ėjimą to paties stulpelio viršutiniame langelyje.

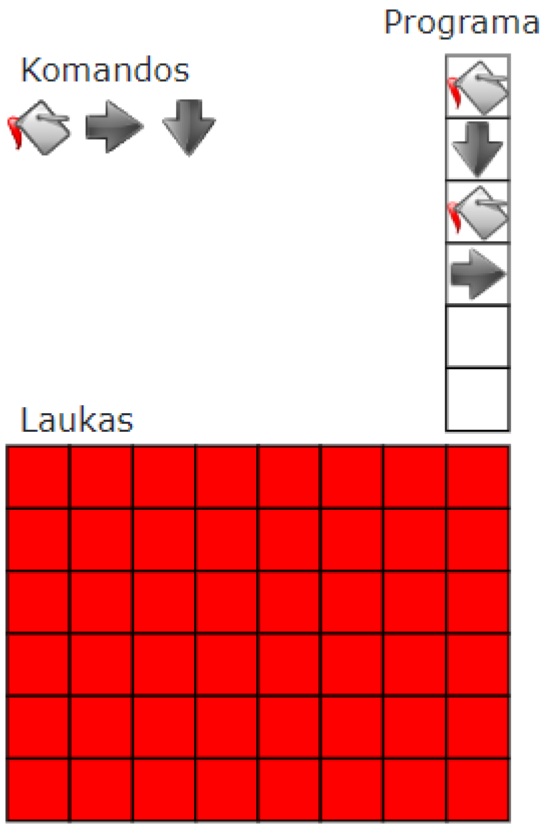
Iš pradžių visi lauko langeliai balti. Robotas pradeda viršutiniame kairiajame langelyje. Robotas kartoja komandas tol, kol atsistoja į jau nuspalvintą langelį.

Sudėliokite trumpiausią programą, kuri nuspalvintų visus lauko langelius. Programa turi tilpti į jai skirtą vietą.



Paaiškinimas

Teisingas atsakymas yra:



Tai yra trumpiausia programa iš galimų. Atkreipiame dėmesį, kad kiekvienoje programoje turi būti mažiausiai viena spalvinimo komanda, kitaip ji niekada nebaigs darbo.

Programa iš vienos komandos negalima – nuspalvinsite vieną langelį ir programa baigs darbą. Programa iš dviejų komandų gali turėti tik vieną rodyklės komandą, taigi ji nuspalvina eilutę (arba stulpelį) ir baigia darbą. Programa iš trijų komandų gali turėti dvi skirtingas rodyklių komandas – tada nuspalvinama tik pusė langelių. Jei programoje būtų dvi tos pačios rodyklės, būtų spalvinama tik viena eilutė (arba vienas stulpelis).

Tai informatika!

Šiuo uždaviniu siekiama supažindinti su labai paprastu įrenginiu, išmokti planuoti veiksmus ir programuoti.

Komandos atliekamos tol, kol sutinkama sąlyga, leidžianti nutraukti programą.

Laukas tarytum susivynioja, taigi iš jo susidaro vadinamasis toras, kurio paviršiumi galima judėti.

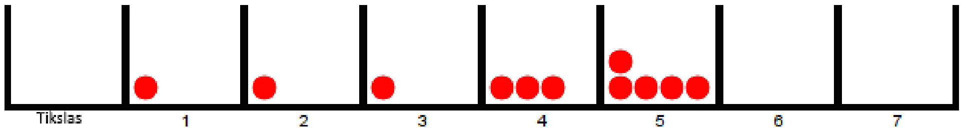
Reikšminiai žodžiai

Programavimas, komanda, kartojimas, gardelė

11. Akmenukų žaidimas

Žaidimo tikslas – perkelti visus akmenukus į kairįjį lovelį (po juo užrašyta **Tikslas**). Reikia laikytis šios taisyklės: lovelį su numeriu nuo 1 iki 7 galima ištuštinti, jei jo akmenukų skaičius yra lygus lovelio numeriui, tada jo akmenukai paskirstomi po vieną į visus kairiau esančius lovelius. Žaidimas laimimas, kai visi akmenukai sukeliami į lovelį **Tikslas**.

Pateikiama situacija, kai žaidimas gali būti laimimas.



Kuria tvarka reikia tuštinti lovelius?

Pateikite tuštinamų lovelių numerių seką – spustelėkite atitinkamus lovelių numerius.

Paaiškinimas

Atsakymas:

1 – 5 – 1 – 2 – 1 – 4 – 1 – 3 – 1 – 2 – 1

Šis žaidimas – tai vieno seniausių pasaulio žaidimų, mankalos, kildinamo iš Afrikos, variantas, kai žaidžia vienas žmogus. Žaidimas laimimas tik tada, jei strategija, kuria galima laimėti, išlaikoma iki pabaigos. Pagal šią strategiją reikia rinktis mažiausio numerio lovelį, kurį galima ištuštinti. Iš tikrųjų pamąstykime: jei galima ištuštinti du lovelius, tai pirmiau ištuštinus didesnio numerio lovelį, mažesnio numerio lovelis liktų užblokuotas. Pavyzdžiui, duotame paveiksle galima rinktis tuštinti vieną iš lovelių, kurių numeriai 1 ir 5. Deja, pasirinkus pastarąjį, 1-as lovelis būtų užblokuotas.

Tai informatika!

Sprendžiant šį uždavinį lemiamą vaidmenį vaidina veiksmų seka. Pateikiama laimėjimo strategija yra ne kas kita, kaip sąrašas algoritmo žingsnių, kuriuos atliekant žaidimas gali būti laimėtas, jei iš viso įmanoma laimėti. Algoritmo teisingumas gali būti įrodomas, pasitelkus matematinę indukciją.

Samprotaujame šitaip: jei yra tik vienas lovelis (numeris 1), aišku, kad žaidimas gali būti laimėtas, jei tame lovelyje yra tik vienas akmenukas, kitaip žaidimas nelaimimas. Toliau tarkime, kad pateiktas algoritmas teisingas N loveliams. Esant N+1 loveliui, reikia nagrinėti 3 atvejus: 1) (N+1)-ame lovelyje yra mažiau nei N+1 akmenukų, tada žaidimas nelaimimas; 2) (N+1)-ame lovelyje yra daugiau nei N+1 akmenukų, tada žaidimas nelaimimas; 3) (N+1)-ame lovelyje yra lygiai N+1 akmenukų, tada akmenukai gali būti paskirstyti (ir lovelis ištuštintas), kai nieko nebegalima daryti su loveliais 1, 2, ..., N. Iš tikrųjų N-ame lovelyje šiuo momentu turi būti tiksliai N-1 akmenukas. Kai (N+1)-ojo lovelio akmenukai perkeliami, (N+1)-asis lovelis ištuštinamas, paliekant akmenukus tik pirmuose N lovelių. Kadangi darėme prielaidą, kad algoritmas tinka, esant N lovelių, tai, remiantis indukcija, šis algoritmas tinka ir esant N+1 lovelių.

Reikšminiai žodžiai

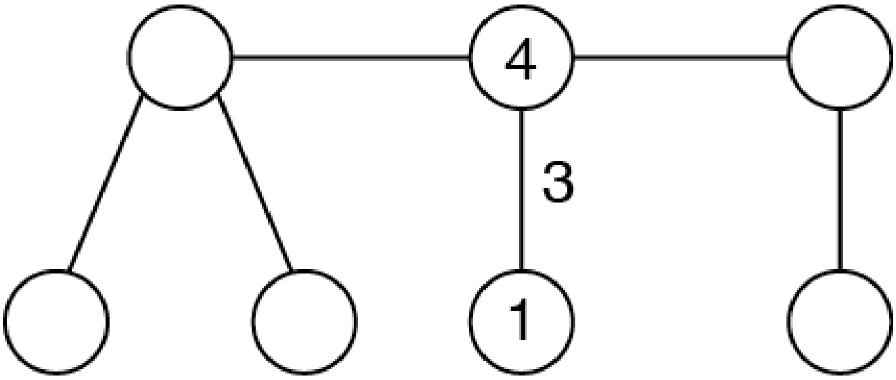
Indukcija, laimėjimo strategija.

12. Tobulasis grafas

Paveiksle vaizduojama schema, sudaryta iš apskritimų ir atkarpų. Į apskritimus reikia įrašyti skaičius 0, 1, ..., 6. Kai tik du skaičiai užrašomi atkarpos galuose, atkarpa žymima šių dviejų skaičių skirtumą vaizduojančia etikete.

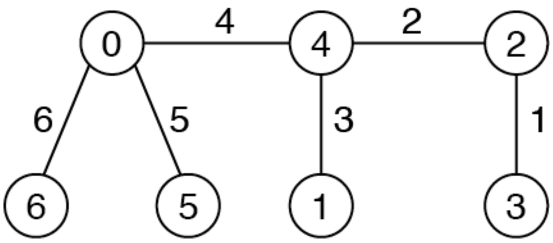
Paveiksle skaičiai 1 ir 4 jau įrašyti į du apskritimus, o juos jungianti atkarpa pažymėta skaičiumi 3.

Įrašykite likusius skaičius į tuščius apskritimus. Įsitikinkite, kad kiekvienos atkarpos etiketė yra skirtinga.



Paaiškinimas

Teisingas atsakymas yra:



Nesudėtinga patikrinti, kad atsakymas teisingas. Yra ir kitas sprendinys (sukeitus vietomis 5 ir 6).

Tai informatika!

Tai grafų teorija – tobulojo grafo ženklinimas, šiuo atveju tai – tobulasis medis. Grafai naudingi modeliuojant įvairius realaus gyvenimo uždavinius: eismo grafikus, žemėlapius, telefonų jungimą. Kompiuterijoje grafų algoritmai naudojami gana dažnai.

