

S
i
d
r
i
s
k
l
e



Informatikos ir informatinio mąstymo uždavinių rinkinys

Nr. 11



Šiame rinkinyje pateikiami XIX informatikos ir informatinio mąstymo konkurso (iššūkio) „Bebras“ I etapo, vykusio 2022 metų rudenį, užduotys, jų atsakymai ir paaiškinimai, koks informatikos turinys ir konceptai atskleidžiami, kaip ir kuo uždavinys įdomus ugdant informatinį mąstymą. Visi uždaviniai (įskaitant grafiką ir kitą medžiagą) licencijuojami pagal Kūrybinių bendrijų licenciją – „Creative Commons Attribution-ShareAlike 4.0 International License“. Šis uždavinių rinkinys skiriamas 1–12 klasių mokinių informatinio mąstymo gebėjimams ugdyti.

Tarptautinis informatinio mąstymo konkursas-iššūkis „Bebras“, kurį Lietuvos informatikai, vadovaujami prof. Valentinos Dagienės, pradėjo prieš dvidešimtį metų, dabar jungia 78 šalis (<https://www.bebbras.org/countries.html>). „Bebro“ konkursas rengiamas vaikams nuo 6 iki 19 metų amžiaus kasmet lapkričio–kovo mėnesiais. Konkurso tikslas yra skatinti moksleivių domėjimąsi informatika, ugdyti gilesnį technologijų supratimą, skatinti gebėjimą spręsti algoritminius, loginius uždavinius, ugdyti kritinį mąstymą, programavimo ir kompiuterinio raštingumo įgūdžius, taip pat pritraukti daugiau gabių jaunuolių į informatikos studijas.

„Bebras“ yra ne tik konkursas, bet ir mokymosi ir atradimų platforma, kuri leidžia moksleiviams ugdyti gebėjimus sprendžiant uždavinius ir gilinti savo žinias informatikos srityje. Tai naudinga ne tik moksleiviams, bet ir mokytojams, kurie gali naudoti „Bebro“ užduotis ir išteklius mokymo procese.

Dėkojame Daumilui Ardickui, dr. Vaidui Giedrimui, Nojui Gudinavičiui, Karoliui Jasučiui, dr. Eglei Jasutei, dr. Tatjanai Jevsikovai, Audronei Klupšaitėi, Alvidai Lozdienei, Aleksandrui Maliuginui, Vaidai Masiulionytei-Dagienei, Modestui Rimkui, dr. Broniui Skūpui, dr. Gabrielei Stupurienei, Tomui Šiauliui, dr. Sigitai Turskienei, Kunigundai Zubytei, Rimantui Žakauskui, talkinusiems verčiant ir adaptuojant uždavinius. Taip pat dėkojame tarptautinei „Bebro“ konkurso bendruomenei ir uždavinių autoriams.

Sudarė Lina Vinikienė

Konsultavo Valentina Dagiene

Redagavo Viktoras Dagys

Viršelį kūrė Vaidotas Kinčius



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

Įvadas

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

Pagrindinis konkurso etapas visame pasaulyje kasmet vyksta lapkričio mėnesį. 2016 m. vasario mėnesį Lietuvoje pradėtas rengti ir antrasis etapas, kuriame varžosi pirmajame geriausiai pasirodę 9–12 klasių mokiniai.

„Galime didžiulius, kad per šiuos metus būtent Lietuvoje sukurta informatikos mokymo metodika tiek išpopuliarėjo, kad šiandien taikoma 68 valstybėse ir kasmet jų skaičius tik auga. Prie jos tobulinimo prisideda žinomiausi pasaulio universitetai ir verslai. Taip pat svarbu, kad kasmet konkurse dalyvauja vis daugiau mergaičių – pernai jų buvo net 47 proc. visų dalyvių. Tai leidžia toliau griauti mitą apie informatikos mokslo nepatrauklumą mergaitėms“, – sako Vilniaus universiteto mokslininkė dr. Gabrielė Stupurienė.

Jau ne vienerius metus konkurse dalyvauja milijonai jaunuolių. 2022 metai buvo rekordiniai – 51 valstybėje dalyvavo 3,1 mln. mokinių, o vien Lietuvoje – 54 tūkstančiai iš 469 mokyklų. Net 47 proc. (25,6 tūkst.) Lietuvoje dalyvavusių mokinių buvo mergaitės.

Švietimo valdymo informacinės sistemos duomenimis Lietuvoje 2022–2023 mokslo metais mokosi apie 341 tūkst. jaunuolių, tad „Bebro“ dalyvavo maždaug 16 proc. Lietuvos mokinių.

„Bebro“ edukacines veiklas savo šalyse įgyvendina Oksfordo universitetas, Masačusetso technologijų institutas, Vaterlo universitetas ir daugelis kitų pasaulio aukštųjų mokyklų. Lietuvoje konkursą organizuoja Vilniaus universitetas, kurio Duomenų mokslo ir skaitmeninių technologijų instituto profesorė Valentina Dagienė yra „Bebro“ idėjos autorė.

„Bebro“ konkursas palaikymo sulaukia ir iš pasaulio verslų – 2017 metais Jungtinėje Karalystėje konkursą nusprendė paremti JAV technologijų milžinė „Google“. 2020 metais „Google“ skyrė dotaciją „Bebro“ organizatoriams Indonezijoje. Organizatorių teigimu, tokia parama svarbi ne tik finansiškai, bet ir idėjine prasme.

„Kai tokio žinomumo kompanijos nusprendžia palaikyti ir prisidėti prie konkurso, tai, pirmiausia, tam tikras kokybės rodiklis. Taip tarsi patvirtinama, kad konkurso metu skatiname ir mokome to, kas ateityje bus naudinga net didžiausiems pasaulio darbdaviams“, – sako G. Stupurienė.

Lietuvoje technologijų bendrovės taip pat tapo konkurso rėmėjomis. Viena jų – pasaulinė kibernetinio saugumo kompanija „Surfshark“, 2022 metų balandį kartu su „Nord Security“ tapusi antruoju Lietuvos vienaaragiu.

„Pasaulyje pastebimus rezultatus sukurti padeda ne kas kitas, o žmonės – talentingi savo srities specialistai. Remdami tokius konkursus kaip „Bebras“ prisidedame prie talentų ugdymo, kurie, be jokios abejonės, ateityje kurs pasaulinio lygio technologinius sprendimus“, – sako Simona Lubytė, „Surfshark“ atstovė žiniasklaidai. (Šaltinis: [BNS Spaudos centras](#))

Kasmet tarptautinė „Bebro“ bendruomenė susirenka kurioje nors šalyje aptarti vykdomas veiklas, atsiskaityti už nuveiktus darbus. Kiekviena šalis turi atsiųsti bent po vieną balso teisę turintį atstovą, paprastai atvyksta keletas, kurių vienas atsakingas už „Bebro“ vykdomąją veiklą, o kiti dirba uždavinių peržiūros, atrinkimo, tobulinimo grupėse.

XVIII „Bebro“ konkurso seminaras-dirbtuvės vyko 2022 m. gegužės 16–20 d. Šiaurės Makedonijoje. Iš Lietuvos į Strumicą vyko prof. Valentina Dagienė, VU Filosofijos fakulteto podoktorantūros stažuotoja dr. Gabrielė Stupurienė, VU Matematikos ir informatikos fakulteto doktorantas Daumilas Ardickas ir Alvida Lozdienė. Nuotoliniu būdu darbavosi dizaineris Vaidotas Kinčius ir sistemų kūrėjas dr. Bronius Skūpas. Strumicoje susirinko 32 šalių atstovai, daugiausia iš Europos, tačiau atvyko ir kelios delegacijos iš toliau: Indijos, Kanados, Pakistano, Pietų Korėjos, Urugvajaus. Nuotoliniu būdu užsiregistravo, dalyvavo plenarinėse sesijose ir uždavinių darbo grupėse 34 šalių atstovai.



XVIII tarptautinio „Bebro“ seminaro-dirbtuvių Strumicoje dalyviai

Vienas iš Strumicoje aptariamų klausimų buvo tarptautinės „Bebro“ asociacijos steigimas. Jau keletą metų diskutuojama šia tema, aptariami įstatai. Vilniaus universitetas, „Bebro“ sumanytojas ir pradininkas, inicijuoja tarptautinės asociacijos „Bebras International“ steigimą.

Šiuo metu „Bebro“ tinklą sudaro 55 šalys tikrosios narės ir 19 šalių kandidačių. Kiekviena šalis privalo organizuoti „Bebro“ konkurso savaitę: lapkričio mėnesį Šiaurės pusrutulio šalys ir keletas Pietų pusrutulio šalių, o dalis Pietų pusrutulio šalių – kovo mėnesį. Tačiau vien konkursu neapsiribojama, vykdoma daug įvairių renginių ir veiklų – todėl ir vadinama „Bebro“ iššūkiu.

Prieš aštuoniolika metų niekas nežinojome, kaip turi atrodyti geras informatikos uždavinys. Pirmieji uždaviniai buvo grynai techninio pobūdžio: kas gaunama paspaudus vienos ar kitos programos mygtuką, kur kokia komanda yra, ką ji daro ir pan. Informatikos mokslu rūpinosi aukštosios mokyklos, nebuvo tradicijos ieškoti patrauklių uždavinių vaikams. Vilniaus universiteto informatikai drauge su edukologais pirmieji pasiūlė įdomesnių, gilesnių informatikos uždavinių rinkinius. Dabar jau net tarptautinėje erdvėje įsigali terminas „Bebro uždavinys“ (angl. *Bebras task*), kas reiškia trumpą, glaustą, patrauklų uždavinuką, perteikiantį kurią nors informatikos sąvoką.

Kiekviena šalis turi laisvę pasirinkti veiklas, įtraukti ne tik mokinius ir mokytojus, bet ir tėvus, globėjus, mokslininkus, verslininkus. Pavyzdžiui, Indonezija subūrė 88 universitetų tinklą, kurie rengia mokymus mokytojams naudodami „Bebro“ uždavinius, juos plėtodami. Nyderlandai pasirašė sutartis su beveik dviem dešimtimis universitetų ir organizuoja ne tik pagrindinį privalomą konkursą, bet ir antrą etapą atrinktiems vyresniųjų klasių mokiniams. Sekdamos jų pavyzdžiu antrąjį etapą jau organizuoja apie pusė šalių, tarp jų ir Lietuva.

„Bebro“ iššūkiui – jau 18 metų. Kas pasiekta? Dauguma šalių ir jų mokyklų jau žino, kaip dalyvauti, ką daryti, kaip motyvuoti mokinius. Mokytojai džiaugiasi ir vertina įdomius informatikos uždavinius. Daugumoje šalių susikūrusios informatikos „Bebro“ iššūkio bendruomenės. Vienas pavyzdžių – Australijos „Bebro“ svetainė mokytojams, kurioje pateikiami, išaiškinami visi „Bebro“ uždaviniai, diskutuojama, pateikiamos didaktinės nuostatos, rekomendacijos mokytojams. Labai patogiu – tiesiog imk ir naudok (<https://digitalcareers.csiro.au/en/Bebras>).

„Bebro“ uždavinius kuria visos jį rengiančios šalys, taip sujungiamos skirtingos mokslininkų mokyklos, derinami siekiai, formuojama ateities informatikos plėtros vizija. „Bebro“ bendruomenė jau geba bendrai kurti ir tobulinti informatikos uždavinius. Dabar jau ribojama, kiek uždavinių kiekviena šalis gali pateikti (ne daugiau kaip po 7), ieškoma geresnio modelio, kad įdomius, daug balsų surinkusius uždavinius juos sukūrusios šalys įgytų teisę pateikti

daugiau uždavinių. Vis tik, nepaisant uždavinių gausos, trūksta originalių uždavinių iš naujų temų, pavyzdžiui, dirbtinio intelekto ar neuroninių tinklų.

Dauguma „Bebro“ uždavinių yra interaktyvūs, dinaminiai, pavyzdžiui, prancūzai naudoja vien šio tipo uždavinius. Tačiau tokiems uždaviniams kurti reikia priemonių, laiko jiems suprogramuoti. Reikėtų specializuoto įrankio. Vilniaus universiteto specialistų grupelė sukūrė programą „Bebras Lodge“, kuri palengvina interaktyvių uždavinių programavimą. Reikia ją tobulinti ir pritaikyti įvairioms sistemoms.

„Bebro“ konkursas pasaulio šalyse organizuojamas įvairiai: daugumoje mokiniai sprendžia individualiai, kai kas organizuoja sprendimus poromis prie vieno kompiuterio (Vokietija), keletas šalių (viena jų Italija), leidžia spręsti grupelėmis po keturis. „Bebro“ iššūkyje pabrėžiamas mokymasis, informatinio mąstymo ugdymas, o ne varžymasis, kas greičiau išspręs tam tikrą skaičių uždavinių. Konkurso rengiant laisviau, sprendimus išdėstant per savaitę ar net kelias, atsiranda sąžiningo sprendimo problema. Lietuvoje informuojame ir mokytojus, ir mokinius, kad uždavinius spręstų savarankiškai, laikydamiesi sąžiningumo taisyklių. Kiekvienas mūsų mokins, prisijungęs prie „Bebro“ varžybų sistemos, deklaruoja, kad jis spręs uždavinius savarankiškai, sąžiningai. Pastebėjome, kad ugdomi sąžiningo darbo kultūra, labai mažai atvejų, kai sprendžia visa klasė ir „nurašoma nuo lentos“.

Dauguma šalių rengia „Bebro“ uždavinių brošiūras, kuriose pateikiami išsamūs sprendimai ir metodinė medžiaga mokytojui. Žinoma, visos šalys turi savo svetaines, tad „Bebro“ uždaviniai verčiami į daugiau kaip pusšimtį kalbų. Pavyzdžiui, Indija verčia į septynias kalbas, Šveicarija – į keturias.

Daug mokslinių tyrimų siejama su „Bebro“ modeliu, nagrinėjami uždaviniai, gilinamasi į jų sudėtingumo lygius, stebimi mokinių rezultatai, mokymosi pažanga. Mokslininkų grupės ima taikyti „Bebro“ uždavinius informatinio mąstymo vertinimui. Vien pasinaudojus „Google Scholar“ ir pateikus paiešką (Bebras + informatics – computational thinking) gauname apie 800 mokslinių publikacijų, paskelbtų per pastaruosius ketvirtą metų. Pagrindines „Bebro“ publikacijas stengiamės nurodyti tarptautinėje konkurso svetainėje „Bebras.org“ (<https://www.bebas.org/publications.html>).

Šaltinis: [VU Filosofijos fak... prof. Valentina Dagienė](#)

2022 metų „Bebro“ konkurso uždavinius sprendė 3 295 494 mokiniai 60-yje valstybių.

Lietuvoje tais metais konkurse dalyvavo 53 975 mokiniai:

- 2 381 nykštukas (1–2 klasės),
- 4 362 mažyliai (3–4 klasės),
- 16 465 bičiuliai (5–6 klasės),
- 12 529 draugai (7–8 klasės),
- 13 817 jaunių (9–10 klasės),
- 4 421 kolegos (11–12 klasės).



2022 metais Lietuvoje kiekviena amžiaus grupė sprendė po 18 uždavinių, išskyrus nykštukus (12 uždavinių) ir mažylius (15 uždavinių): trečdalis buvo lengvesnių, trečdalis – vidutinio sunkumo ir trečdalis – sunkesnių. Uždaviniams spręsti skiriamos 45 minutės.

Sutartas „Bebro“ uždavinių vertinimo modelis, tačiau šalys gali jį kiek modifikuoti. Lietuvoje taškai skaičiuojami taip:

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

Daugiau informacijos apie „Bebro“ konkursą pateikiama interneto svetainėse:

- Tarptautinė „Bebro“ iššūkio svetainė: www.bebbras.org
(iš čia galite pasiekti visų dalyvaujančių šalių svetaines – spustelėkite šalies vėliavėlę)
- Lietuvos „Bebro“ svetainė: bebras.lt
- Konkurso sistema – vadinamasis „Bebro“ varžybų laukas:
lt.bebbras.lt



Lentelėje pateikiamas šio rinkinio uždavinių skirstymas pagal amžiaus grupes.

Nr.	Uždavinio pavadinimas	Uždavinio identifikatorius	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
1	Tradicinio meno mokymasis	2022-IN-01	6	6				
2	Vėrinys	2022-MK-05b	6					
3	Sutvarkyk marškinius!	2022-PK-01	6					
4	Nuotrauka	2019-LT-10	6					
5	Spalvinimo paveikslėlis	2022-AT-01a	9	6				
6	Biblioteka	2022-DE-06	9	6				
7	Kelias pas močiutę	2022-LT-08	9	6				
8	Penki saldainiai	2022-UK-02	9	6				
9	Spėlionių žaidimas	2022-CH-14	12	9	6			
10	Vėžlys ir kiškis	2022-PH-03	12	9	6			
11	Spalvota dėlionė	2022-VN-05a	12	9	6			
12	Paimk lazdeles	2022-BR-01	12	9				
13	Bičių avilys	2022-FR-02a		9	6			
14	Maisto slėptuvės	2022-CH-08		12	9	6		
15	Sumuštinio receptas	2022-KR-03		12	9	6		
16	Jūreivio vėrinys	2022-SK-03		12	9	6		
17	Širdžių grafika	2022-DE-02		12	9			
18	Miško vakarėlis	2022-IE-02		12				
19	Šokinėjimas	2022-CZ-04b			6			
20	Kur bebras?	2022-JP-02			6			
21	Antivirusinė programa	2022-HU-02			9	6		
22	Siuvimo mašina	2022-SK-02			9	6		
23	Ijunk šviesą!	2022-AU-03			12	9	6	
24	Veržlės ir varžtai	2022-CA-06			12	9	6	
25	Marijos kaimynai	2022-CY-01			12	9	6	
26	Šifras: 8	2022-SK-04			12	9	6	
27	Kilimo audimas	2022-TR-02			12	9		
28	Bebrukių užtvanka	2022-TW-06			12			
29	Saugokitės šungrybių	2022-HU-04				6	6	
30	Spalvota dėlionė	2022-VN-05b				9	6	
31	Sąrašai	2022-AT-04				12	9	6

Nr.	Uždavinio pavadinimas	Uždavinio identifikatorius	Nykštukai	Mažyliai	Bičiuliai	Draugai	Jauniai	Kolegos
32	Gobšūs troliai	2022-CH-04				12	9	6
33	Klausyk ir eik	2022-DE-03				12	9	6
34	Karuselė	2022-DE-05				12	9	6
35	Mėgstamiausias filmas	2022-DE-07				12	9	6
36	Nuliukai ir kryžiukai	2022-LV-03				12	9	6
37	Gražiausias brangakmenis	2022-CA-04					12	9
38	Kriauklės ir akmenukai	2022-IT-02					12	9
39	Lobio dėžutė	2022-MK-01					12	9
40	Dovana	2022-NL-03					12	9
41	Slaptažodžiai	2022-US-06					12	9
42	Labirintas	2022-UZ-03					12	9
43	Grindų raštas	2022-AT-02						12
44	Keturios kortelės	2022-BE-02						12
45	Magiška šalis	2022-CA-02						12
46	Bebrų žaidynės	2022-KR-06						12
47	Virusai	2022-NZ-01						12
48	Rikiavimas	2022-UA-03b						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.



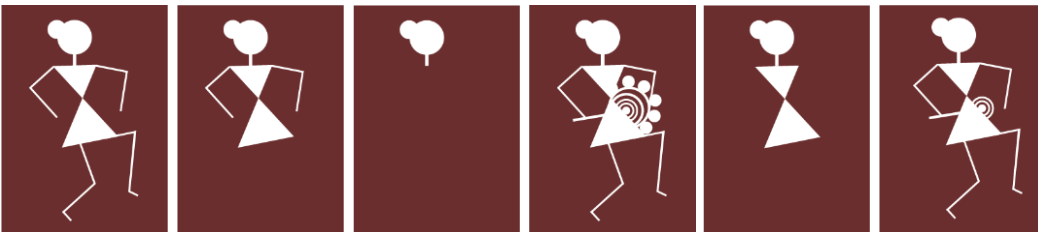
1. Tradicinio meno mokymasis

Ditė mokosi *varli* tautelės dailės: duotų kortelių pagalba ji nori sukurti tokį paveiksluką:



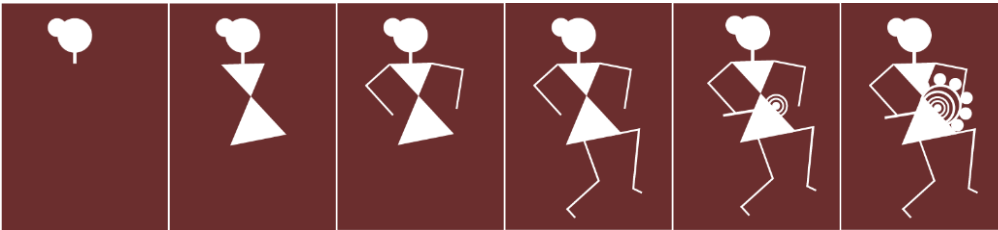
Ji turi nuosekliai sekti instrukcijas, pateiktas ant kortelių, tačiau kortelės yra sumaišytos.

Sudėkite korteles tinkama seka.



Paaiškinimas

Teisinga kortelių seka:



Kiekviena kortelė prideda po vieną dalį, kol sudedamas nurodytas paveikslėlis.

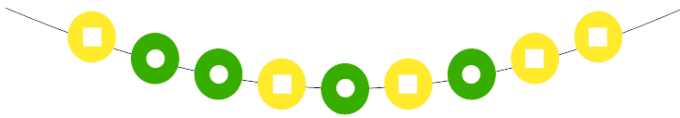
Tai informatika!

Ši užduotis yra gera iliustracija informacinių technologijų įgūdžio, vadinamo modelio atpažinimu, kuris specifiskai naudojamas rasti panašius arba bendrus elementus. Norint sudėti korteles eilės tvarka reikia lyginti kortelių poras ir surasti tas, kurios turi panašiausius modelius. Supratęs, kaip paveikslukas yra kuriamas (vis pridedant baltų geometrinių formų), galima sudėti visus paveikslukus eilės tvarka.

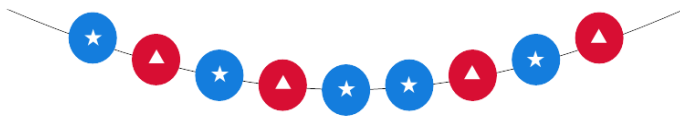
2. Vėrinys

Monika ir Veronika, grįždamos iš atostogų, parsivežė karoliukų vėrinius:

Monika



Veronika

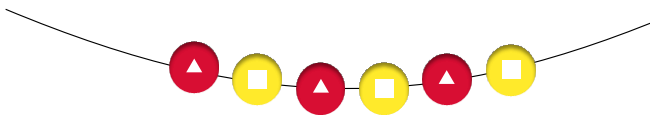


Abi mergaitės iš savo 6 karoliukų padarė vėrinį savo draugei Anastasijai. Dabar Monikos ir Veronikos vėriniai atrodo taip:

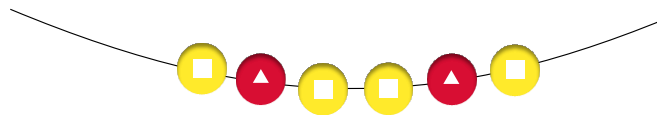


Kuris vėrinys yra Anastasijos?

A



B



C

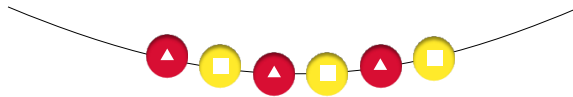


D



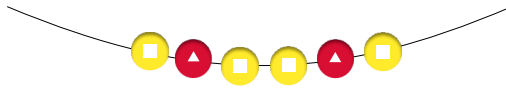
Paaiškinimas

Teisingas atsakymas yra A:



Monikos vėrinyje trūksta trijų geltonų karoliukų, o Veronikos – trijų raudonų, būtent tai yra A vėrinyje.

Atsakymas



neteisingas, nes jame yra keturi geltoni

karoliukai. Monikos karoliai prasidėjo nuo penkių geltonų karolių, o dabar jai liko du. Todėl didžiausias geltonų karoliukų skaičius, kuris gali būti Anastasijos vėrinyje, yra trys.

Atsakymas



neteisingas, nes jame yra mėlynų ir žalių

karoliukų. Monikos ir Veronikos vėriniuose lieka visi žali ir mėlyni karoliukai, todėl Anastasijos vėrinyje negali būti nei žalių, nei mėlynų karoliukų.

Dėl tos pačios priežasties atsakymas



irgi neteisingas.

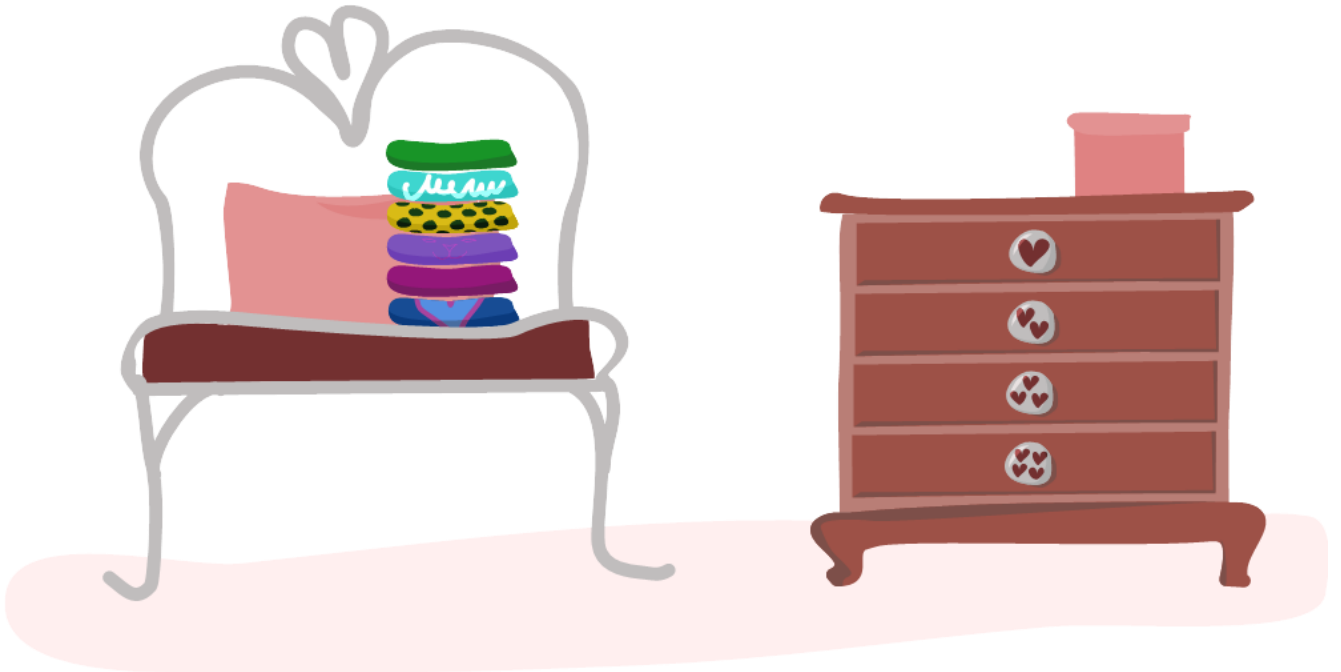
Tai informatika!

Ši užduotis iliustruoja algoritmų koncepciją, konkrečiai – kaip sukuriamas algoritmas.





Jei vaikas išsprendė šią užduotį, tikėtina, kad jis bus sugalvojęs aibių atimties algoritmą. Kai kuriose aukšto lygio programavimo kalbose, pvz., „Python“, aibė yra abstraktus duomenų tipas, leidžiantis atlikti dvivietes operacijas, pvz., atimtį. Nieko nežinodamas apie aibių matematiką, vaikas vis dėlto supras, kad bendra karoliukų kolekcija turi būti tokia pat prieš ir po šešių karoliukų panaudojimo Anastasijos vėriniui. Sukurtas kelių žingsnių algoritmas turi apimti įvairių spalvų pasikartojimų skaičiavimą, skaičių sąrašo sudarymą (arba skirtingų spalvų kartojimą) ir tarpinių palyginimų stebėjimą. Šios operacijos yra viskas, ko reikia norint atlikti daugybę naudingų kompiuterių mokslo algoritmų. Ši užduotis taip pat iliustruoja kompiuterinio mąstymo abstrakcijos ir reprezentacijos sąvokas. Tikėtina, kad, norėdamas teisingai išspręsti šią užduotį, kiekvienam vėriniui vaikas turi sukurti atvaizdą, paremtą spalvų skaičiumi (Monikos vėrinys turi penkis geltonus ir keturis žalius) ir abstrahuoti tokias detales, kaip spalvos padėtis.

3. Sutvarkyk marškinius!








Ant Pajautos lovos sudėta šešių švarių marškinių krūva. Pajauta turi sudėti marškinius į keturis komodos stalčius. Ji ima po vienus marškinius nuo krūvos viršaus ir deda į stalčius, pradėdama nuo viršutinio stalčiaus: pirmus marškinius į viršutinį stalčių, antrus – į antrąjį nuo viršaus ir t. t.. Kai Pajauta įdeda marškinius į apatinį stalčių, kitus ji vėl pradeda dėti nuo viršaus.



Į kurį stalčių ji įdės paskutinius marškinius?

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

Paaiškinimas

Pajauta pirmuosius marškinius deda į viršutinį stalčių . Antrieji marškiniai dedami į stalčių . Tada tretieji ir ketvirtieji marškiniai dedami į stalčius  ir . Kadangi stalčius  yra apatinis, tai penktieji marškiniai vėl dedami į viršutinį stalčių . Galiausiai šeštieji marškiniai yra dedami į žemiau esantį stalčių .

Tai informatika!

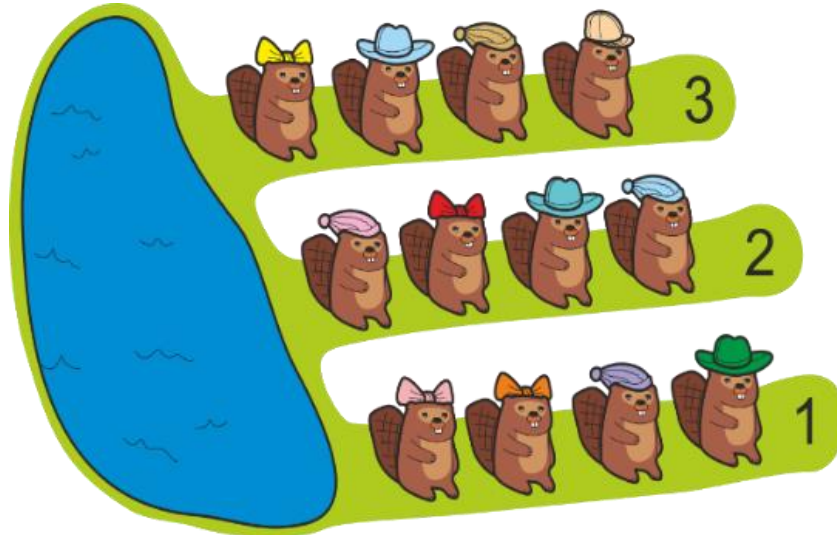
Kiek bus 6 padalijus iš 2? Žinoma, kad 3. Taip pat 10 padalijus iš 5 gauname 2, 8 padalijus iš 2 – 4 ir t. t. Tačiau ne visi dalijimai tokie paprasti: 7 padalijus iš 3 negauname sveikojo skaičiaus – gaunate skaičių 2 ir liekaną 1.