Reikšminiai žodžiai

Tobulasis grafas, tobulojo medžio radimas

13. Svarbiausioji sekos dalis

Informacija žmogaus genome dažniausiai koduojama keturių ženklų – A (adeninas), C (citozinas), G (guaninas) ir T (timinas) – sekomis, pavyzdžiui, CAGGAGGAT. Sekos gali būti labai ilgos. Mokslininkai jose ieško svarbių, bent du kartus pasikartojančių sekos dalių.

Sekos dalies svarba apibūdinama jos verte, apskaičiuojama taip:

Svarbiausioji sekos dalis yra ta, kurios vertė didžiausia. Pavyzdžiui, svarbiausioji sekos CAGGAGGAT dalis yra AGGA. Jos vertė yra 6 – jos ilgis 4 ženklai, pakartota 2 kartus. Sekos dalies G svarba mažesnė, nes jos vertė yra 5 – pakartota 4 kartus, jos ilgis tik vienas ženklas.

Kokia svarbiausioji sekos CATTGTTGTTGCATT dalis?

Paaiškinimas

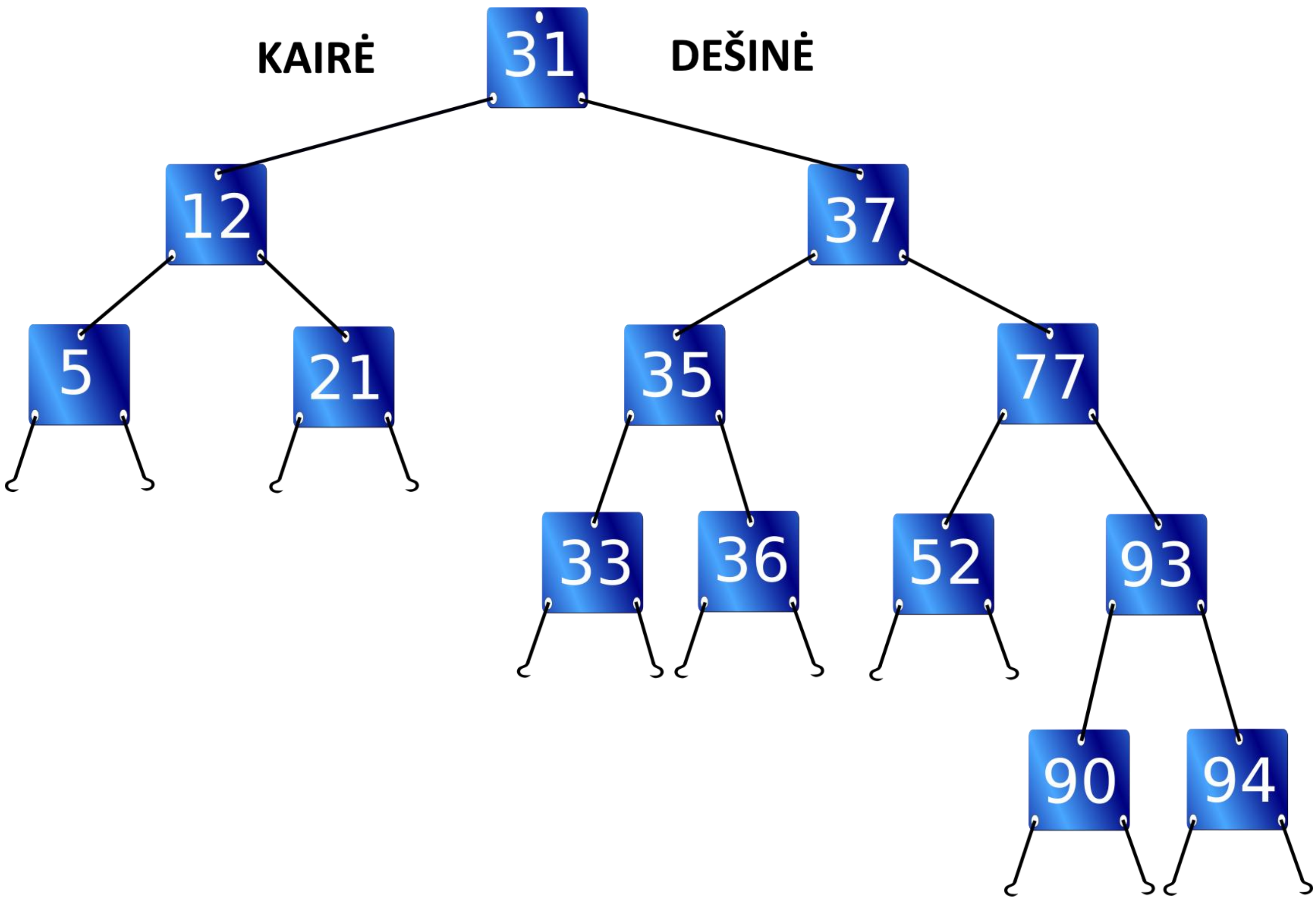
Sekos vertė yra 9 (ilgis – 1, pasikartojimų – 8). Daugiau dalių, turinčių tokią pat arba didesnę vertę, nėra.

Vienintelė sekos dalis, kurios vertė tik vienetu mažesnė, yra TTGTTG (ilgis – 6, pasikartojimų – 2). Kitų sekos dalių vertės dar mažesnės.

Tai informatika!

Genetinio kodo atradimas buvo lūžio taškas biologijoje. Tai pakeitė požiūrį į gyvybę – laikoma, kad ji užkoduota genomų sekomis. Informacijos kodavimas ir apdorojimas tapo esmine genetikos mokslo dalimi. XX amžiaus dešimtajame dešimtmetyje biologai pradėjo bendradarbiauti su informatikais, ėmė sparčiai vystytis bioinformatika. Nuo tada buvo sukurta daug naujų kompiuterinių biologinės informacijos vaizdavimo ir apdorojimo metodų. Visa tai turėjo didelį poveikį ne tik biologijai, bet ir medicinai bei farmacijai. Genetinių sekų analizė – tai tik viena bioinformatikos sritis.

14. Sunumeruotos kortelės



Paveiksle pavaizduota konstrukcija, sudaryta iš kortelių su numeriais:

- kiekviena kortelė turi skylutę viršuje ir du kabliukus apačioje;
- ant kiekvienos kortelės yra užrašytas natūralusis skaičius N.

Jei kortelė yra prikabinama prie kairiojo kabliuko, jos ir visų kitų po ja sukabintų kortelių numeriai turi būti mažesni už N.

Jei kortelė yra prikabinama prie dešiniojo kabliuko, jos ir visų kitų po ja sukabintų kortelių numeriai turi būti didesni už N.

Kelios sunumeruotos kortelės turi laisvus kabliukus.

Prie kelių laisvų šios konstrukcijos kabliukų galima prikabinti korteles?

Įrašykite didžiausią skaičių.

Paaiškinimas

Yra 14 laisvų kabliukų. Nė vienas iš kortelės su numeriu 36 kabliukų negali būti sujungtas su kitomis kortelėmis. Kortelė, prijungta prie kairiojo kabliuko, turi turėti numerį, didesnį nei 35 ir mažesnį nei 36, bet tokio natūraliojo skaičiaus nėra. Kortelė, prijungta prie dešiniojo kabliuko, turi turėti numerį, didesnį nei 36, bet mažesnį nei 37, bet tokio natūraliojo skaičiaus irgi nėra. Taip pat prie kortelės su numeriu 94 kairiojo kabliuko negalima prikabinti jokios kortelės.

Tai informatika!

Sunumeruotų kortelių struktūra vadinama dvejetainiu paieškos medžiu. Tai atrodo kaip tikras, tačiau apverstas medis. Jis vadinamas dvejetainiu (lotyniškas žodis „bis“ reiškia „du kartai“), nes kiekviena sunumeruota kortelė turi du kabliukus, todėl prie jos tiesiogiai gali būti prikabintos daugiausiai dvi kortelės. Dvejetainiai paieškos medžiai naudojami visų rūšių duomenims saugoti. Dvejetainio paieškos medžio privalumas – saugomus duomenis galima rasti labai greitai.

Reikšminiai žodžiai

Paieškos medis, dvejetainis medis

	Jauniai 12	Kolegos

15. Paštas

Paštininkas bebras turi atnešti pašto siuntas į vienos gatvės namus. Gretimi namai vienas nuo kito yra nutolę 1 km atstumu. Bebrui mokama už atstumą, kurį jis nueina, nešdamas siuntas į visus namus. Taigi bebras nori nueiti kuo daugiau kilometrų. Jis gali pradėti nuo bet kurio namo ir bet kuriuo namu baigti. Siuntos pristatomos į kiekvieną namą po kartą.

				
1	2	3	4	5

Kuria tvarka bebras turėtų aplankyti penkis namus, kad nueitų ilgiausią atstumą?

Spustelėkite namus ta tvarka, kuria bebrui reikėtų juos aplankyti.