6 padalijus iš 4 gauname 1 ir liekaną 2. Skaičius 2 ir reiškia šios užduoties antrąjį stalčių. Taigi, atliekant šią užduotį reikia rasti skaičių dalybos liekaną. Informatikos moksle šis skaičiavimas vadinamas modulio operacija (angl. *modulo operation*).

Modulio operacija (liekanos radimas) labai svarbi informatikoje. Saugus ryšys, pavyzdžiui, tarp jūsų saityno naršyklės ir svetainės serverio, naudoja šią operaciją. Be to, dideliuose tinkluose, kur kelios linijos jungia tuos pačius du kompiuterius, modulio operacija siunčiami duomenys tolygiai paskirstomi tarp tų linijų. Taigi bandymas vienodai paskirstyti marškinius į stalčius iš tikrųjų yra informatikai būtinas procesas.

4. Nuotrauka

Bebriukai fotografuojasi. Jie sustoję trimis eilėmis po 4 kiekvienoje.



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

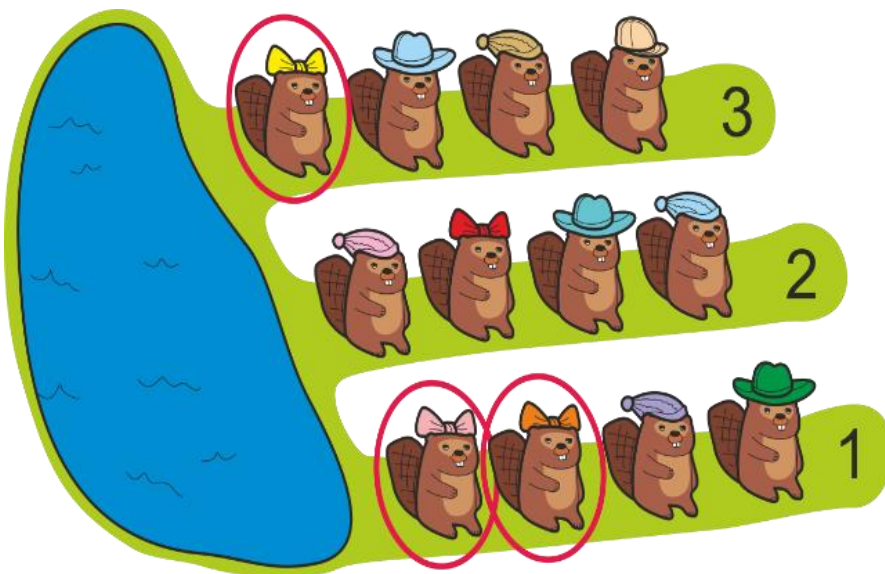
Kuri bebrukų eilė sustojo pagal mamos pageidavimą?

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

Paaiškinimas

Atsakymas: C.

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

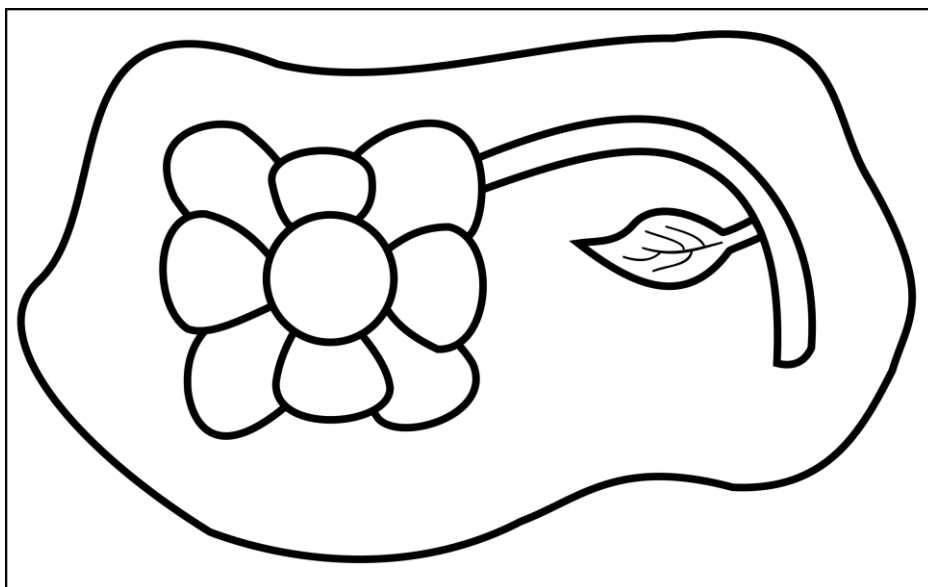


Tai informatika!

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

5. Spalvinimo paveikslėlis

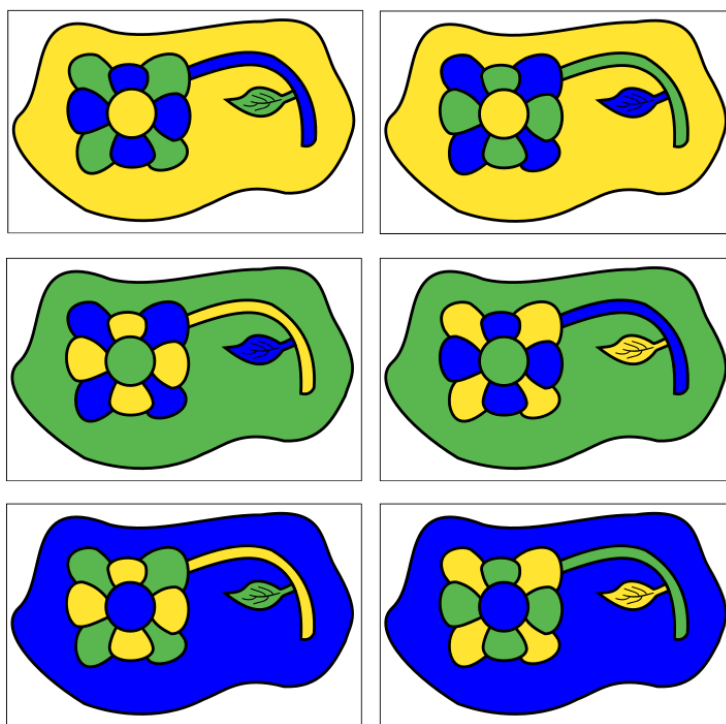
Nuspalvinkite paveikslėlį žalia, geltona ir mėlyna spalvomis taip, kad niekur nesiliestų dvi tos pačios spalvos dalys.



Paaiškinimas

Pateikiame vieną iš variantų kaip galima nuspalvinti paveiksluką.

Pirmiausia pasirinkite spalvą išorinei daliai, šiuo atveju geltoną. Visos kitos dalys, išskyrus gėlės centrą, liečia šią dalį, todėl jos turi būti mėlynos arba žalios spalvos. Pradėkite nuo bet kurios dalies ir spalvinkite keisdami spalvas (žalią ir mėlyną), kol liks nenuspalvintas tik centras.



Tai informatika!

Ši užduotis tradiciškai yra vadinama „spalvų priskyrimu grafo sritims“, šiuo atveju grafas yra gėlė. Matematikoje ir informatikoje tokie uždaviniai vadinami grafo spalvinimo uždaviniais.

Grafo spalvinimo uždaviniai turi daug panaudojimų. Spalvos naudojamos pasaulio žemėlapiuose, siekiant atskirti šalis. Kiti grafo spalvinimo uždavinių pavyzdžiai: sporto turnyro dalyvių suskirstymas į grupes, užsiėmimų planavimas klasėse, eismo valdymas. Šiais atvejais svarbu naudoti kuo mažiau spalvų.

Realiam pasaulyje spalvinimo uždaviniuose yra daug daugiau sričių nei atliekant šią užduotį ir nebeįmanoma ranka rasti galutinio varianto su keliomis spalvomis – tokiems uždaviniams išspręsti reikia kompiuterio.

6. Biblioteka

Vaikai renkasi knygas iš bibliotekos. Biblioteka yra parengusi vaizdžią populiarių knygų lentelę.

Kurią iš lentelėje pateiktų knygų vaikai pasirenka dažniausiai?

Paaiškinimas

Du vaikai prašė knygos su drugeliu. Trys vaikai prašė knygos su vikšru. Vienas vaikas prašė knygos su boružėle. Vienas vaikas prašė knygos su sraige. Iš to galima pastebėti, kad dažniausiai buvo pasirinkta knyga su vikšru.

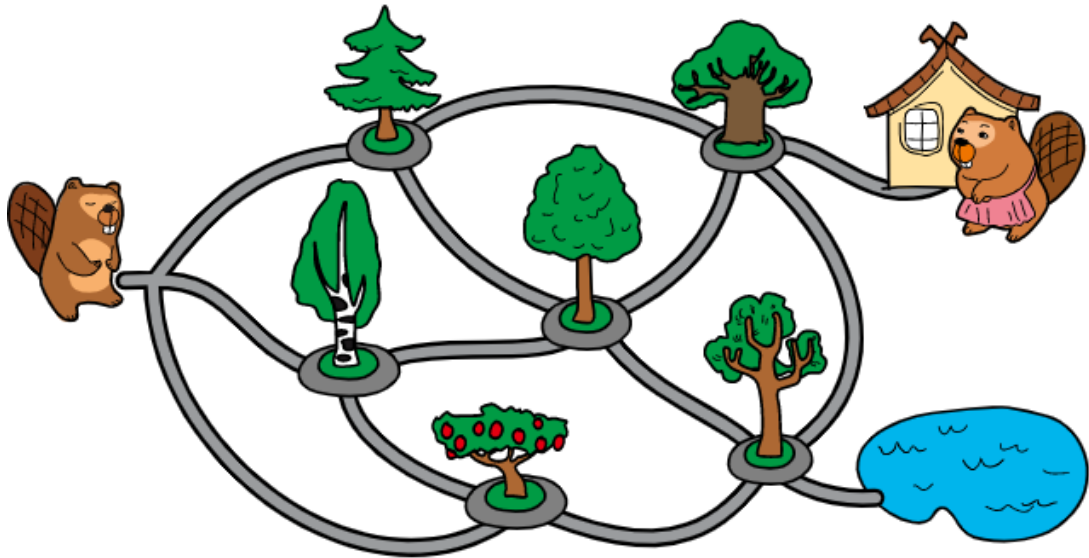


Tai informatika!

Džiugu, kad vaikai mėgsta skaityti knygas! Bet ar tikrai mums reikalinga lentelė, kad sužinotume, kuriam vaikui kuri knyga patinka? Kas būtų, jei pasirenkamas knygas žymėtume linijomis? Tiesa, tai būtų lengva ir patogu žmonėms, tačiau ne kompiuteriams. Kompiuteriams sunku aptikti linijas, tad žymiai paprasčiau duomenis identifikuoti lentelėje. Jei norime, kad kompiuteris apdorotų duomenis, pvz., kuris vaikas pasirinko kurią knygą arba koks žmogus naudoja kokį banką ir pan. Tokiu atveju geriau duomenis perkelti į lenteles. Lentelėse matome, kaip dalykai (arba vaikai) yra susiję su kitais dalykais. Kaip suskirstyti duomenis lentelėje, priklauso nuo to, kaip jie tarpusavyje susiję. Pavyzdžiui, jei būtų taisyklė, kad vaikas visada gali pasirinkti tik vieną vienintelę knygą, tada šiems duomenims suvesti būtų galima kurti lentelę ir kiekvienam vaikui skirti atskirą eilutę. Taip pat šią lentelę galima papildyti informacija, kuri būdinga tik vienam vieninteliam vaikui, pvz., vardas, akių spalva, plaukų spalva ar kita. Tačiau įprasta, kad vaikai gali pasirinkti daugiau negu vieną knygą ir tokias pačias knygas, kurias jau pasirinko kiti vaikai. Tad tokiu atveju reikalinga papildoma ir sudėtingesnė lentelė, kuri turėtų daugiau eilučių kiekvienam vaikui ir knygai.

7. Kelias pas močiutę

Bebriukas nori aplankyti savo močiutę. Ją aplankyti galima keliais skirtingais keliais.

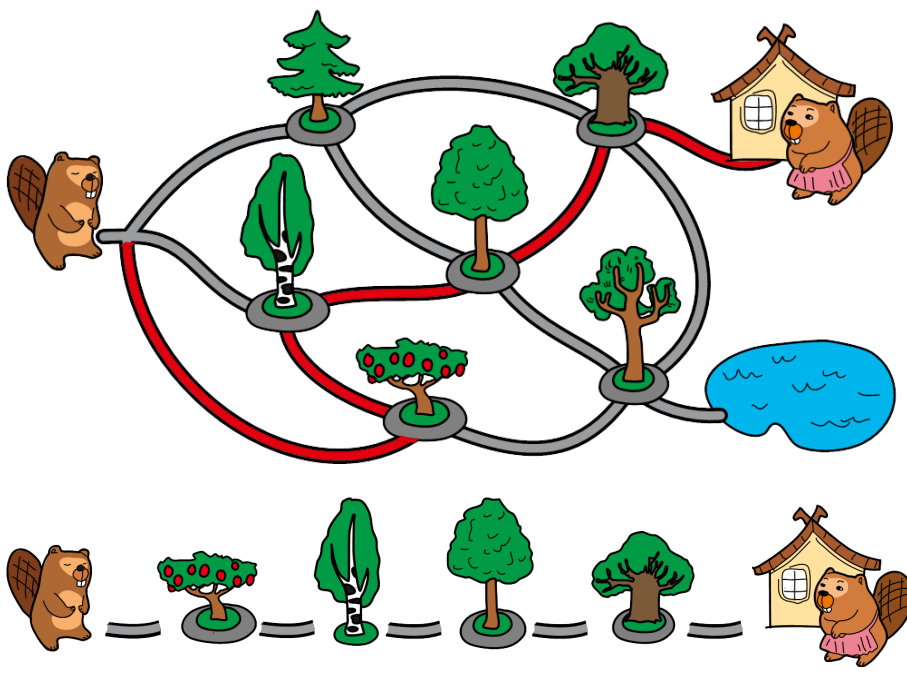
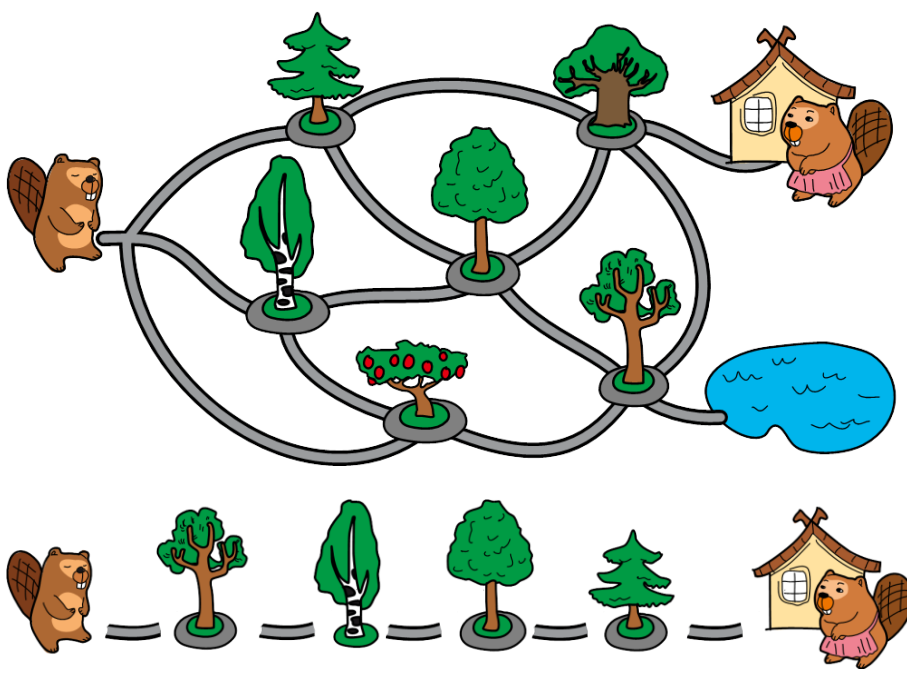