Paaiškinimas

Ilgiausias atstumas yra 11 kilometrų. Yra 8 tokio ilgio keliai:

3—1—5—2—4

3—5—1—4—2

4—2—5—1—3

2—4—1—5—3

3—4—1—5—2

3—2—5—1—4

2—5—1—4—3

4—1—5—2—3

Lengva surasti kelią 1—5—2—4—3, kuris yra 10 kilometrų ilgio: eiti pirmyn ir atgal nuo pabaigos ir pabaigti, kai visi namai bus aplankyti. Tačiau tai nėra optimalus sprendimas. Galite pagerinti šį sprendimą, pradėdami nuo 3 namo ir pasirinkdami tolimiausią namą kiekviename žingsnyje. Tada gausite seką 3—5—1—4—2, kuri sudaro 11 kilometrų. Įsitikinti, kad neįmanoma nueiti daugiau negu 11 kilometrų, galima, atsižvelgus į atstumus tarp namų:

1—5: 4 km

1—4: 3 km

2—5: 3 km

Visi kiti atstumai yra mažesni negu 3 km. Eidami visais šiais trimis atstumais, galėsime nueiti daugiausiai 11 kilometrų. Pavyzdžiui, 3—4—1—5—2 arba 4—1—5—2—3. Jei vieno iš šių pateiktų trijų atstumų nenueisime, ilgiausias atstumas bus 2 + 2 + 3 + 3 (toks pavyzdys yra jau aptartas 1—5—2—4—3) arba 2 + 2 + 3 + 4 (toks pavyzdys yra 3—5—1—4—2, tai vienas iš optimalių sprendinių), kurie abu yra trumpesni negu 12. Visos kitos galimybės (kai nenueiname dviejų arba daugiau iš minėtų trijų atstumų) – aiškiai trumpesni keliai. Kiti optimalūs sprendiniai galimi, keičiant takų kryptį (nuo pabaigos iki pradžios) ir simetriją (keičiant 2 su 4 ir 1 su 5).

Tai informatika!

Optimizavimu vadinama paieška uždavinio sąlyga apibrėžtoje aibėje tokio elemento, kuriam kriterijaus reikšmė būtų minimali arba maksimali.

Namų aplankymo tvarkos, kai ieškoma ilgiausio kelio, radimas – tai uždavinio sprendimo optimizavimo pavyzdys. Intuityvūs šio uždavinio sprendimo metodai atitinka godžiuosius algoritmus. Tokių uždavinių sprendinio paiešką dažniausiai išskaidome į *n* etapų ir kiekviename žingsnyje renkamės iš nedidelio baigtinio skaičiaus variantų *m*. Godieji algoritmai rekomenduoja rinktis lokaliai geriausią variantą duotojo žingsnio metu. Todėl ne visada godieji algoritmai suranda viso uždavinio optimalų sprendimą.

Šio uždavinio atveju maršrutas 1—5—2—4—3 pradedamas pirmuoju namu ir toliau vis keliaujama prie paties tolimiausio namo. Deja, tai nėra geriausias sprendimas. Tačiau tokiu pačiu metodu gautas maršrutas 3—4—1—5—2 (pradedama nuo 3-o namo) jau pateikia optimalų atsakymą. Šiame uždavinyje yra tik 5 namai, tai galima perrinkti visus įmanomus maršrutus. Tačiau didėjant elementų skaičiui jau neįmanoma apsieiti be programavimo. Šis uždavinys gali būti modeliuojamas grafu, kuris yra apibrėžiamas kaip aibė viršūnių kartu su aibe briaunų, jungiančių viršūnių poras. Kiekvienas namas šiuo atveju yra viršūnė, kiekviena pora namų sujungti briauna (taigi grafas yra pilnasis), o atstumas tarp dviejų namų – atitinkamos briaunos vertė (svoris).

Maršrutas (kelias) grafe yra briaunų, nuosekliai jungiančių viršūnes, seka. Šiuo atveju reikia surasti ilgiausią Hamiltono maršrutą (kelią), kuriame kiekviena grafo viršūnė aplankoma lygiai vieną kartą. Jei grafas didžiulis, pasitelkiami euristiniai metodai.

Reikšminiai žodžiai

Optimizavimo uždaviniai, godusis algoritmas, grafai

16. Pasivyk

Bebrė Alisa turi dešimties nugraužtų medžių krūvą, o bebrė Beatričė – tik vieną medį.

Bebrės pradeda graužti medžius miške vienu metu. Kiekvieną nuverstą medį jos prideda prie savo medžių krūvos. Alisa nugraužia medį per valandą. Beatričė graužia medžius vis sparčiau. Pirmąją valandą ji nugraužia tik vieną medį, antrąją – du, trečiąją – jau tris medžius ir taip toliau.

Po kelių mažiausiai valandų Beatričė turės ne mažiau medžių, negu Alisa?

(A) 4

(B) 5

(C) 6

(D) 7

Paaiškinimas

Teisingas atsakymas – B.

Laikas	Alisos krūva – medžių skaičius	Beatričės krūva – medžių skaičius
Iš pradžių	10	1
Po 1 valandos	11	2
Po 2 valandų	12	4
Po 3 valandų	13	7
Po 4 valandų	14	11
Po 5 valandų	15	16

Taigi po 5 valandų Beatričė turi 16 medžių, o Alisa – tik 15.

Jei norite pasunkinti užduotį ir pritaikyti ją vyresnių klasių mokiniams, galite pakeisti medžių skaičių krūvose.

Toliau pateikiamas užduoties sprendimas, jei Alisa iš pradžių turi 1000, o Beatričė – 10 medžių.

Medžių, Alisos nuverstų po *t* valandų, skaičius yra 1000 + t.

Medžių, Beatričės nuverstų po *t* valandų, skaičius yra 10 + 0,5 t + 0,5 t2.

Pasitelkę matematiką, galime pastebėti: jei *t*=45, tai Alisa nuvertė 1045 medžius, taip pat ir Beatričė nuvertė 10 + 0.5 (45) (452) = 1045 medžius.

Tai informatika!

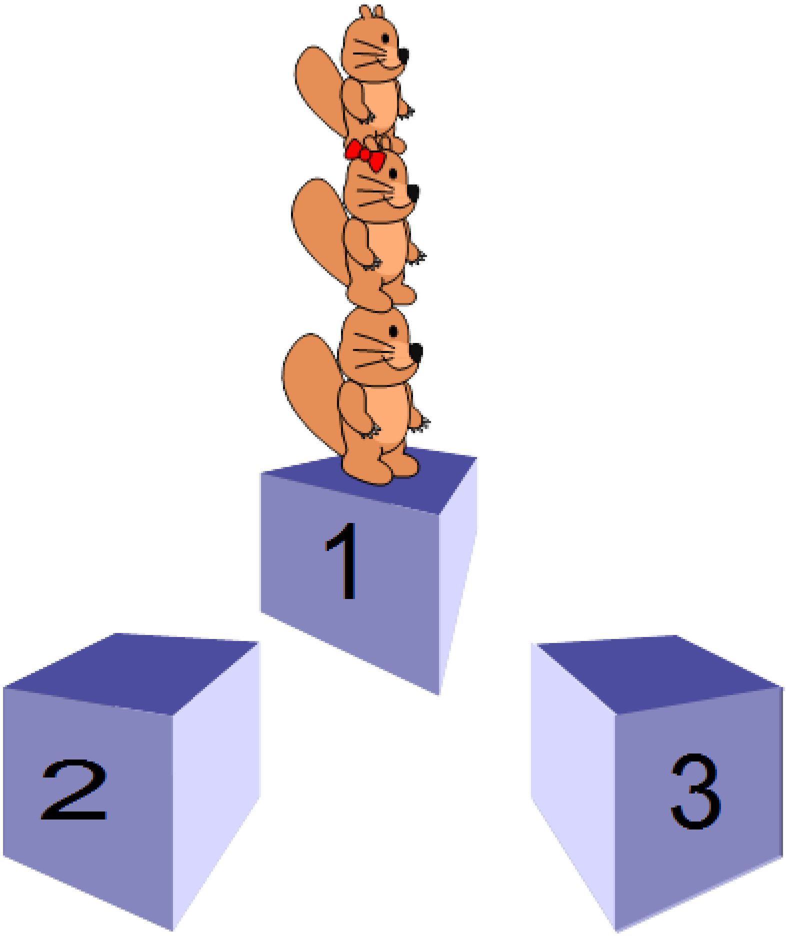
Šis uždavinys atkreipia dėmesį į asimptotinę algoritmų analizę. Būtų galima sakyti, kad Alisa nuverčia medžius tiesiniu dažniu, vaizduojamu O(*n*). Vadinasi, po *n* valandų Alisa bus nuvertusi medžių skaičių, proporcingą *n*. Beatričė nuverčia medžius proporcingai laikui, pakeltam kvadratu, vaizduojama O(*n*2). Tai reiškia, kad po *n* valandų Beatričė bus nuvertusi medžių skaičių, proporcingą skaičiui *n*2.

Reikšminiai žodžiai

Asimptotinė analizė, augimo tempas

17. Akrobatai

Akrobatų bebrų šeima – tėvas, mama ir vaikas – rengia akrobatinių šuolių kaskadą. Yra trys pakylos. Iš pradžių šeima stovi ant pirmos pakylos, vienas kitam ant pečių.



Šokti nuo vienos platformos ant kitos galima pagal šias dvi taisykles:

- Šoka tik viršutinis bebras.
- Bebras ant savo pečių gali laikyti tik mažesnį bebrą:
    - Tėvas gali laikyti bet kurį kitą.
    - Mama gali laikyti tik vaiką, bet ne tėvą.
    - Vaikas nieko negali laikyti.

Kiek mažiausiai šuolių reikia atlikti, kad visa šeima sustotų kaskada ant 3-ios pakylos?