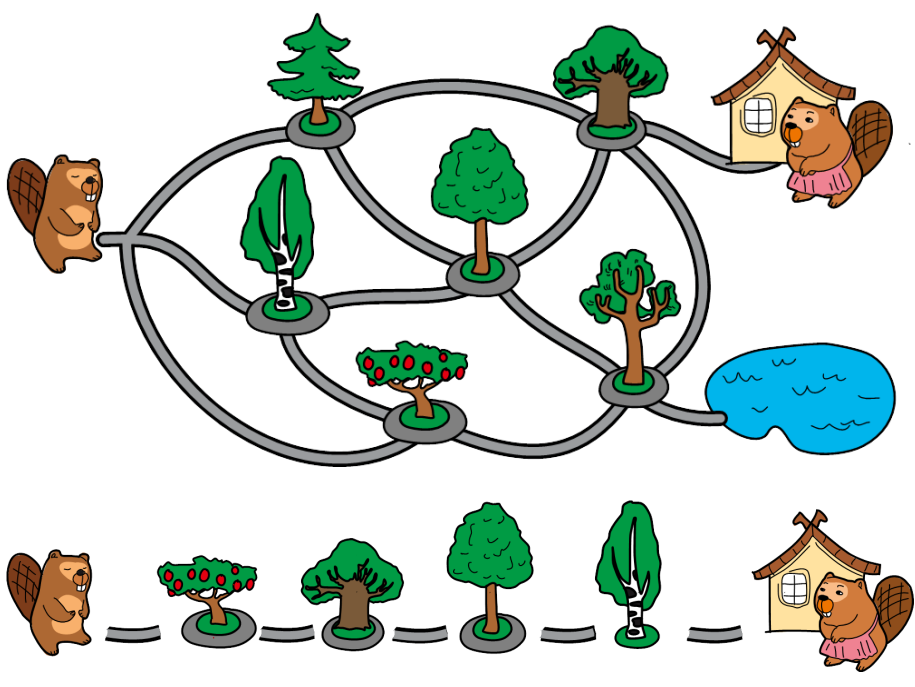

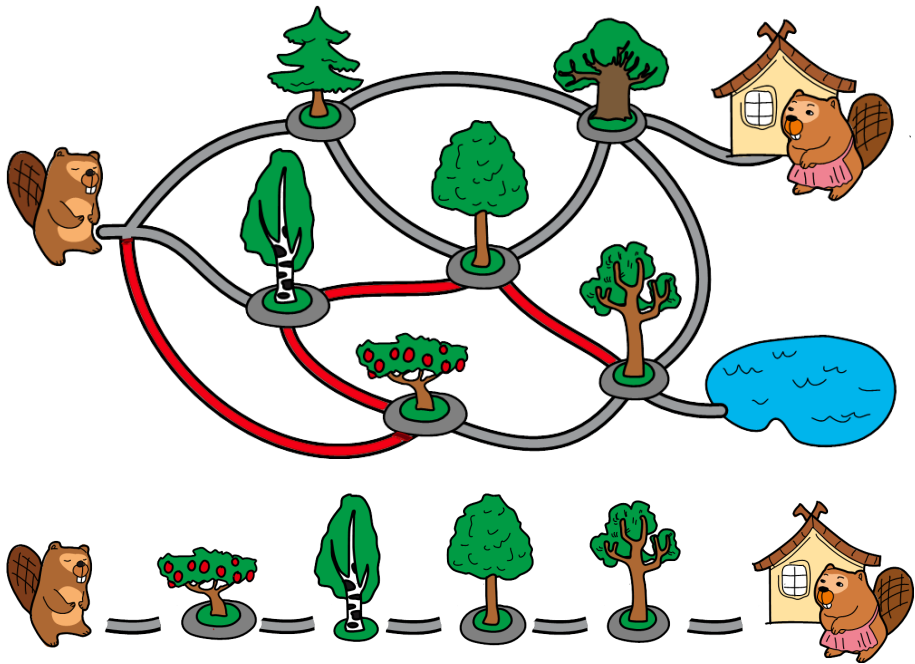
Kuris iš pateiktų kelių leis bebriukui pasiekti savo močiutę?

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

Paaiškinimas

Bebriukui nukeliauti pas močiutę yra keli būdai, bet iš duotų variantų teisingas yra tik vienas – A variantas.

<p>A</p>		<p>Teisingas atsakymas. Kelias pažymėtas raudonai.</p>
<p>B</p>		<p>Šis medis negali būti pirmas praeitas medis:</p> 

<p>C</p>		<p>Šis medis  gali būti pirmas einant link močiutės namo, bet bebrūkui nejmanoma pasiekti tolesnio po jo nurodyto medžio.</p>
<p>D</p>		<p>Šis kelias veda prie ežero, ne pas močiutę.</p>






Tai informatika!

Judėjimo planavimas (arba kelio planavimas) yra informatikos problema, kurios tikslas – rasti tinkamą konfigūracijų seką, leidžiančią perkelti objektą iš šaltinio į paskirties vietą. Šis terminas vartojamas kompiuterinėje geometrijoje, kompiuterinėje animacijoje, robotikoje ir kompiuteriniuose žaidimuose.

8. Penki saldainiai

Beno mėgstamiausi saldainiai yra penkių skonių.

Kasdien eidamas į mokyklą Benas įsideda į vamzdžio formos dėžutę po vieną kiekvieno skonio saldainį. Dienos metu Benas ima saldinius iš dėžutės nuo viršaus ir suvalgo.

Šiandien jis norėtų suvalgyti saldinius šia skonių tvarka: vynuogių , apelsinų , citrinų , braškių  ir mėlynių .

Sudėk Benui saldinius į dėžutę taip, kad jis suvalgytų juos norima tvarka.



Paaiškinimas

Siekiant saldinius išimti iš dėžutės norima tvarka, svarbu suprasti, kad pirmas įdėtas saldainis į dėžutę bus paimtas paskutinis. Tai reiškia, kad saldainiai turi būti sudėti šia tvarka:



Tai informatika!

Eilės tvarka yra svarbu. Jeigu Benas nori valgyti saldinius kita tvarka, jis ir turėtų juos sudėti į dėžutę kitu eiliškumu.

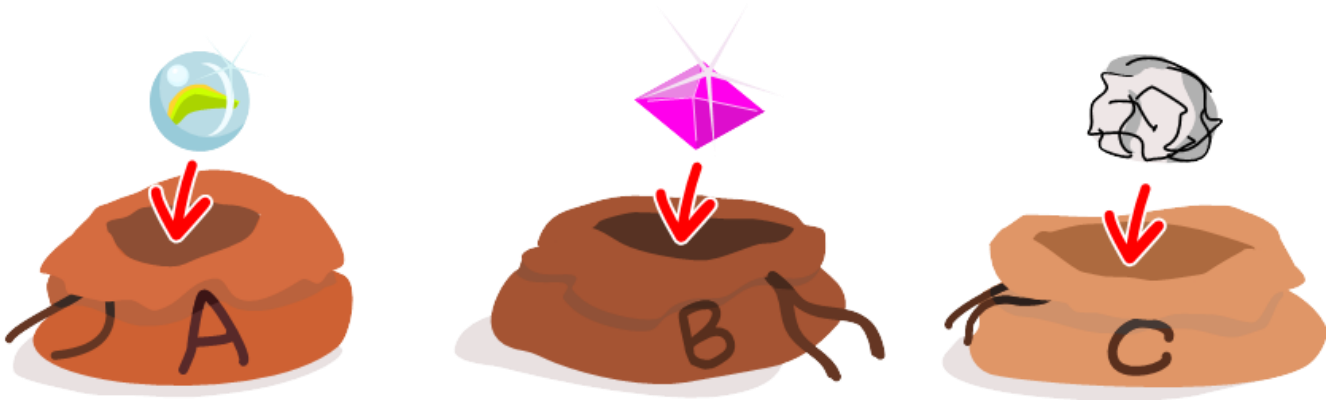
Vienas iš pirmųjų dalykų, kuriuos programuotojai išmoksta, yra tai, kaip svarbu, kad viskas būtų tinkamai išdėstyta.

Pirmiausia Benas turėjo pasirinkti tinkamą eiliškumą. Tada reikia įsivaizduoti, kas nutiks sudėjus saldinius atitinkama tvarka. Būtent taip elgiasi programuotojai, kurie yra linkę numatyti situaciją į priekį.

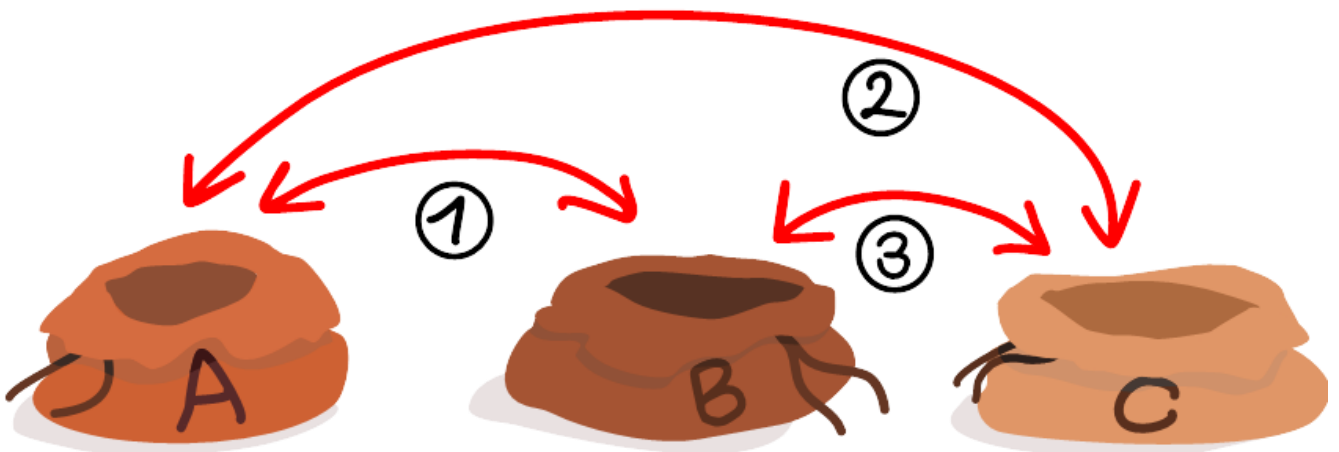
Šioje užduotyje naudojama tvarka vadinama dėklo (arba rietuvės, angl. *stack*) duomenų struktūra. Į dėklą elementai sudedami arba išimami pagal taisyklę „paskutinis į indą – pirmas iš jo“ (angl. LIFO).

9. Spėlionių žaidimas

Lilė su draugais žaidžia spėjimo žaidimą. Norėdama pradėti žaidimą, Lilė į maišelį A įdeda rutuliuką, į maišelį B – brangakmenį, o į maišelį C – suglamžytą popieriaus lapą.



Tada ji paprašo draugų užmerkti akis. Jiems užsimerkus ji sukeičia maišelių turinį. Pirmiausia ji sukeičia daiktus, esančius maišeliuose A ir B. Tada sukeičia daiktus, esančius maišeliuose A ir C. Galiausiai sukeičia daiktus, esančius maišeliuose B ir C.



Kur dabar yra Lilės daiktai?

Paaiškinimas

Lilė tris kartus sukeičia daiktus vietomis. Po pirmojo pakeitimo maišeliai atrodo taip:



Po antrojo pakeitimo maišeliai atrodo taip:



Po trečio pakeitimo maišeliai atrodo taip:



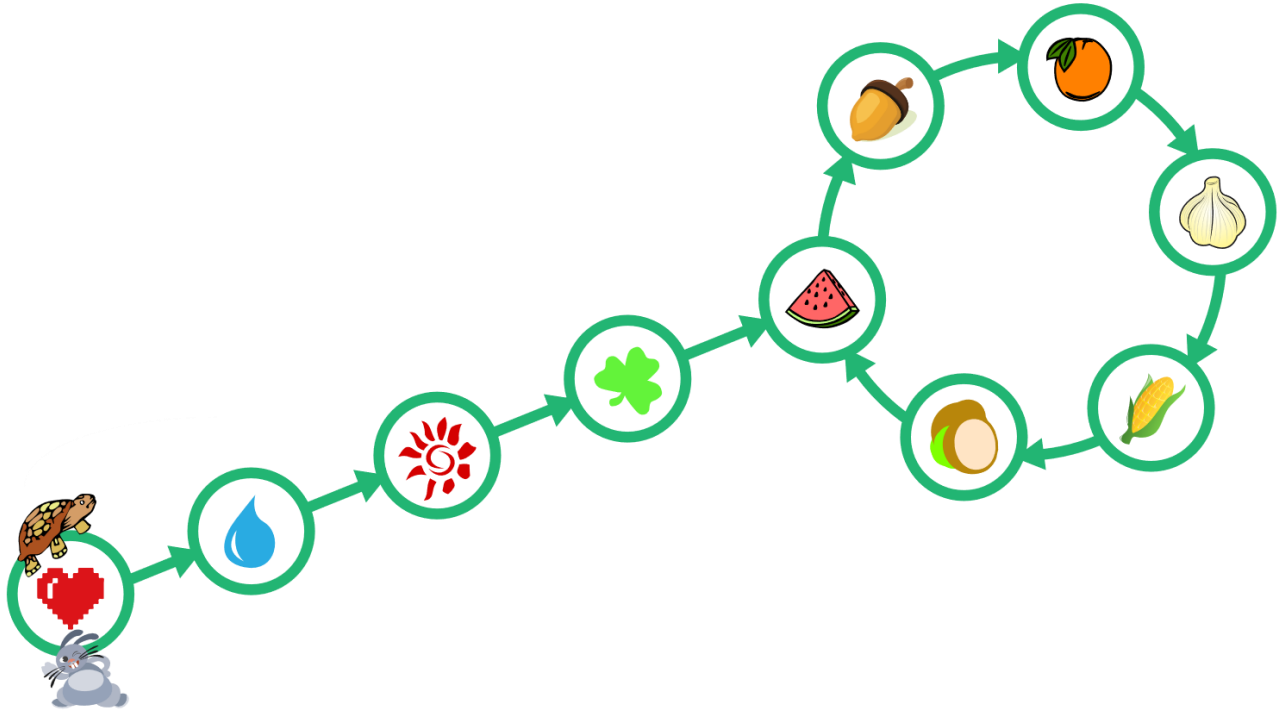
Taigi, popierius yra A maišelyje, brangakmenis – B maišelyje, o rutuliukas – C maišelyje.