A. 5

B. 6

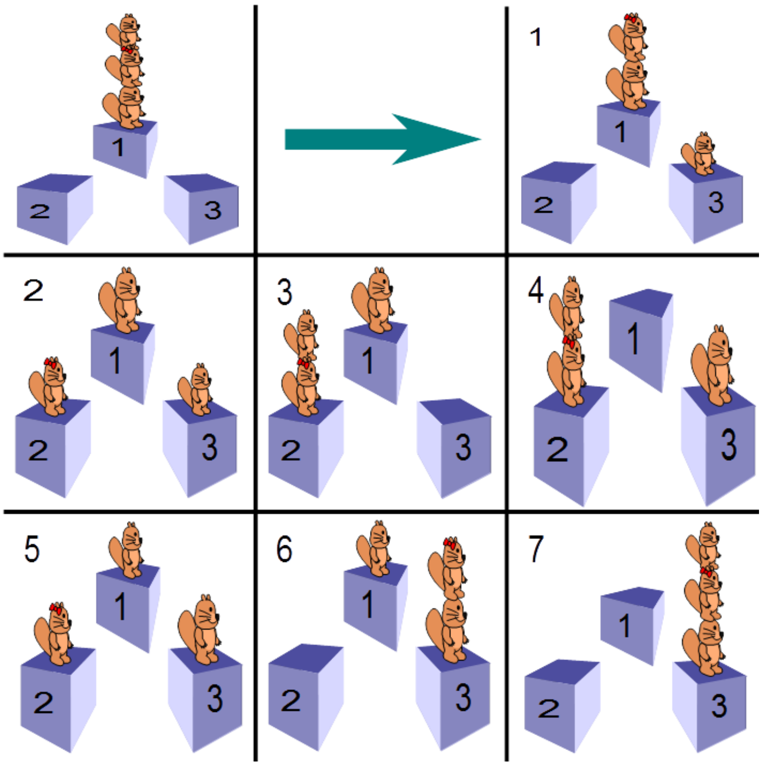
C. 7

D. 8

Paaiškinimas

Teisingas atsakymas – C.

Sprendimas paaiškintas paveikslėliais:



Visa bebrų šeima turi peršokti ant 3-ios pakylos. Pirmiausia tėvas turi atsidurti ant 3-ios pakylos. Vadinasi, mama turi būti ant 2-os pakylos. Kad mama galėtų šokti ant 2-os pakylos, vaikas turi šokti ant 3-ios pakylos. Gauname štai tokią šuolių seką:

1. Vaikas šoka (nuo 1 ant 3)
2. Mama šoka (nuo 1 ant 2)
3. Vaikas šoka (nuo 3 ant 2)
4. Tėvas šoka (nuo 1 ant 3)
5. Vaikas šoka (nuo 2 ant 1)
6. Mama šoka (nuo 2 ant 3)
7. Vaikas šoka (nuo 1 ant 3)

Tai informatika!

Šio uždavinio idėja kilo iš Hanojaus bokštų uždavinio. Esant trims diskams, uždavinys išsprendžiamas septyniais ėjimais. Minimalus ėjimų skaičius, reikalingas Hanojaus uždaviniui išspręsti, yra 2*n*-1, čia *n* yra diskų skaičius. Tai įrodoma indukcijos metodu: vienas bebras ant 3-ios pakylos gali patekti vienu ėjimu. Įsivaizduokime, kad žinome sprendimą, kai yra *n*-1 bebrų – tarkime, reikia 2*n*-1-1 ėjimų. Jei papildome vienu bebru iš apačios, būtina peršokdinti *n*-1 viršuje esantį bebrą ant 2-os pakylos – tam prireiks 2*n*-1-1 ėjimų. Tada apatinis bebras galės vienu šuoliu atsidurti ant 3-ios pakylos ir *n*-1 bebrų nuo 2-os pakylos galės peršokti ant 3-ios jau žinomu būdu. Bendras ėjimų skaičius būtų 2*n*-1-1+1+2n-1-1 = 2·(2*n*-1-1)+1 = 2·2*n*-1-1 = 2*n*-1.

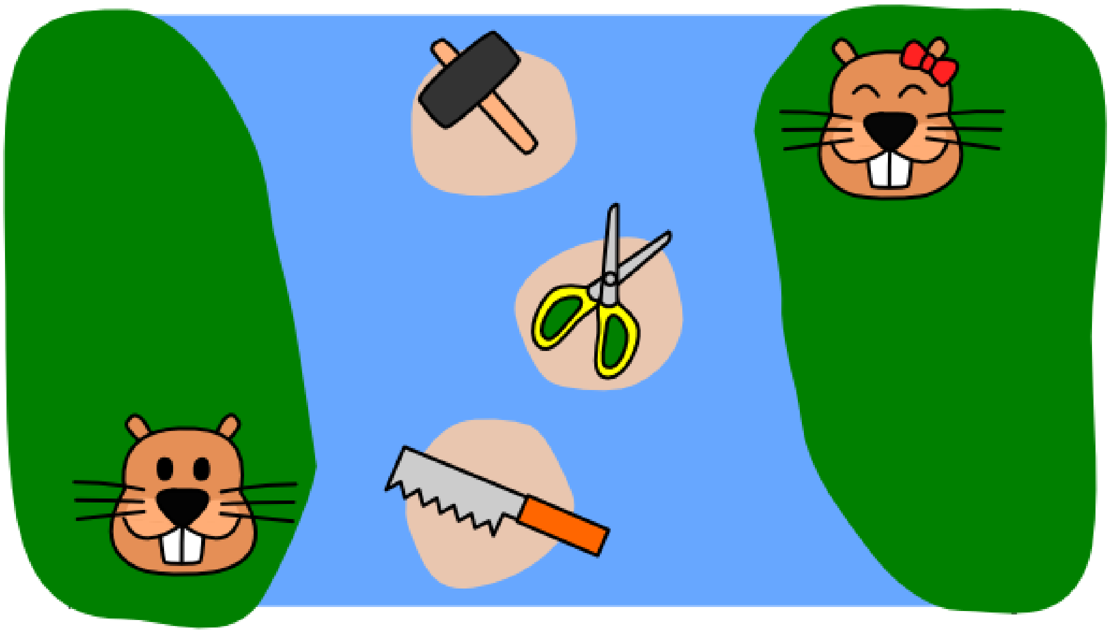
Reikšminiai žodžiai

Hanojaus bokštai

18. Medžio dirbiniai

Prie upės įsitaisę bebriukai Bitaro ir Bibako kuria medinius žaislus. Abu naudojasi tuo pačiu plaktuku, žirklėmis ir pjūklu. Įrankiai padėti ant upėje esančių smėlio salelių. Kuriam prireikia įrankio, tas jį ir pasiima, kai nebereikia, grąžina atgal į smėlio salelę. Jei reikiamo įrankio nėra salelėje, bebriukas laukia, kol tą įrankį grąžins kitas bebriukas.

Kartais abiem bebriukams vienu metu reikia to paties įrankio. Jei taip atsitinka, jie tiesiog nustoja kurti žaislus ir eina maudytis.



Kuriai iš pateiktų situacijų esant bebriukai tikrai eis maudytis?

A) Bitaro turi  ir jam reikia . Bibako turi  ir jam reikia .

B) Bitaro turi  ir jam reikia . Bibako turi  ir jam reikia .

C) Bitaro turi  ir jam reikia . Bibako turi  ir jam reikia .

D) Bitaro turi  ir jam reikia . Bibako turi  ir jam reikia .

Paaiškinimas

Jei susiklosto situacijos A, C ir D, vienas iš bebriukų gali gauti reikiamą įrankį, baigti kurti žaislą ir grąžinti į salelę abu įrankius. Tada kitas bebriukas gali gauti reikiamą įrankį ir baigti kurti savo žaislą.

B situacijoje bebriukai turės laukti vienas kito amžinai, nes pasiėmė vienas kitam reikiamą įrankį.

Tai informatika!

Įrankio pasiėmimas atitinka proceso blokavimą. Situacija, kai kiekvienas sąveikaujančios procesų aibės procesas laukia įvykio, kurį gali pateikti tik tos aibės procesas, vadinama aklaviete. Kai programos bando prieiti prie bendrai naudojamų išteklių, atsižvelgti į procesų blokavimą yra svarbu, siekiant išvengti aklavietės.

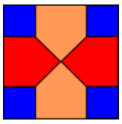
Nemažai kompiuterių ir kompiuterių tinklų programų sąveikauja tarpusavyje ir veikia lygiagrečiai. Bendras išteklių naudojimas ir aklavietės vengimas yra svarbūs informatikos konceptai.

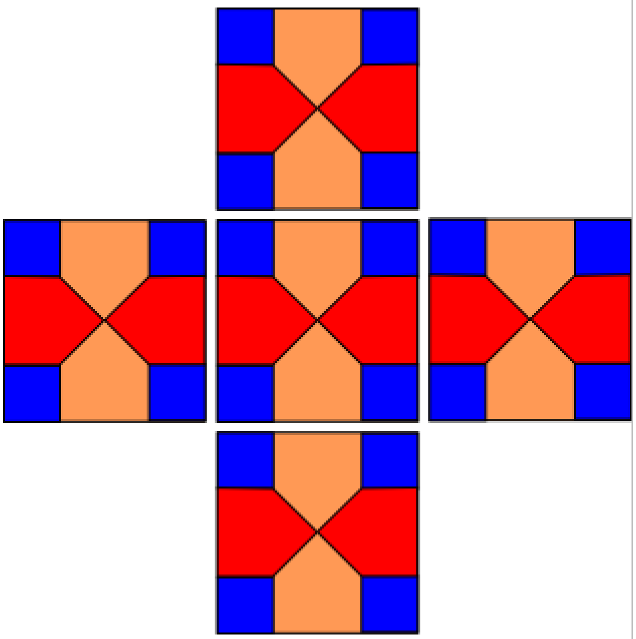
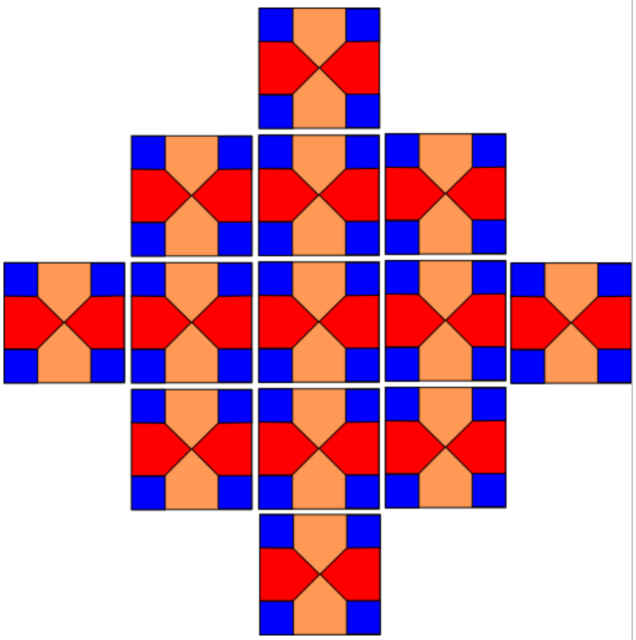
Reikšminiai žodžiai

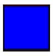
Aklavietė, išimčių valdymas, lygiagretusis apdorojimas

19. Vitražas

Robotas puošia langus vitražais iš spalvoto stiklo gabaliukų. Kiekviename vitraže kartojamas vienas toks pat fragmentas:



3-ų stulpelių lango vitražą sudaro 5 fragmentai:	5-ų stulpelių lango vitražas atrodo šitaip:
	

Kiek mėlynų stiklo gabaliukų  **reikia, norint sukurti 7 stulpelių lango vitražą?**

Paaiškinimas

Teisingas atsakymas – 100.