Tai informatika!

Perstatymas (angl. *permutation*) – objektų išdėstymas tam tikra tvarka. Jei objektai išdėstomi kita tvarka, sukuriama kitoks perstatymas. Ta pati objektų grupė gali turėti daugybę perstatymo variantų. Šios užduoties pradžioje Lilės daiktai yra taip sustatyti: rutuliukas–brangakmenis–popierius. Užduoties pabaigoje tie patys daiktai sudaro kitokį perstatinį: popierius–brangakmenis–rutuliukas. Kiek skirtingų perstatymų galima sudaryti iš trijų objektų? Perstatymai susiję su rikiavimu. Išrikuotas sąrašas yra tiesiog vienas iš daugelio galimų sąrašo perstatymų. Rikiavimas yra dažnai sutinkamas informatikoje. Pavyzdžiui, kompiuterio aplankuose failai dažnai rikiuojami pagal pavadinimą arba datą. Sukurta daug įvairių rikiavimo algoritmų (rikiavimo metodų). Visi rikiavimo algoritmai prasideda nerikiuotu sąrašu ir visi baigiasi surikiuotu sąrašu. Jų skirtumas – kas vyksta rikiavimo proceso metu. Su sąrašo elementais bus atlikta daug perstatymų, tačiau kokie konkrečiai perstatymai vyks priklauso nuo to, koks rikiavimo algoritmas buvo panaudotas.

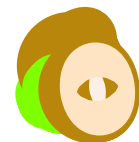
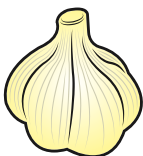
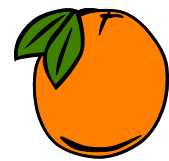
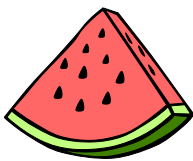
10. Vėžlys ir kiškis

Vėžlys ir kiškis lenktyniauja tarpusavyje. Trasa pavaizduota paveikslėlyje:



Jie abu startuoja širdute pažymėtame laukelyje ir juda rodyklių kryptimi. Vėžlys per minutę pajuda per vieną laukelį, kiškis – per du laukelius.

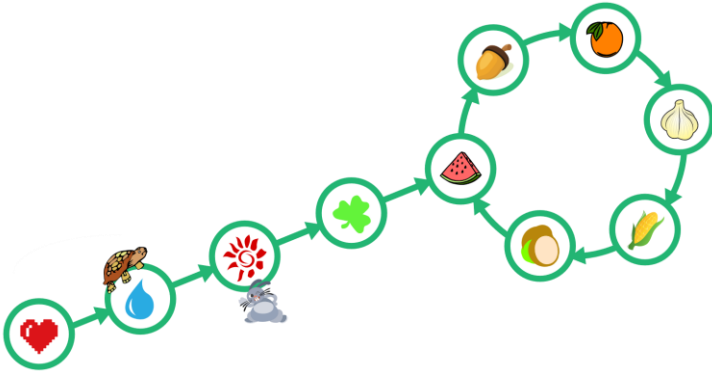
Kas pavaizduota laukelyje, kuriame vėžlys ir kiškis pirmą kartą po starto susitiks?



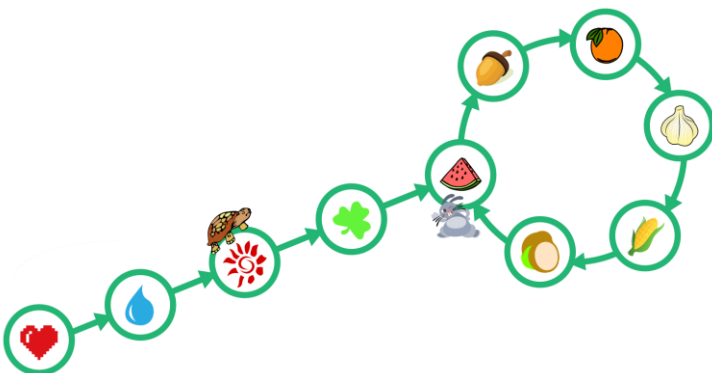
Paaiškinimas

Teisingas atsakymas yra 🍊. Toliau paveikslėliuose parodyta, kuriose vietose bus vėžlys ir kiškis po kiekvienos minutės:

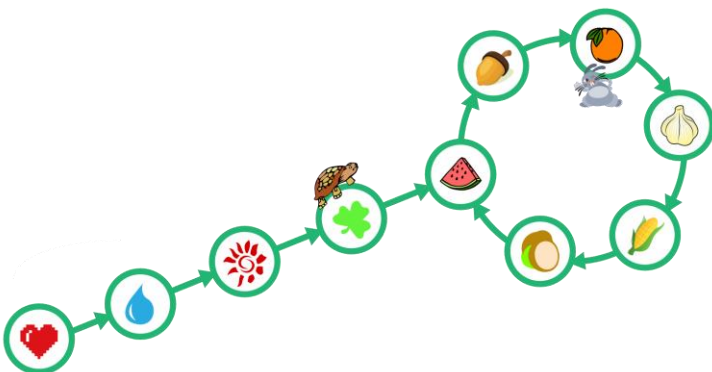
Po 1 minutės



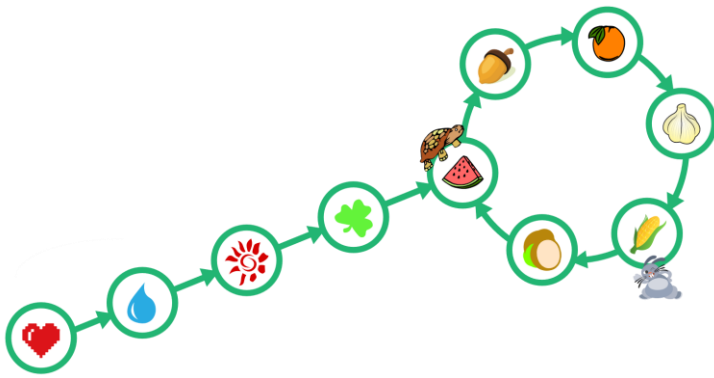
Po 2 minučių



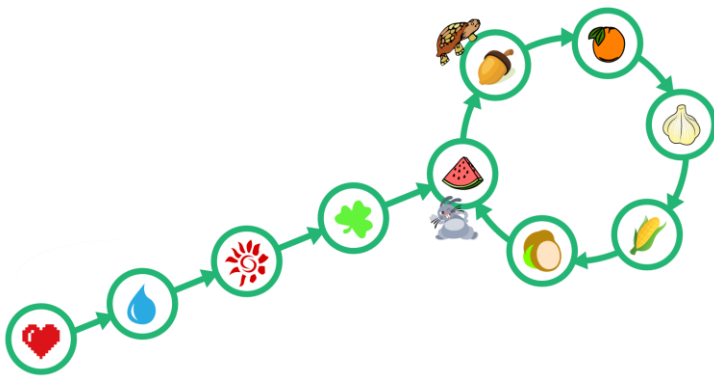
Po 3 minučių



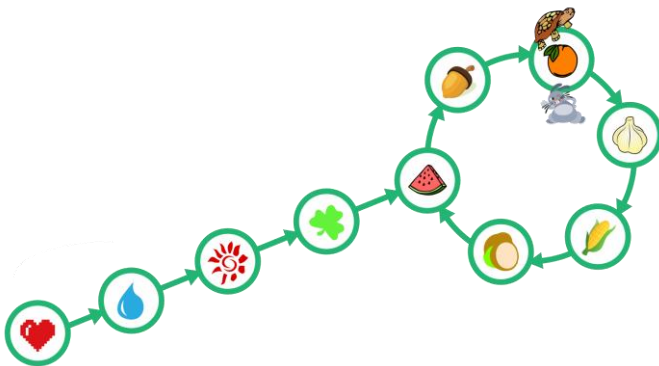
Po 4 minučių



Po 5 minučių



Po 6 minučių



Tai informatika!

Užduotyje naudojama tam tikra kryptinė duomenų struktūra – orientuotasis grafas su ciklu.

Susietasis sąrašas yra tokia duomenų struktūra, kurioje laikomi elementai, sujungti į seką. Šie elementai dar gali būti vadinami mazgais. Susietojo sąrašo pavyzdžiai galėtų būti veiksmų seka patiekalo recepte, lankytini objektai pakeliui iš vieno taško į kitą ar net algoritme algoritmo operacijos. Kad būtų išlaikyti sąryšiai, kiekvienas sąrašo elementas „žino“ kito elemento adresą. Kitaip tariant, kiekvienas mazgas turi rodyklę į kitą mazgą.

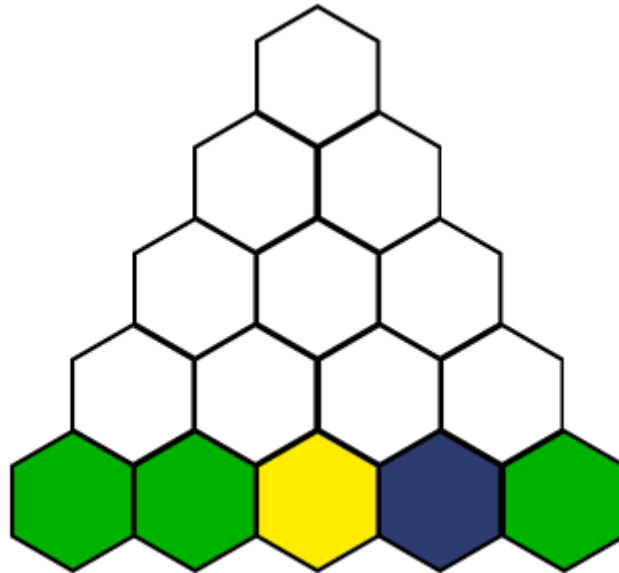
Paprastai susietasis sąrašas būna tiesinis – pradėję nuo pirmojo mazgo ir sekdami rodyklėmis, pasiektume paskutinį mazgą, nė vieno iš mazgų neaplankę dukart.

Jei nutiko taip, kad tą patį mazgą aplankėme dukart, vadinasi, patekome į ciklą. Kaip kompiuteriai aptinka ciklus? Vienas sumanus būdas, priskiriamas amerikiečių informatikui Robertui W. Floydui, yra vėžlio ir kiškio algoritmas (pavadintas pagal Ezopo pasakėčią). Kaip ir šioje užduotyje, algoritme naudojami du elementai, judantys susietuoju sąrašu skirtingais greičiais: vienas juda dvigubai greičiau už kitą. Jei jie susitinka, vadinasi, sąraše yra ciklas. Jei ne – susietasis sąrašas yra tiesinis.

Informatikoje labai svarbu aptikti ciklus. Pavyzdžiui, naudodami ciklą aptikimo metodus, galime patikrinti, ar parašyta programa nekartoja tam tikros veiksmų sekos be galo (begalinis ciklas), tokiu būdu neleisdama programai pasibaigti. Sudėtingesnis šių metodų panaudojimo atvejis – atsitiktinių skaičių generatorių kokybės analizė, ypač jautrių duomenų šifravimo ir apsaugos srityje. Tokiais atvejais dažniausiai naudojamas dar ir priešciklis, kuris nesikartoja, o tik po jo eina ciklas. Šioje mūsų užduotyje tiesia judėjimo linija pavaizduotas priešciklis, o ratu – ciklas. Kuo ciklai ilgesni, tuo duomenų apsaugos algoritmas stipresnis ir sunkiau įveikiamas programišiams.

11. Spalvota dėlionė

Kristupas turi trijų spalvų šešiakampes dėlionės detales. Trys detalės sudėtos į trikampio formos figūrą (žr. paveikslėlį). Jos turi būti tos pačios spalvos arba visos skirtingų spalvų.



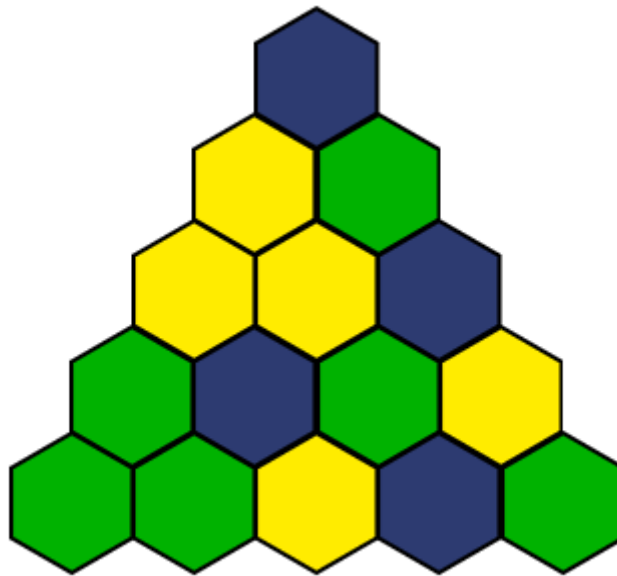
Kristupas nori sudėti didelę trikampę dėlionę, pavaizduotą paveiksle. Kokios spalvos turi būti viršutinė detalė?



Paaiškinimas

Kai žinome dviejų šalia esančių detalių spalvas, tada aišku, kokia turi būti virš jų esančios detalės spalva. Pavyzdžiui, panagrinėkime, kokios spalvos turėtų būti antroje eilėje esanti kairiausia dėlionės detalė. Kadangi dvi detalės, esančios po šia nežinoma detale, yra žalios, tad ir ši detalė taip pat turi būti žalia. Greta dešinėje esanti detalė turi būti mėlyna, nes dvi po ja esančios detalės yra žalios ir geltonos spalvos.

Šį metodą, eidami nuo apačios iki viršaus, galime taikyti visai dėlionei. Rezultatas turėtų būti toks:



Atkreipiame dėmesį, kad kiekvienoje dėlionės eilėje galima nustatyti bet kurioje vietoje esančių detalių spalvas.

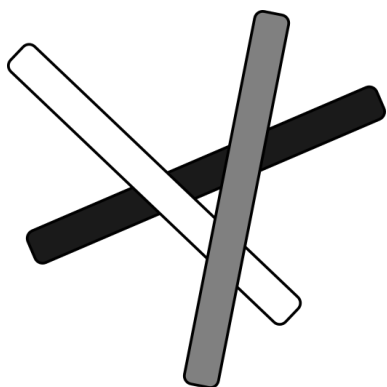
Tai informatika!

Ši užduotis yra gera iliustracija informacinių technologijų įgūdžio, vadinamo modelio atpažinimu, kuris specifiškai naudojamas rasti panašius arba bendrus elementus. Norint sudėti korteles eilės tvarka reikia lyginti kortelių poras ir surasti tas, kurios turi panašiausių modelius. Supratęs, kaip paveikslukas yra kuriamas (vis pridėdant baltų geometrinių formų), galima sudėti visus paveikslukus eilės tvarka.

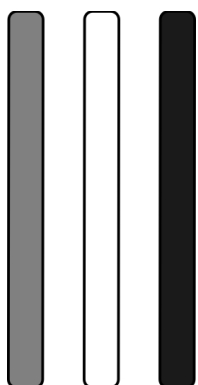
12. Paimk lazdeles

Ana žaidžia su lazdelėmis. Ji numeta keletą lazdelių ant stalo, o paskui išrikiuoja eilute po vieną. Ji vis ima viršutinę lazdelę, kurios neuždengia kita lazdelė.

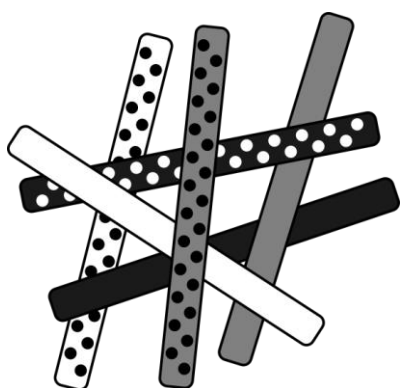
Pavyzdžiui, jei Ana numeta 3 lazdeles taip:



Ji turi jas surikiuoti tokia tvarka:



Ana numetė 6 tokias lazdeles:



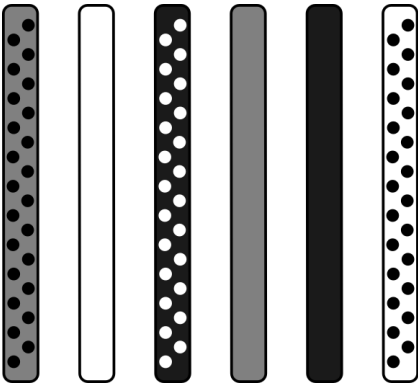
Kokia tvarka ji turėtų paimti lazdeles?

Paaiškinimas

Kadangi Ana turi pakelti po vieną lazdelę nejudindama kitų, ji ima viršutinę lazdelę.

Pirma virš kitų esanti lazdelė – taškuota pilka. Paėmus šią lazdelę, viršutinė bus balta lazdelė.

Paėmus baltąją – taškuota juoda lazdelė. Visa Anos paimtų lazdelių seka yra tokia: taškuota pilka – balta – taškuota juoda – pilka – juoda – taškuota balta.

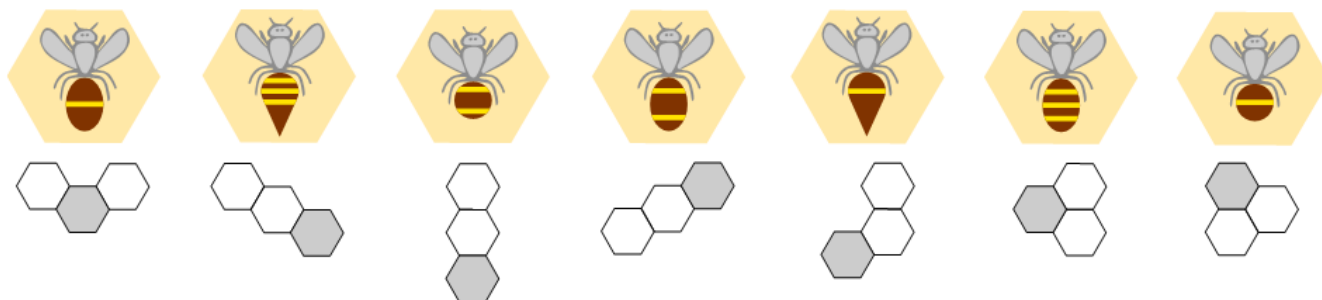
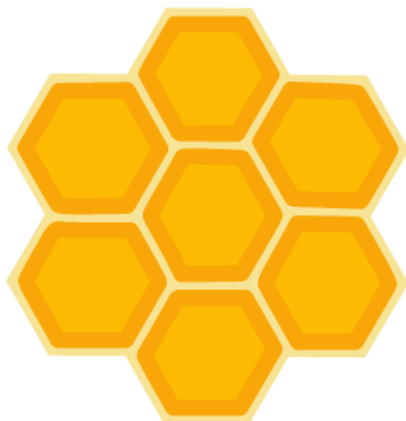


Tai informatika!

Mokiniai turi galvoti, kurios lazdelės jau paimtos ir kuri yra viršuje. Jie turi laikytis tam tikrų taisyklių: „paimti po vieną lazdelę“ ir „paimti vieną lazdelę, nejudindami kitų“ (algoritminis mąstymas). Be to, jie turi susieti lazdelių paėmimo tvarką su spalvų seka diagramoje (duomenų struktūros ir atvaizdai). Be to, lazdelės paėmimo veiksmą galima laikyti abstrakčios struktūros, pavyzdžiui, grafo, operacija, kurią galima atlikti tik tuo atveju, jei tenkinami tam tikri reikalavimai. Po galiojančių transformacijų sekos (lazdelių paėmimo iš krūvos veiksmo) rezultatas gali būti laikomas supaprastinta arba sumažinta pradinės struktūros versija (lazdelių krūva vis mažesnė ir mažesnė).

13. Bičių avilys

Bebrui reikia pagalbos – suginti bites į avilį.

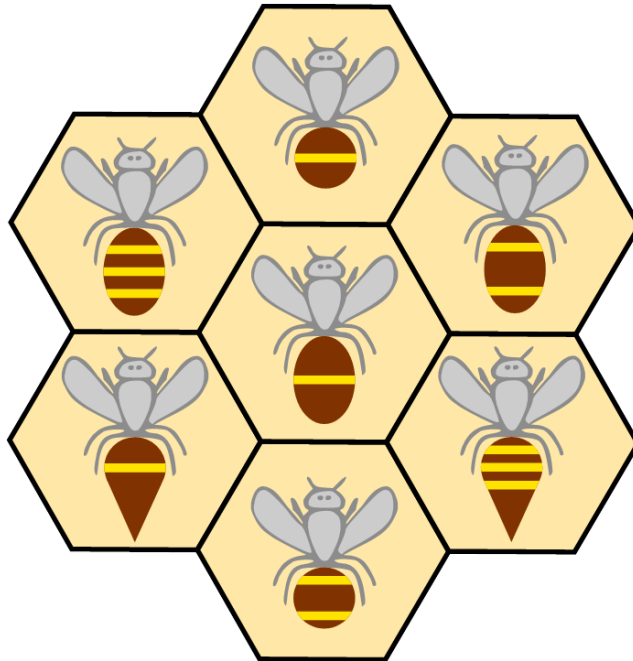


Po kiekviena bite yra taisyklė: parodyta, kuriame pilkai nuspalvintame langelyje turi atsirasti ta bitė.

Vilkite bites į avilį, vadovaudamiesi nustatytais taisyklėmis.

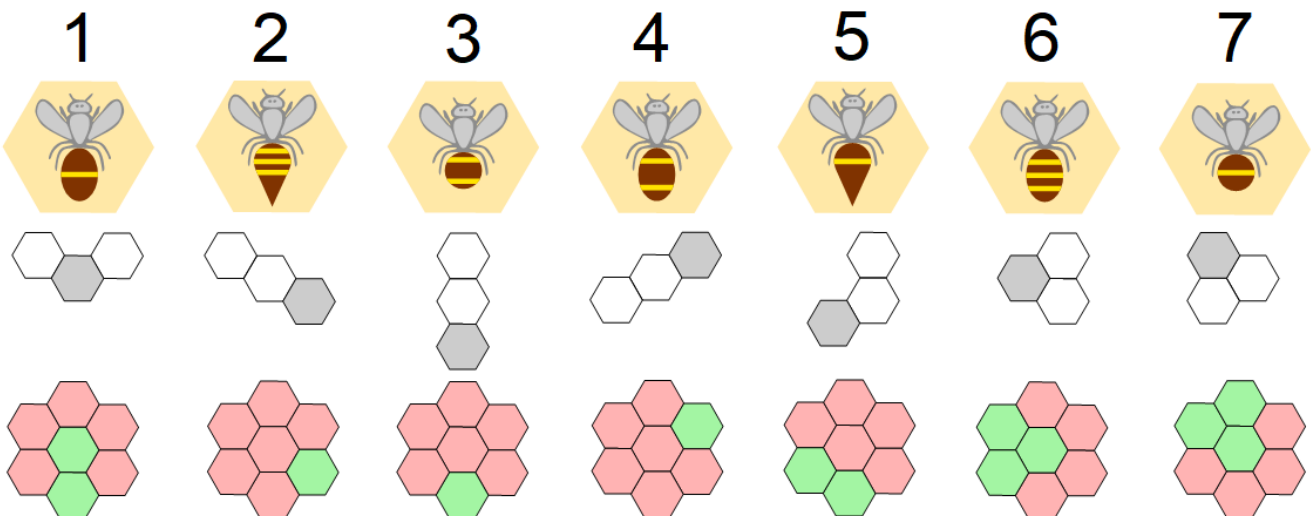
Paaiškinimas

Pateikiame sprendimą:



Užduotį galite išspręsti tiesiog bandydami. Tačiau tai gali užimti daug laiko. Norėdami rasti greitesnį būdą, atidžiau pažvelkite į bičių taisykles. Toliau pateiktame paveikslėlyje matote kiekvieną bitę ir jos taisyklę. Langeliai, į kuriuos bitė gali būti įdėta pagal savo taisyklę, nuspalvinti žaliai.

Matome, kad kai kurias bites galima įdėti tik į vieną avilio akutę, o kitas – į skirtingas akutes. Pavyzdžiui, 2-ą bitę galima įdėti tik į vieną langelį, nes visame avilyje yra tik vienas galimas variantas, kur pagal taisyklę ši bitė gali būti.



Spręsdami užduotį atlikite šiuos veiksmus: pirmiausia suvilkite bites, kurios gali būti tik viename langelyje, t. y., 2-ą, 3-ą ir 4-ą bites.

Tuomet 1-ai ir 5-ai bitėms lieka vienintelė galima vieta. Tokiu pat būdu iš eilės nuvilkite 6-ą bitę ir galiausiai 7-ą bitę.

Tai informatika!

Atlikdami šią užduotį turite septynias bites sudėti į septynias skirtingas vietas. Galimų kombinacijų skaičius yra labai didelis. Jei atsižvelgsite į taisykles, galimybių skaičius labai sumažės, tačiau vis tiek bus pakankamai didelis, jas visas išbandyti reikėtų nemažai darbo. Sprendžiant užduotį svarbiausia yra vadovautis tinkama tvarka, kuri dar labiau sumažina minėtą skaičių. Šiuo atveju pradėjome nuo labiausiai apribotų elementų, kad sumažintume nagrinėjamų atvejų skaičių.

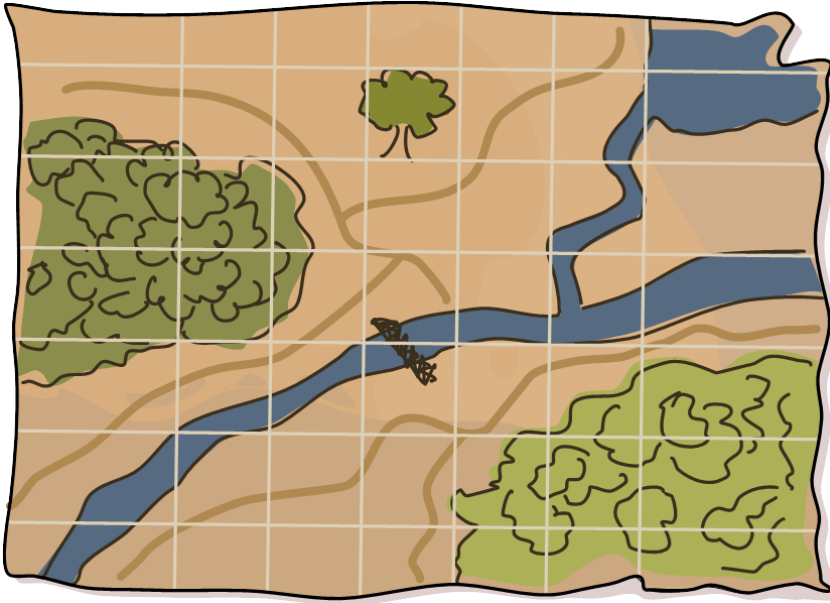
Tokie kaip šio uždavinio nagrinėjimo aspektai gali padėti pašalinti problemas, kurių kitaip būtų neįmanoma išspręsti.

Vyresniems mokiniams

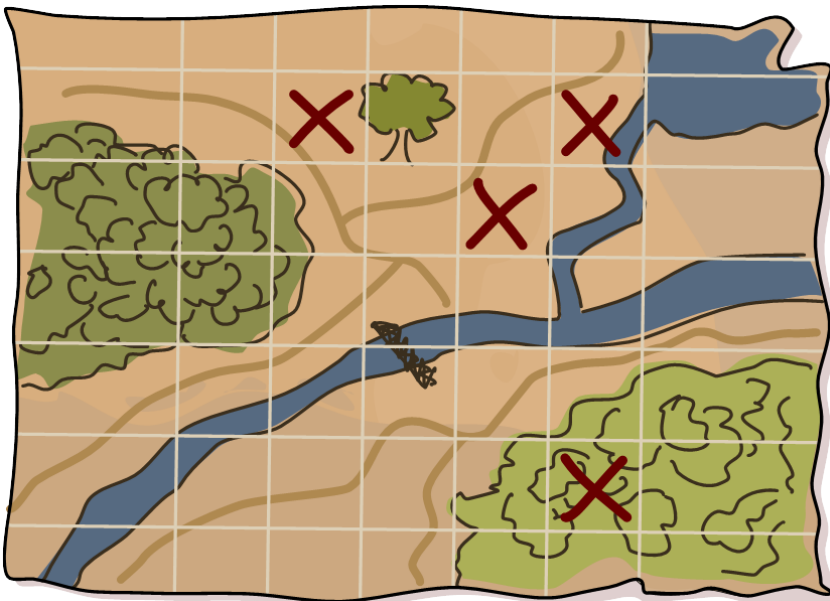
Problemos sprendinio paieška yra tarsi kelio per grafa, jungiančio pradinę padėtį su sprendiniu, radimas. Grafo briaunos reiškia sprendimus, o mazgai – būsenas. Kai kuriomis problemų sprendimo strategijomis stengiamasi rasti priemonių, kurios sumažintų kelio sudėtingumą, kai kuriomis strategijomis dėmesys sutelkiamas į tai, kaip kelias įveikiamas, pavyzdžiui, kartais lengva rasti žingsnius, vedančius į sprendimą, einant atgal nuo sprendimo iki pradinės padėties.

14. Maisto slėptuvės

Bebras surado dvi geras vietas maistui paslėpti ir norėtų pasižymėti jas žemėlapyje:



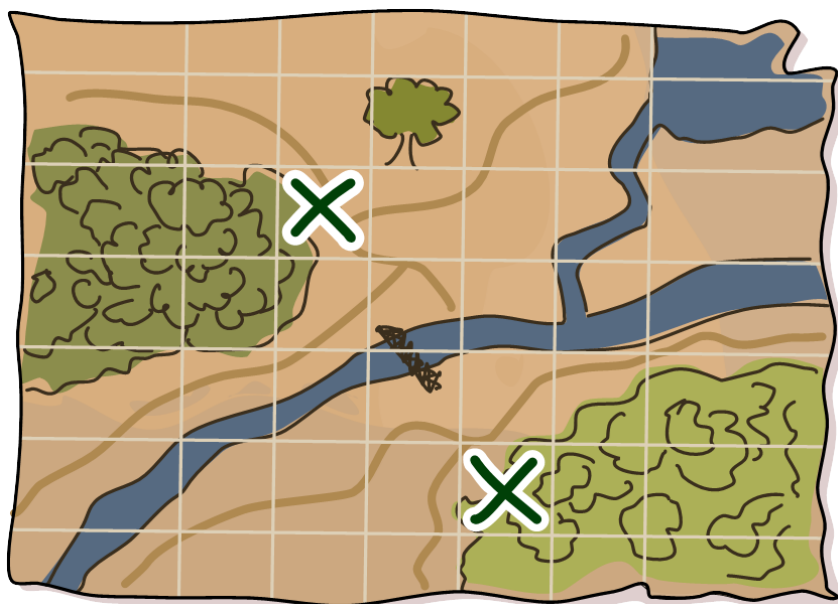
Maisto slėptuvės galėtų būti žymimos X, tačiau kyla pavojus, kad kas nors kitas, pamatęs žemėlapij, jas suras. Bebras, siekdamas suklaidinti, kai kuriuose žemėlapio langeliuose pažymi X taip, kad bendras X skaičius kiekvienoje eilutėje ir kiekviename stulpelyje būtų lyginis (pastaba: nulis laikomas lyginiu skaičiumi). Tada bebras ištrina žemėlapyje du X, kurie žymi jo tikrąsias slėptuves. Žemėlapis dabar atrodo taip:



Kur yra dvi bebro maisto slėptuvės?