Vienas iš galimų sprendimo būdų – rasti dėsningumą, kaip, langui didėjant, pridedami nauji fragmentai.

Jei norime padidinti langą, pridėdami du stulpelius (pavyzdžiui, iš 3 gauti 5 arba iš 5 gauti 7), turime:

1) pridėti po 4 fragmentus visomis 4 kryptimis nuo lango centro (į kairę, į dešinę, viršuje, apačioje),

2) įdėti po vieną fragmentą į kiekvieną kampą (pridedamas fragmentas turi liesti dvi kitų fragmentų kraštines). Tokių kampų 5 stulpelių lange yra 8.

Prie esančių 13 reikia pridėti 4 + 8 = 12 naujų fragmentų. Taigi 7 stulpelių langą sudaro 25 fragmentai.

Kiekviename fragmente yra po 4 mėlynus stiklo gabaliukus, tad 7 stulpelių lange yra 100 tokių gabaliukų.

Jei norėtume suskaičiuoti mėlynus stačiakampius gabaliukus bet kurio dydžio lange, galėtume pritaikyti šią matematinę formulę: SUM((i-1)*2+1) * 2 + N.

Arba kitaip: fragmentų skaičius W =$N + 2 * \sum_{i=0}^{N/2}(i-1) * 2 + 1$.

Tada visų stačiakampių gabaliukų skaičius būtų W x 4.

Galime įrodyti tiesiog samprotaudami:

3 stulpeliai: 1 + 3 + 1 fragmentai,

5 stulpeliai: 1 + 3 + 5 + 3 + 1 fragmentai,

7 stulpeliai: 1 + 3 + 5 + 7 + 5 + 3 + 1 fragmentai = 25 fragmentai. Kiekviename fragmente yra po 4 mėlynus stačiakampius. Iš viso – 4 x 25.

Arba galima tiesiog pastebėti, kad šachmatine tvarka nudažius vitražo fragmentus, vienos spalvos fragmentai sudaro kvadratą su kraštine iš (N+1)/2 fragmentų, o kitos spalvos – su kraštine iš (N-1)/2 fragmentų, taigi iš viso yra $(N+1)^2/4 + (N-1)^2/4$ fragmentų, taigi $(N+1)^2 + (N-1)^2 = 2N^2 + 2$ mėlynų stiklo gabaliukų.

Tai informatika!

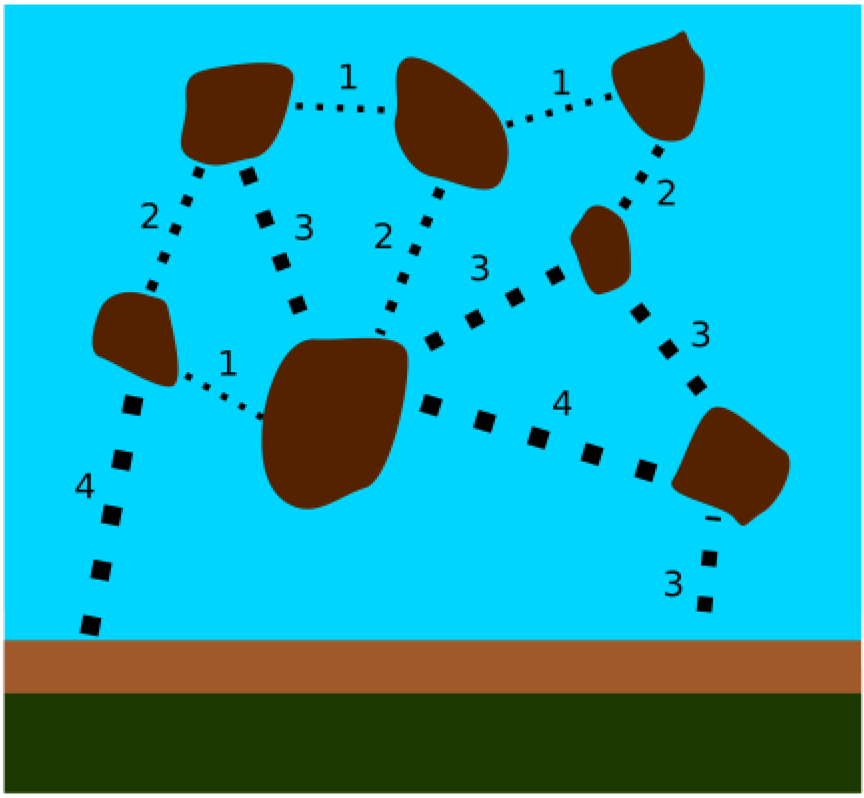
Robotas veikia pagal programą. Programos funkcionalumas aprašomas, pateikiant porą nedidelių pavyzdžių. Sprendėjui reikia perskaityti funkcinę uždavinio specifikaciją ir paversti ją programa, t. y., sugalvoti algoritmą, kaip galima kurti vitražą, ir tada apskaičiuoti prašomą rezultatą.

Reikšminiai žodžiai

Funkcinis projektavimas

20. Bebro tunelių jungimas

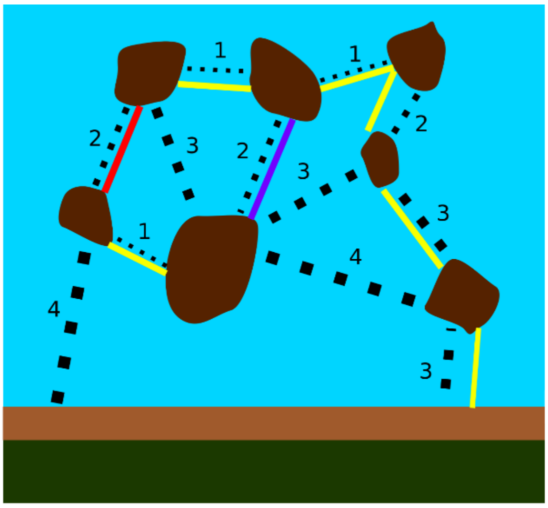
Tvenkinyje, netoli kranto, yra septynios bebro trobelės (žr. paveikslą). Taškinės linijos rodo tiltus, kuriuos galima nutiesti. Skaičiai rodo, kiek medžių prireiks atitinkamam tiltui nutiesti. Bebras nori, kad bet kuri trobelė nuo kranto būtų pasiekiama tik tiltais (kad jam nereikėtų plaukti), ir turi nuspręsti, kuriuos tiltus tiesti.



Kiek mažiausiai medžių reikia tiltams nutiesti?

Paaiškinimas

Kad būtų sunaudota kiek įmanoma mažiau medžių, turi būti nutiesti visi geltoni tiltai ir dar vienas raudonas arba violetinis tiltas:



Šiems septyniems tiltams iš viso reikia 1 + 1 + 1 + 2 + 2 + 3 + 3 = 13 medžių.

Kodėl 13 yra geriausias sprendimas?

Neformalus argumentas

Reikalingi lygiai septyni tiltai. Jei būtų nutiesta mažiau kaip septyni tiltai, bent viena trobelė būtų nepasiekiama. Jei būtų nutiesta daugiau kaip septyni tiltai, statyboms tiesiog reikėtų daugiau medžių. Septyniems tiltams reikalingas mažiausias medžių skaičius: 1 + 1 + 1 + 2 + 2 + 2 + 3 = 12. Tačiau, jei bus nutiesta tik tiek tiltų, lengva įsitikinti, kad bent viena trobelė tiltais bus nepasiekiama.

Formalesnis argumentas

Vienas iš dviejų tiltų prie viršuje, dešinėje, esančios trobelės turi būti nutiestas, kad ji būtų pasiekiama neplaukiant. Kadangi abi gretimos trobelės taip pat bus pasiekiamos tiltais, nėra prasmės rinktis tiltą, kuriam nutiesti reikia dviejų medžių. Taip samprotaujant, galima įrodyti, kad turi būti nutiesti visi trys tiltai, kuriems reikia po vieną medį. Patartina atsargiai rinktis tiltus, kuriems reikia dviejų medžių, nes raudonas ir violetinis tiltai yra nebūtini. Pavyzdžiui, jei būtų nutiestas raudonas tiltas ir jei bet kuri trobelė yra pasiekiama tiltais, vadinasi, ir visos keturios trobelės būtų pasiekiamos tiltais. Taip pat nutiktų, jei būtų nutiestas violetinis tiltas. Taigi reikalingas tik vienas iš šių tiltų. Taip pat samprotaujant turėtų būti nutiestas trečias tiltas, kuriam reikia dviejų medžių (tarp visų tiltų iki trobelės tai vienintelis tiltas, kuriam reikia mažiausiai medžių). Matome, kad du tiltai, kuriems reikia trijų medžių, turi būti nutiesti ir kad nėra prasmės statyti daugiau tiltų, nes tam prireiktų daugiau medžių.

Tai informatika!

Šiuo uždaviniu sprendžiama minimalaus jungiančiojo medžio radimo grafe problema. Reikia rasti pigiausią būdą sujungti aibę objektų. Grandinių ir tinklų projektavimas yra praktinis minimalių jungiančiųjų medžių taikymas. Tokio medžio radimas taikomas ir kompiuterinės regos, finansų rinkos, ranka rašyto teksto atpažinimo srityse.

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

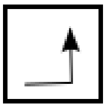
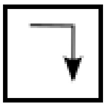
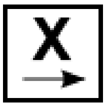
Paaiškinime pateiktas būdas, kuriuo tiesiami tiltai, remiasi tuo, kad nustatomas pigiausias (kuriam nutiesti reikia mažiausiai medžių) nenutiestas tiltas iš tų, kurie nejungia dviejų trobelių, jau sujungtų iki tol nutiestais tiltais. Tai yra tas pats, kas rinktis pigiausią nenutiestą tiltą, nesudarantį ciklinių kelių. Pavyzdžiui, tiesiamas tik vienas iš dviejų tiltų (raudonas arba violetinis). Šis procesas žinomas informatikoje kaip Kruskalo algoritmas. Kitas būdas atsakymui rasti – pigiausio tilto, jungiančio krantą ir trobelę, pasirinkimas. Tada veiksmai kartojami, renkantis pigiausią nenutiestą tiltą, jungiantį trobelę, pasiekiamą tik vandeniu, su trobele, pasiekiama tiltais. Procesas baigiamas, kai kiekviena trobelė pasiekiama tiltais. Ši procesas vadinamas Primo algoritmu.

Reikšminiai žodžiai

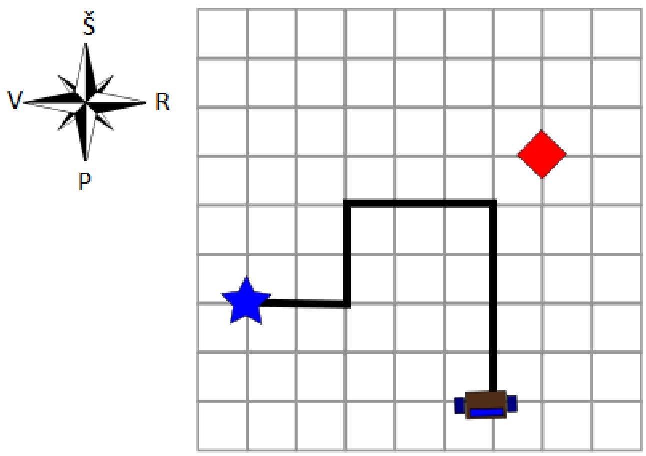
Grafas, minimalus jungiantysis medis, Primo algoritmas, Kruskalo algoritmas

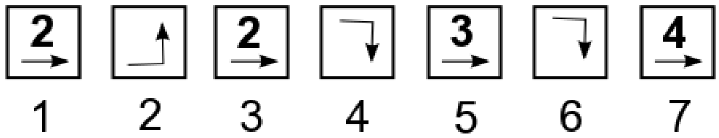
21. Klaidų paieška

Roboto veiksmai valdomi trijų rūšių mygtukais:

	posūkis į kairę
	posūkis į dešinę
	X žingsnių (langelių) į priekį

Robotas, pasisukęs į rytus, pradeda judėti iš taško, pažymėto mėlyna žvaigždute. Jonas norėjo penkiomis komandomis robotą nuvaryti į raudonu rombu pažymėtą tašką, bet netyčia paspaudė dar du mygtukus ir į rombą nepataikė. Visi septyni jo paspausti mygtukai piešinyje surašyti paspaudimo eilės tvarka.





Spustelėkite dviejų komandų mygtukus, kuriuos reikėtų pašalinti, kad robotas pasiektų raudoną rombą.

Paaiškinimas

Robotas turi žengti vertikaliai tris žingsnius. Šiam atstumui įveikti yra vienintelė komanda „3“, gaunama 5-uoju mygtuku. Prieš ją atlikdamas, robotas turi žiūrėti į šiaurę, todėl reikia pašalinti prieš ji esantį 4-ąjį mygtuką. Gauname gerą kryptį, bet robotas nueis per toli. Akivaizdu, kad reikia pašalinti 3-ąjį mygtuką, be reikalo robotą pastumiantį į šiaurę. Dabar galime įsitikinti, kad paspaudus likusius mygtukus (1, 2, 5, 6 ir 7) robotas pasieks raudoną rombą.

Tai informatika!

Kompiuteriams, panašiai kaip ir robotams, veiksmai nurodomi komandomis. Tik komandų būna daugiau ir jos sudėtingesnės. Iš jų sudaromos programos, turinčios šimtus ir tūkstančius komandų. Todėl klaidų daro net ir labai aukštos kvalifikacijos programuotojai. Svarbu mokėti rasti ir taisyti klaidas. Klaidos kompiuterių programose dažnai vadinamos švelniau – riktais, o jų paieška ir taisymas – programų derinimu arba testavimu.

Kiekvienas esame patyrę nemalonumų, kai programa sutrinka. Taip dažniausiai atsitinka dėl klaidų programose. Klaidos gali būti ir didelių nelaimių ar nuostolių priežastimi, pavyzdžiui, kai kompiuteriai naudojami ligoninėse arba valdo į kosmosą siunčiamas raketas. Tokiose situacijose programų derinimas ir testavimas ypač svarbūs.

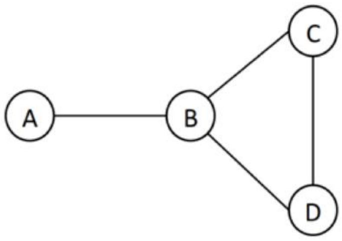
Ne kiekvienas būsime programuotojas. Tačiau kiekvienas galime paieškoti klaidų pamėgtose programose, pranešti apie jas programų autoriams, kad jas ištaisytų, ir taip prisidėti prie programų kokybės gerinimo.

Reikšminiai žodžiai

Programavimas, programų derinimas, klaidų paieška, algoritmas

22. Draugystė

Abu paveikslai paaiškina tuos pačius bebravietėje gyvenančių bebrų draugystės ryšius. Pavyzdžiui, bebras A turi tik vieną draugą – bebrą B (žinoma, ir bebras B yra bebro A draugas). Jei bebras A norėtų susidraugauti su bebru C, turėtų prašyti rekomendacijos iš bebro B.



	A	B	C	D
A		O		
B	O		O	O
C		O		O
D		O	O	

Lentelėje parodyti 7 bebrų draugystės ryšiai.

	A	B	C	D	E	F	G
A		O	O	O			
B	O		O	O			
C	O	O		O			
D	O	O	O		O		
E				O		O	O
F					O		O
G					O	O	

Kiek mažiausiai rekomendacijų reikia gauti bebrui A, norinčiam susidraugauti su bebru G?

(A) 4

(B) 3

(C) 2

(D) tik vienos

Paaiškinimas

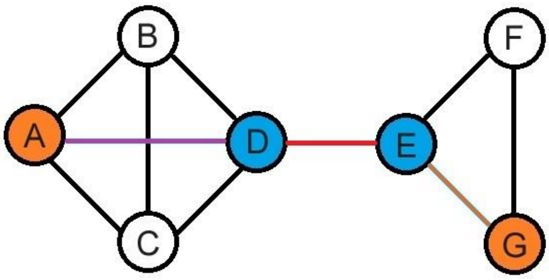
Teisingas atsakymas – C.

Neužteks tik vienos rekomendacijos, nes nė vienas bebro A draugas nėra susidraugavęs su bebru G.

Bebrui A reikia bent dviejų rekomendacijų – tai parodyta lentelėje.

	A	B	C	D	E	F	G
A		O	O	●			
B	O		O	O			
C	O	O		O			
D	O	O	O		O		
E				●		O	●
F					O		O
G					O	O	

Galime lentelės ryšius pavaizduoti aiškesne schema – socialiniu tinklu:



Taigi bebrui A, norinčiam susidraugauti su bebru G, reikia prašyti rekomendacijų bent iš dviejų bebrų – D ir E.

Tai informatika!

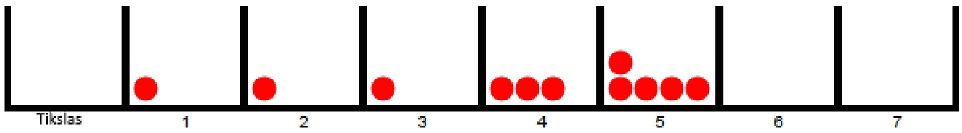
Tinklas (arba grafas) yra vizuali schema, skirta žmogui. Tačiau programose grafas vaizduojamas gretimumo matrica. Sudarę šią matricą, toliau galime taikyti visus grafų teorijos algoritmus. Matrica taip pat galime transformuoti grafo viršūnes ir briaunas. Taigi grafo ir gretimumo matricos transformacija yra svarbus informatikos mokslo konceptas.