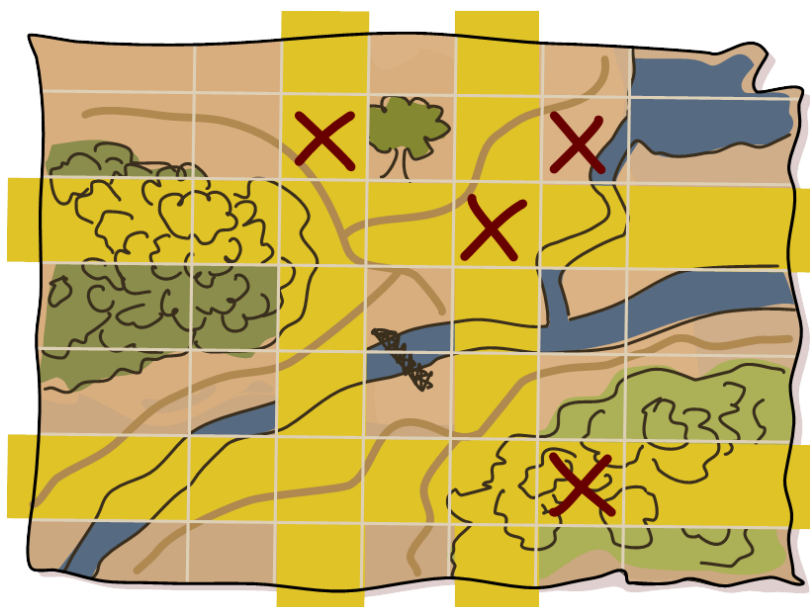
Paaiškinimas

Štai dviejų maisto slėptuvių vietos:



Norėdami rasti slėptuves turime patyrinėti sąlygoje pateiktą žemėlapi, kuriame matome, kad yra dvi eilutės ir du stulpeliai, kuriuose X sumos yra nelyginės: 2-oji ir 5-oji eilutės (iš apačios) bei 3-asis ir 5-asis stulpeliai (iš kairės).

Šias eilutes ir stulpelius pažymėkime geltonai:



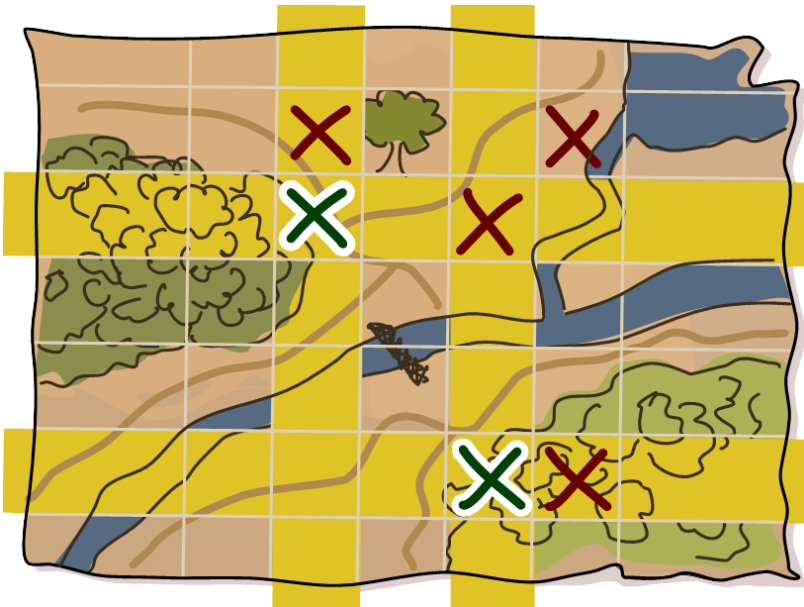
Ieškomi X, rodantys tikrąsias maisto slėptuvių vietas, turi būti geltonosiose zonose. Turime rasti būdą, kaip įdėti juos atgal į šias eilutes ir stulpelius, kad rastume maisto slėptuves.

Geltonų eilučių ir stulpelių sankirtose yra 4 langeliai. Kadangi viršuje jau yra vienas X, kuris negali būti tikroji slėptuvė, nes žinome, kad bebras tikrąją slėptuvę ištrynė. Taigi, kad gautume

lyginę X sumą šioje eilutėje, turime žymėti kitą viršutinį susikirtimą. Taip sužinome tikrąją maisto slėptuvę.

Antrąją slėptuvę galima nustatyti taip: radę pirmąją, antrosios negalime dėti apatiniame kairiajame sankirtos langelyje, nes tame stulpelyje būtų trys X. Liko vienintelis pasirinkimas – apatinis dešinysis langelis.

Čia parodytas žemėlapis prieš bebrui ištrinant X, žyminčius tikrąsias slėptuves. Kiekviename stulpelyje ir eilutėje yra lyginės X sumos.



Tai informatika!

Bebras naudoja dažnai programavime sutinkamą būdą (lyginio bito naudojimas), kuris taikomas klaidoms rasti ir joms ištaisyti. Šio būdo idėja: visada, kai reikia perduoti duomenis, dvejetainiame kode (vienetai ir nuliai) pridedame papildomus bitus, kurie padeda nustatyti, ar siunčiant duomenis neatsiranda klaidos, pvz., ar nepakeistas bitas (iš 1 į 0 ar atvirkščiai).

Pavyzdžiui, jei naudotume šį klaidos nustatymo būdą, tai pridėtume papildomą vieneta, kad jų suma būtų lyginė. Prie 0110101 pridėtume nulį, kad gautume 01101010. Jei antrasis bitas pasikeistų ir gautume kodą 00101010, tai jis neatitiktų lyginumo reikalavimų (trys bitai yra vienetai). Tiesa, jei du bitai būtų pakeisti klaidingai, tai šis būdas netiktų.

15. Sumuštinio receptas

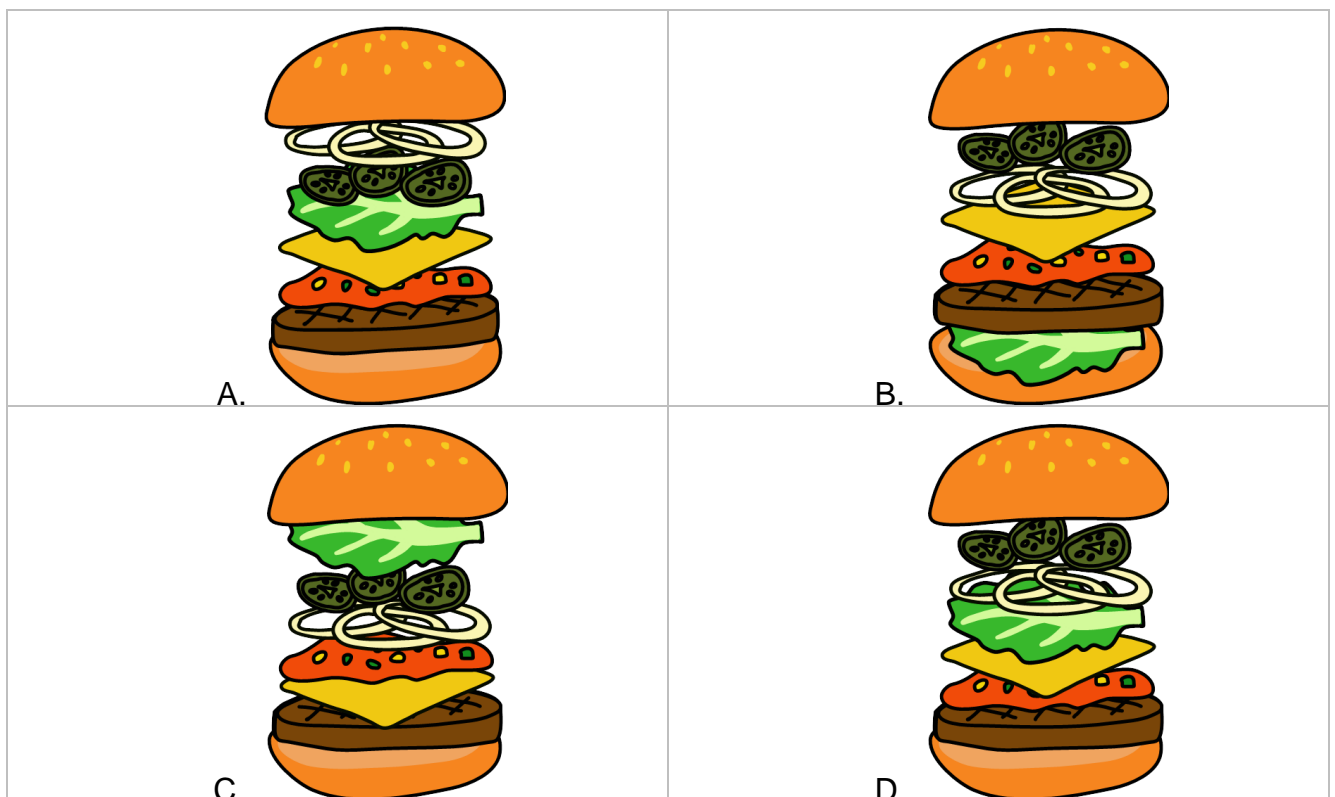
Sumuštinio sudedamosios dalys:

Bandelės	Mėsa	Padažas	Marinuoti agurkėliai	Salotos	Svogūnai	Sūris
						

Bebriukė Lina gamina sumuštinius pagal taisykles.

1. Padažas turi būti užpiltas iškart ant mėsos.
2. Mėsa ir sūris turi būti po marinuotais agurkėliais, salotomis ir svogūnais.
3. Svogūnai neturi liesti bandelių.
4. Visi produktai turi būti tarp bandelių.

Kuris sumuštinis pagamintas pagal taisykles?



Paaiškinimas

Teisingas atsakymas yra D.

(A) Šis sumuštinis atitinka 1 ir 2 taisykles, bet svogūnas liečia viršutinę bandelę, todėl neatitinka 3 taisyklės.

(B) Šis sumuštinis atitinka 1 taisyklę, bet salotos yra po mėsa ir sūriu, todėl neatitinka 2 taisyklės.

(C) Sūris yra tarp mėsos ir padažo, todėl neatitinka 1 taisyklės.

(D) Atitinka visas taisykles!

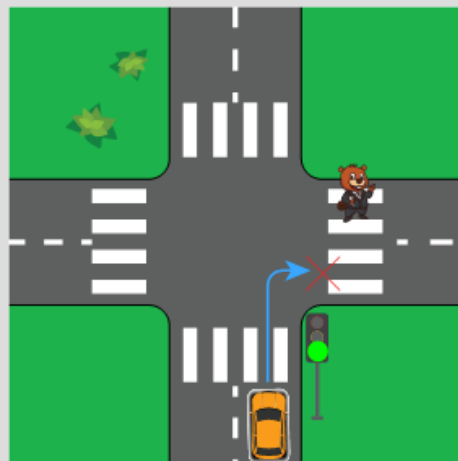
(Visi sumuštiniai atitinka 4 taisyklę.)

Tai informatika!

Kompiuterių moksle išsiaiškinimas, ar sprendinys atitinka visas duotas taisykles, vadinamas apribojimų tikrinimu.

Panagrinėkime pavyzdį. Pateiktame paveikslėlyje savavaldis automobilis turi nustatyti, ar jis gali sukti į dešinę. Automobilis juda laikydamasis dviejų taisyklių:

- Turi degti žalias šviesoforo signalas.
- Neturi būti pėsčiųjų, einančių per gatvę.



Vienas dalykas yra patikrinti, ar tam tikras sprendinys atitinka apribojimus, kitas – rasti sprendinį, atitinkantį apribojimus. (Tai vadinama apribojimų tenkinimo problema.) Dažniausiai net ir kompiuteriui rasti sprendimą yra daug sunkiau, nei patikrinti apribojimus.

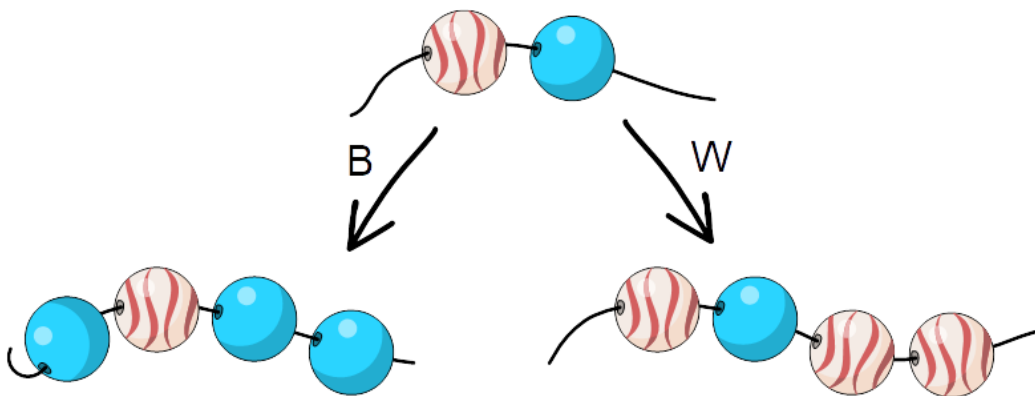
16. Jūreivio vėrinys

Pateikta instrukcija, kaip padaryti jūreivio vėrinį iš dviejų rūšių karoliukų: šviesių su rausvomis bangelėmis ir vienspalvių mėlsvų. Jūreivio vėrinys pradedamas verti šviesiu su bangelėmis ir

mėlsvu karoliuku tokia tvarka:

Tada vėrinys gali būti ilginamas dviem būdais:

- užveriant po vieną mėlsvą karoliuką iš abiejų virvelės galų (B veiksmas);
- užveriant du šviesius su bangelėmis karoliukus iš dešiniojo virvelės galo (W veiksmas).

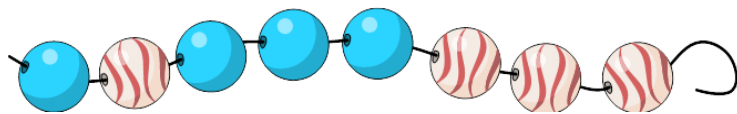


Šiuos veiksmus galima kartoti veriant vis ilgesnį ir ilgesnį jūreivio vėrinį.

Kuris vėrinys nėra jūreivio vėrinys?

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

Paaiškinimas



Teisingas atsakymas:

Šį uždavinį galima išspręsti daugeliu būdų. Štai trys galimybės.

Jūs galite gauti kiekvieną jūreivio vėrinio variantą pirmiausiai identifikavę du pradinius karoliukus ir tada atlikę veiksmų B ir W seka. Vėrinys A gali būti gautas pradėjus nuo antro ir trečio karoliuko ir tada atlikus veiksmus B-W-W. Vėrinys B gali būti gautas pradėjus nuo trečio ir ketvirto karoliuko ir tada atlikus veiksmus B-B-W. Vėrinys C gali būti gautas pradėjus nuo antro ir trečio karoliuko ir tada atlikus veiksmus W-B-W. Tačiau pažvelgę į D vėrinį matome, kad pradiniai turi būti antras ir trečias karoliukai ir po to atliktas veiksmas B. Deja, nėra tokių veiksmų, kuriais galima būtų gauti likusią jūreivio vėrinio dalį.