Reikšminiai žodžiai

Tinklas, grafas, gretimumo matrica

23. Akmenukų žaidimas

Žaidimo tikslas – perkelti visus akmenukus į kairįjį lovelį (po juo jo parašyta **Tikslas**). Reikia laikytis šios taisyklės: lovelį su numeriu nuo 1 iki 7 galima ištuštinti, jei jo akmenukų skaičius yra lygus lovelio numeriui, tada jo akmenukai paskirstomi po vieną į visus kairiau esančius lovelius. Žaidimas laimimas, jei įmanoma visus akmenukus sukelti į lovelį **Tikslas**. Pateiktoje situacijoje, norint perkelti visus akmenukus į lovelį **Tikslas**, reikia atlikti šią ėjimų seką: 1—5—1—2—1—4—1—3—1—2—1.



Koks galimas didžiausias akmenukų skaičius, kai žaidimas laimimas su 7 loveliais (neįskaitant kairiojo lovelio *Tikslas*)?

Paaiškinimas

Šis žaidimas – tai vieno seniausių pasaulio žaidimų, mankalos, kildinamo iš Afrikos, variantas, kai žaidžia vienas žmogus. Žaidimas laimimas tik tada, jei strategija, kuria laimima, išlaikoma iki pabaigos. Pagal šią strategiją reikia rinktis mažiausio numerio lovelį, kurį galima ištuštinti. Iš tikrųjų pamąstykime: jei galima ištuštinti du lovelius, tai pirmiau ištuštinus didesnio numerio lovelį, mažesnio numerio lovelis liktų užblokuotas. Pavyzdžiui, duotame paveiksle galima rinktis tuštinti vieną iš lovelių, kurių numeriai 1 ir 5. Deja, pasirinkus pastarąjį, lovelis su numeriu 1 būtų užblokuotas.

Kad rastume didžiausią akmenukų skaičių, žaisti turime nuo galo, darydami atbulinį veiksmą – į mažiausio numerio lovelį su numeriu N įdedant N akmenukų, ir iš visų mažesnių numerių lovelių išimant po vieną. Taip gauname tokią seką: 1 → 0 2 → 1 2 → 0 1 3 → 1 1 3 → 0 0 2 4 → ... → 1 2 2 4 → 0 1 1 3 5 → 1 1 1 3 5 → 0 0 0 2 4 6 → ... → 1 1 3 2 4 6 → 0 0 2 1 3 5 7 → ... → 1 2 2 1 3 5 7.

Tai informatika!

Sprendžiant šį uždavinį, lemiamą vaidmenį vaidina veiksmų seka. Pateikiama laimėjimo strategija yra ne kas kita, kaip sąrašas algoritmo žingsnių, kuriuos atliekant žaidimas laimimas, jei tik jis gali būti laimimas. Algoritmo teisingumas gali būti įrodomas, pasitelkus matematinę indukciją.

Samprotavimo eiga. Jei yra tik vienas lovelis (numeris 1), aišku, kad žaidimas gali būti laimėtas, jei tame lovelyje yra tik vienas akmenukas, kitaip žaidimas nelaimimas. Toliau tarkime, kad pateiktas algoritmas teisingas N loveliams. Esant N+1 loveliui, reikia nagrinėti 3 atvejus: 1) (N+1)-ame lovelyje yra mažiau nei N+1 akmenukų, tada žaidimas nelaimimas; 2) (N+1)-ame lovelyje yra daugiau nei N+1 akmenukų, tada žaidimas nelaimimas; 3) (N+1)-ame lovelyje yra lygiai N+1 akmenukų, tada akmenukai gali būti paskirstyti (ir lovelis ištuštintas), kai nieko nebegalima daryti su loveliais 1, 2, ..., N. Iš tikrųjų N-ame lovelyje šiuo momentu turi būti tiksliai N-1 akmenukas. Kai (N+1)-ojo lovelio akmenukai perkeliami, (N+1)-asis lovelis ištuštinamas, paliekant akmenukus tik pirmuose N lovelių. Kadangi darėme prielaidą, kad algoritmas tinka esant N lovelių, tai, remiantis indukcija, šis algoritmas tinka ir esant N+1 lovelių.

Reikšminiai žodžiai

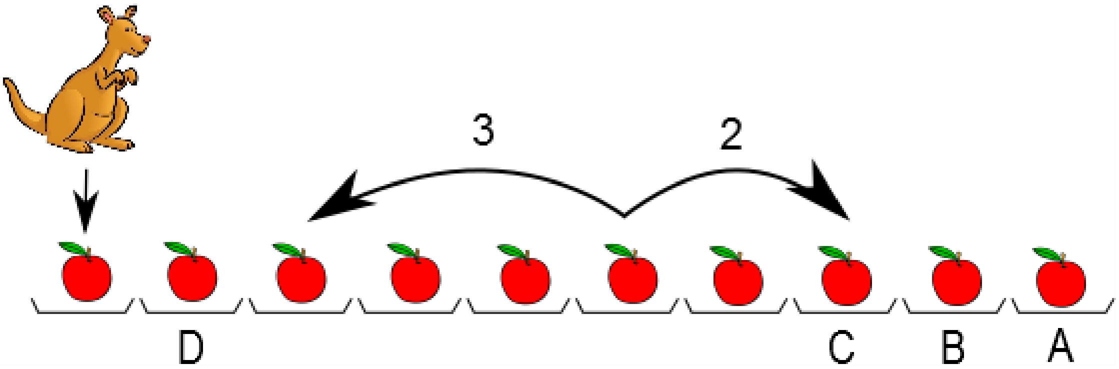
Indukcija, laimėjimo strategija

24. Šokinėjanti kengūra

Į eilę išdėliota 10 lėkščių. Kiekvienoje lėkštėje yra po vieną obuolį.

Kengūriukas Tomas mėgsta šokinėti. Pirmiausia jis šoka prie kairiosios lėkštės. Po to kiekvienu šuoliu jis gali šokti arba per dvi lėkštes pirmyn, arba per tris lėkštes atgal. (Paveiksle rodyklės vaizduoja galimus šuolius nuo vienos iš lėkščių.)

Tomas šoka tik prie lėkščių, kuriose yra obuolių. Prišokęs prie lėkštės, pasiima iš jos obuolį.



Jei Tomas susirinks visus obuolius, kurį paskutinį jis paims?

A. Pirmą obuolį iš dešinės

B. Antrą obuolį iš dešinės

C. Trečią obuolį iš dešinės

D. Antrą obuolį iš kairės

Paaiškinimas

Teisingas atsakymas – B.

Sunumeruokime lėkštes iš kairės į dešinę nuo 1 iki 10. Tada Tomas visus 10 obuolių susirinks tokia tvarka: 1, 3, 5, 2, 4, 6, 8, 10, 7, 9.

Tokia tvarka yra vienintelė galima. Kodėl? Tomas pradeda nuo 1-os lėkštės, po to šoka prie 3-os, o toliau prie 5-os lėkštės, nes, šokdamas į kairę, nušoktų kairiau pirmos lėkštės. Toliau jis šoka prie 2-os lėkštės, nes tik nuo 5-os lėkštės gali prišokti prie 2-os lėkštės, o vėliau prie jos grįžti nebegalės. Tokiu pat būdu galima aiškinti ir tolesnius Tomo šuolius kaip vienintelius galimus.

Tai informatika!

Vienas iš šio uždavinio sprendimo būdų yra visų galimų lėkščių sekų išnagrinėjimas ir radimas sekos, kuri atitiktų galimus šuolius. Kiekviena galima seka yra vadinama kėliniu, o jų yra labai daug. Toks metodas vadinamas visišku perrinkimu ir reikalauja daug laiko.

Kitas būdas – sudėlioti kėlinį po vieną lėkštę. Jei tik suvokiama, kad kėlinys ar jo pradžia negalima, pvz., nustatoma, kad antra lėkštė gali būti tik su numeriu 3, galima pašalinti paskutinę lėkštę (paskutines lėkštes) ir toliau kurti naujus kėlinius. Tai – grįžimo metodas. Jei galima kėlinius atmesti iš anksto, daug greičiau surandami teisingi kėliniai. Toks atmetimo metodas vadinamas „genėjimu“ (angl. *pruning*).

Galima šį uždavinį nagrinėti kaip grafą. Lėkštes galima laikyti grafo viršūnėmis ir sujungti briaunomis, jei Tomas gali tarp jų šokinėti. Užduotis tampa kelio, judant briaunomis, radimu, kai kiekviena viršūnė aplankoma lygiai vieną kartą. Tai Hamiltono maršrutas. Paprastai rasti tokį maršrutą labai sudėtinga. Šiuo atveju grafas yra mažas ir turi tam tikrų savybių.

Hamiltono maršruto radimas yra „NP sudėtingumo“ uždavinys. Jis priskiriamas rinkiniui uždavinių, kurie labai svarbūs ir dar neturi sprendinių. Įdomu tai, kad, jei kas nors rastų efektyvų vieno iš šių uždavinių sprendimą, tai iš karto turėtume metodą efektyviai išspręsti juos visus.

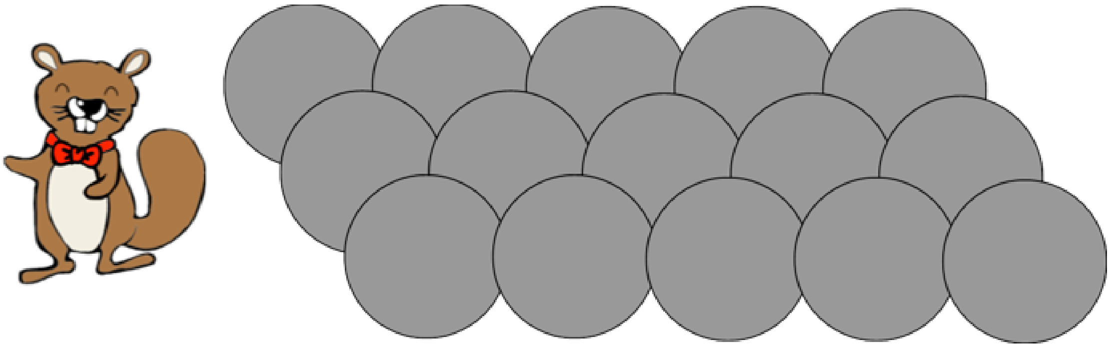
Reikšminiai žodžiai

Visiškas perrinkimas, grįžimo metodas, genėjimo metodas, grafas, Hamiltono maršrutas

25. Akmenukai

Jūs ir bebras žaidžiate žaidimą su akmenukais. Kiekvieno ėjimo metu žaidėjas gali paimti 1, 2 arba 3 akmenukus. Laimi tas, kuris paima paskutinį akmenuką.

Šiuo atveju žaidimas buvo žaidžiamas su 9 akmenukais ir paaiškėjo, kad žaidimą pradėjęs paimdamas 1 akmenuką žaidėjas visada laimi, kad ir kaip elgtųsi kiti žaidėjai. Dabar žaisite su 15 akmenukų.



Jūs pradedate žaidimą. **Kiek akmenukų paimsite, kad tikrai laimėtumėte žaidimą?**

A) 1

B) 2

C) 3

D) Nėra tokios strategijos, kad galėtumėte tikrai laimėti žaidimą

Paaiškinimas

Teisingas atsakymas – C.

Turite imti tris akmenukus. Jei po bebro ėjimo lieka 1, 2 arba 3 akmenukai, tiesiog paimate juos ir laimite. Tačiau, jei yra 4 akmenukai, laimi bebras, nes jis pasiima visus akmenukus po jūsų ėjimo, kad ir kiek akmenukų imtumėte. Taigi jūs galite laimėti žaidimą, palikdami bebrą su 4 akmenukais, kai yra likę 5, 6 arba 7 akmenukai. Vadinasi, jei paliksite bebrui tokį skaičių akmenukų, kuris yra 4 kartotinis, tai jam liks 4 akmenukai ir jūs laimėsite.

Tai informatika!

Šis uždavinys yra susijęs su lošimų teorija, kuri nagrinėja strateginį sprendimų priėmimą. Lošimų teorijos pavyzdžių galima rasti mūsų kasdieniame gyvenime, pavyzdžiui, kai kompanijos varžosi dėl rinkos pasidalijimo arba kai politikai varžosi rinkimuose. Lošimų teorija taip pat taikoma ekonomikoje, biologijoje, socialiniuose moksluose, informatikoje.

Reikšminiai žodžiai

Lošimų teorija, sprendimų priėmimas, laimėjimo strategija

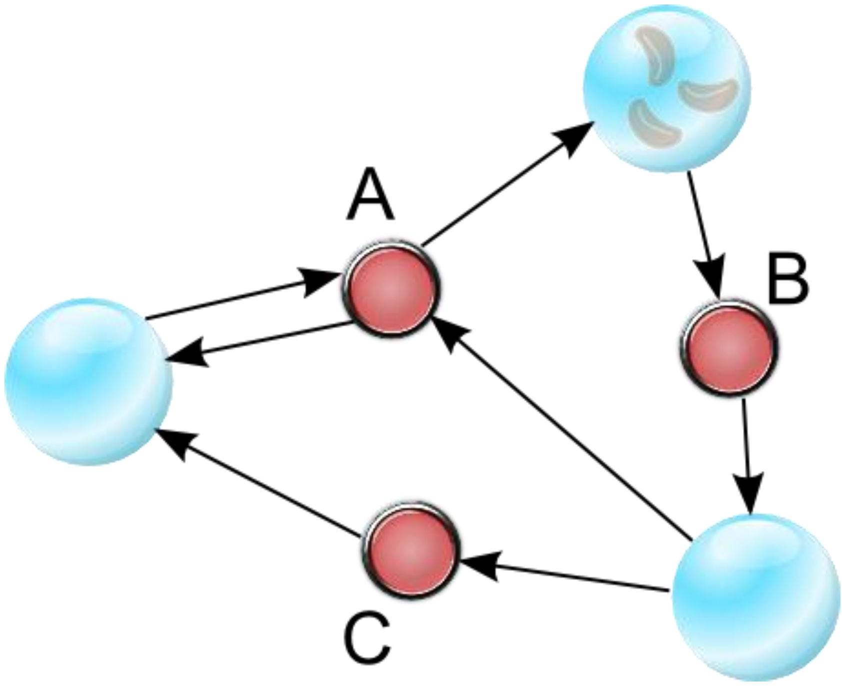
26. Stebuklingas įrenginys

Bebras Bronius apžiūri įdomų įrenginį. Jį sudaro stikliniai rutuliai, kuriuose gali būti pupelių. Rutuliai sujungti, be to, valdomi mygtukais A, B ir C.

Paspaudus bet kurį mygtuką, paeiliui atliekami šie du veiksmai:

- įrenginys tikrina, ar visuose rutuliuose, iš kurių rodyklės rodo į paspaustą mygtuką, yra nors po vieną pupelę;
- jei taip, tada iš šių rutulių išimama po vieną pupelę, o į kitus rutulius, į kuriuos rodo rodyklės iš paspausto mygtuko, pridedama po vieną pupelę.

Pavyzdžiui, paspaudus mygtuką B, viena pupelė išimama iš viršutinio rutulio, o į apatinį rutulį viena pupelė pridedama.



Kuri mygtukų paspaudimų seka sukuria tokią įrenginio būseną, kuri nebekinta spaudžiant bet kuriuos mygtukus?

A) B—B—C—A—B—A

B) B—C—B—C—B—A

C) B—B—C—B—C—C

D) B—C—B—B—A—A

Paaiškinimas

Teisingas atsakymas – C.

Nekintanti įrenginio būsena, nors spaudžiami kurie nors mygtukai, sudaro vadinamąją aklavietės situaciją. Kad įrenginys atsidurtų aklavietėje, reikia patikrinti kiekvieną mygtuką ir su į jį įeinančiomis rodyklėmis susietus rutulius. Tada reikia ištuštinti bent vieną kiekvieno mygtuko rutulį. Pastebime, kad aklavietėje atsiduriame, kai viršutinis ir apatinis rutuliai ištuštėja – iš jų rodyklės rodo į visus tris mygtukus. Vadinasi, reikia bandyti perkelti visas pupeles į kairįjį rutulį. Tai įmanoma padaryti, tris kartus paspaudus mygtukus B ir C (galimos ir kitokios šių mygtukų kombinacijos).

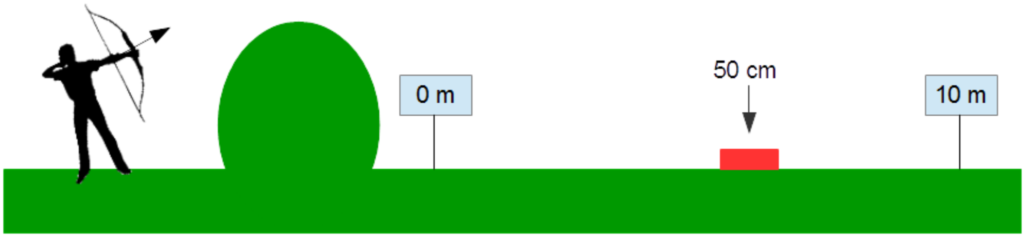
Spaudžiant kitus mygtukus, į aklavietę nepatenkama.

Tai informatika!

Šiuo uždaviniu supažindinama su Petri tinklais. Petri tinklas – formali sistema lygiagrečioms, sąveikaujančioms sistemoms aprašyti, taip pat jų elgsenai imituoti ar jų atliekamiems veiksmams analizuoti. Gebėjimas imituoti Petri tinklą padeda geriau suprasti, kaip modeliuoti kompleksines lygiagrečiai sąveikaujančias sistemas.

27. Šaudymas į taikinį

Lankininkas Arnoldas nori pataikyti strėle į taikinį, kurio nemato. Jis gali suderinti lanką strėlei paleisti norimu atstumu tarp 0 ir 10 metrų. Po kiekvieno šūvio jo draugas Markas patikrina ir praneša, ar strėlė įsmigo į žemę prieš taikinį ar už jo.



Taikinys yra 50 cm pločio. **Kiek mažiausiai strėlių reikia iššauti, kad tikrai būtų pataikyta į taikinį, kad ir kokioje vietoje jis būtų?**

A. 3

B. 4

C. 5

D. 6

Paaiškinimas

Teisingas atsakymas – C.

Geriausia strategija – dvejetainė paieška. Pirmasis šūvis paleidžiamas, taikant į 5 metrų atstumą. Tokiu būdu šaudymo laukas padalijamas į dvi 5 metrų pločio sritis. Taikinys arba kliudomas, arba jis yra vienoje iš tų dviejų sričių. Antru šūviu taikoma 2,5 metro arba 7,5 metro atstumu – šaudymo laukas padalijamas į dvi 2,5 metro pločio sritis. Taikinys vis dar gali likti nekliudytas. Trečias šūvis sumažina šaudymo lauką į dvi 1,25 metro pločio sritis, ketvirtuoju šūviu šaudymo sritys sumažėja iki 0,625 metro. Šaunant penktą kartą, šaudymo laukas padalijamas į dvi 0,3125 metrų pločio sritis. Tai mažiau už taikinio plotį, vadinasi, taikinys tikrai turi būti kliudytas.

Likusieji atsakymai neteisingi.

Tai informatika!

Šiuo uždaviniu supažindinama su dvejetaine paieška, t. y., algoritmu elementui rasti, esant O(log *n*) sudėtingumui laiko atžvilgiu, kiekvienu žingsniu dukart sumažinant paieškos erdvę.

Reikšminiai žodžiai

Dvejetainė paieška, dalijimo pusiau metodas

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“ 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.