Šis metodas netiktų, jei vėrinys būtų labai ilgas ir pradiniai karoliukai galėtų būti parinkti daugeliu būdų. Tokiu atveju verčiau reikėtų pabandyti dekonstravimo metodą, kai karoliukai nuimami nuo virvelės galų kartojant veiksmus B ir W, kol liks tik du pradiniai karoliukai.

Trečioji strategija – atkreipti dėmesį, koks yra suvertų vienos spalvos karoliukų skaičius. Remiantis instrukcija, vėrinį visada sudarys nelyginis skaičius vienspalvių melsvų karoliukų ir nelyginis skaičius šviesiu su bangelėmis karoliukų. (Ar aišku, kodėl?) D vėrinį sudaro po lyginį skaičių vieno ir kito tipo karoliukų, todėl tai negali būti jūreivio vėrinys.

Tai informatika!

Karoliukų galima pridėti tik iš virvelės galų. Negalima įterpti karoliuko į vėrinio vidurį.

Lygiai taip pat, norėdami išardyti jūreivio vėrinį, galite nuverti karoliukus tik nuo virvelės galų. Negalima pašalinti karoliuko iš vėrinio vidurio.

Tokio tipo struktūra, kurios dalis galima lengvai pridėti ir pašalinti iš kraštų, bet ne iš vidurio, yra naudojama informatikoje ir yra vadinama dvipuse eile (angl. *double-ended queue*, *deque*). Dvipusės eilės gali būti naudojamos naršyklės istorijai saugoti, spausdinimo darbų planui nustatyti, matematinio reiškinių teisingumui patikrinti. Matematinį skliaustų atitikimą gali būti patikrintas panašiai, kaip buvo tikrinamas jūreivio vėrinio teisingumas.

17. Širdžių grafika

Tėja pradėjo nuo vieno skritulio ir vieno kvadrato:



Tada ji sukūrė šią širdį:



Tėja gali atlikti tik šiuos veiksmus su figūromis:

Pasukti: pasukti figūrą bet kuria kryptimi

Perkelti: perkelti figūrą į bet kurią vietą

Kopijuoti: sukurti figūros kopiją toje pačioje vietoje

Kurią veiksmų seką atliko Tėja?

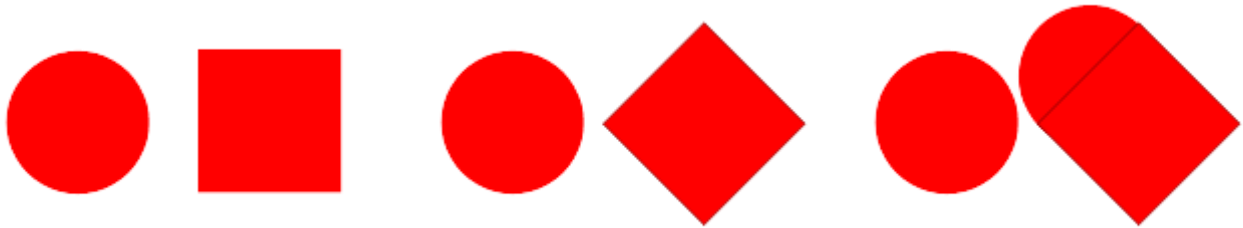
A: Kopijuoti skritulį. Pasukti kvadrata. Perkelti skritulį. Perkelti skritulį.

B: Kopijuoti kvadrata. Pasukti kvadrata. Perkelti kvadrata. Perkelti skritulį.

C: Kopijuoti skritulį. Pasukti skritulį. Perkelti skritulį. Perkelti kvadrata.

D: Perkelti skritulį. Perkelti skritulį. Kopijuoti skritulį. Perkelti kvadrata.

Paaiškinimas



Teisingas atsakymas yra A.

Kvadratą reikia pasukti taip, kad vienas jo kampas būtų nukreiptas žemyn. Taip bus tik sekose A ir B. Taigi, teisingas sprendimas yra A arba B. Toliau matome, kad širdžiai sukurti reikia dviejų skritulių. B seka negali būti teisinga, nes joje nesukuriamas antras skritulys. Taigi, vienintelė teisinga seka yra A.

Pateiktuose paveikslėliuose pavaizduotos figūros po kiekvieno A sekos veiksmo. Po paskutiniojo veiksmo sukuriama širdis. Tai patvirtina, kad Tėja naudojo seką A.

Tai informatika!

Konstruodami vaizdus grafikos rengykle atliekame įvairius veiksmus. Šios užduoties širdies paveiksle kopijuojame skritulį, sukame kvadratą ir perkeliame skritulius. Jei norėtume, kad šį darbą atliktų kompiuteris, turėtume pridėti tam tikrą informaciją. Turėtume nurodyti, kur perkelti kiekvieną skritulį ir kiek pasukti kvadratą.

Atsakymų variantuose pateiktos veiksmų sekos yra neišsamūs nurodymai, ką reikia daryti.

Informatikos mokslininkai sakytų, kad jos yra komandų abstrakcijos. Jos parodo idėją, ką daryti, bet nepaaiškina to pakankamai išsamiai, kad kompiuteris galėtų įvykdyti. Tai naudinga, nes ignoruodami detales galime sutelkti dėmesį į tai, kas svarbu.

Informatikos mokslininkai, kurdami kompiuterio programą, dažnai pradeda nuo abstrakcijų, kurios sudaro vaizdą, kaip programa veiks, o trūkstamas detales prideda vėliau.

18. Miško vakarėlis













Bebriukė Asta, lapė Lina ir lokys Rokas dalyvauja vakarėlyje. Lentelėje parodyta, kokį maistą kiekvienas gali ēsti.

	 Lapai	 Žuvis	 Grybai	 Uogos
 Bebriukė Asta	Taip	Ne	Ne	Taip
 Lapė Lina	Ne	Taip	Ne	Taip
 Lokys Rokas	Ne	Taip	Taip	Taip






















Vakarėlyje pateikti devyni maisto produktai. Kiekvienas gyvūnas turi gauti po tris maisto produktus.



Padalink trims draugams po tris maisto produktus. Instrukcija: nutempk tinkamą maisto produktą prie kiekvieno gyvūno.

Paaiškinimas

Spręsti šią užduotį galima pradėti nuo bebrukės Astos, nes ji ėda lapus. Tai reiškia, kad abu lapai turėtų atitekti jai. Vienintelis kitas maistas, kurį ji ėda, yra uogos, todėl jos trečiasis maistas turi būti uogos. Toliau lokys Rokas yra vienintelis, kuris ėda grybus, todėl visi trys grybai turėtų atitekti jam. Lapė Lina gaus tai, kas liko: dvi kekes uogų ir žuvį. Šį uždavinį galima spręsti ir kitaip, pavyzdžiui, galima pradėti nuo lapės ir pastebėti, kad ji ėda tik žuvį ir uogas, todėl ji turi gauti bent dvi kekes uogų, o trečias produktas bus arba žuvis arba kekė uogų. Bebrukė vienintelė ėda lapus. Tai reiškia, kad abu lapai turi atitekti jai. Vienintelis kitas maistas, kurį ji ėda, yra uogos, todėl jos trečioji porcija turi būti uogų kekė. Kadangi uogų neliko, dabar žinome, kad trečiasis lapės produktas turi būti žuvis. Lokys gaus tai, kas liko – tris grybus.

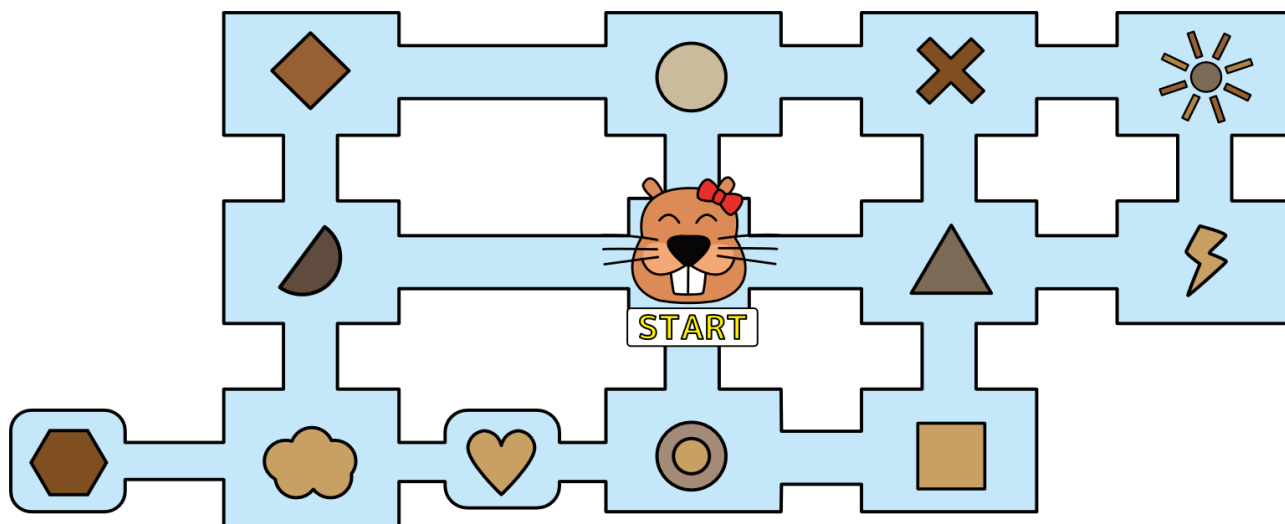
Tai informatika!

Šiai užduočiai (apribojimų tenkinimo uždaviniui) spręsti reikia informatinio mąstymo logikos ir užduoties skaidymo įgūdžių. Logika. Šioje užduotyje pateikiamas loginių savybių lentelės pavyzdys ir ja naudojama sprendžiant. Kiekvienam gyvūnui yra aiškiai nurodyta („taip“ arba „ne“), ar jis ėda tos rūšies maistą. Be to, kiekviena maisto porcija gali būti paskirta tik vienam gyvūnui vadovaujantis lentele. Pirmiausia paskirstydami nesudėtingas maisto porcijas, sukuriame naujas žinias, kurios leidžia toliau spręsti uždavinį. Skaidymas į dalis. Ši užduotis yra uždavinio skaidymo į dalis pavyzdys. Kiekviena maisto porcija paskirstoma atskirai, nors svarbu, kokia tvarka paskirstome porcijas, kad rastume tokį paskirstymą, kuris tenkintų uždavinio apribojimus.

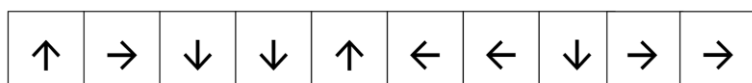
20. Kur bebras?

Bebrė yra kambaryje, pažymėtame žodžiu START, ir nori patekti į Karlsono kambarį.

Bebrė naudojami kambarių schema, kurioje kiekvienas kambarys pažymėtas paveiksliuku:



Bebrei judėjimo maršrutas yra nurodytas tokia rodyklių seka:



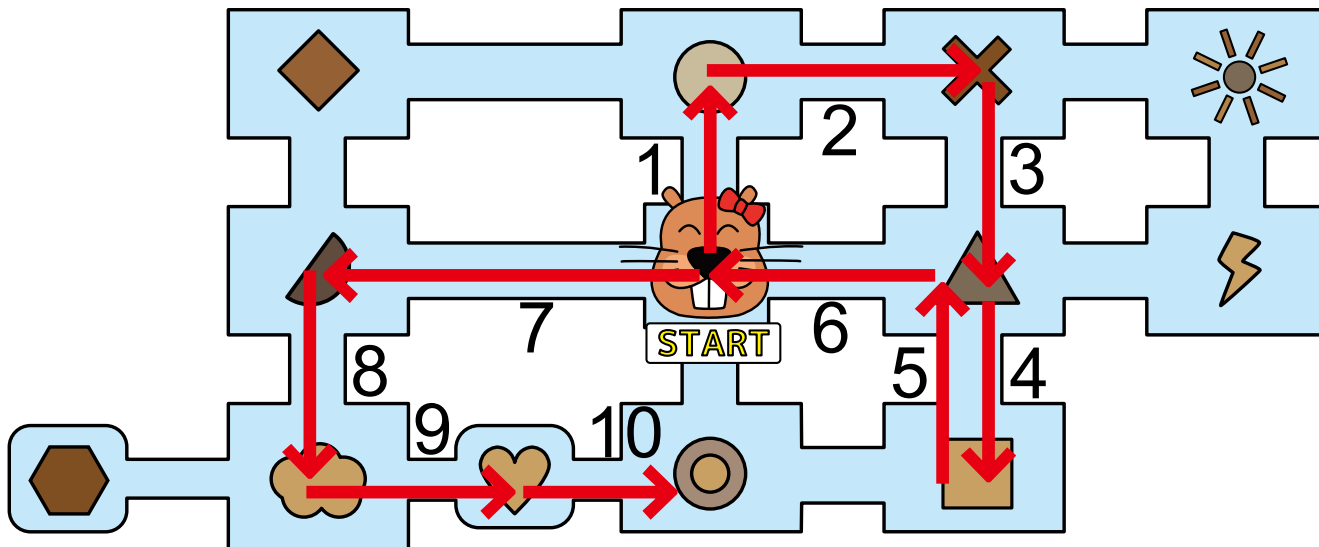
Kiekviena rodyklė parodo, kuria kryptimi reikia judėti einant iš kambario į kambarį.

Kuriuo paveiksliuku yra pažymėtas Karlsono kambarys?



Paaiškinimas

Bebės kelias schemoje parodytas raudonomis rodyklėmis.



Tai informatika!

Kiekviena rodyklė šioje „Bebro“ užduotyje nurodo Bebrei, kur jai toliau eiti. Visa rodyklių seka apibrėžia kelią. Ši užduotis iliustruoja vieną pagrindinių programavimo sąvokų – seką, rodyklės vaizduoja komandas ir jų vykdymo tvarką (iš kairės į dešinę).

Žymeklio (šiuo atveju – uždavinio herojės Bebrės) judėjimo nurodytu maršrutu stebėjimas yra labai svarbus dalykas nagrinėjant kompiuterinę programą, kai norima suprasti programos veiksmų eigą. Kokius paveikslukus sutinkame maršrute ir kada sustojama?

21. Antivirusinė programa

Ajus sukūrė naują antivirusinę programą, kuri gali aptikti pavojingą vykdomąjį failą. Programa tikrina norimą atverti failą – tam naudojamas sąrašas, sudarytas iš keturių teiginių:

- įrašyti į sistemos aplanką,
- skaityti iš sistemos aplanko,
- įtraukti į automatinio paleidimo programą,
- siųsti el. paštu.

Jei antivirusinė programa aptinka 2 teiginius (turi būti pavartoti visi atitinkamo teiginio žodžiai), failas laikomas pavojingu ir patenka į karantiną.

Kuris iš šių failų pateks į karantiną?

A	C
skaityti failą iš dokumentų aplanko; atverti failą; įrašyti failą į dokumentų aplanką; siųsti failą el. paštu;	skaityti failą iš sistemos aplanko; atverti failą; įrašyti failą į dokumentų aplanką; siųsti failą el. paštu;
B	D
skaityti failą iš dokumentų aplanko; atverti failą įtraukti failą į automatinio paleidimo programą	skaityti failą iš automatinio paleidimo aplanko; atverti failą; įrašyti failą į dokumentų aplanką; siųsti failą el. paštu;

Paaiškinimas

Teisingas atsakymas: C.

A, B, ir D variantuose yra tik viena iš nurodytų frazių.

A	C
skaityti failą iš dokumentų aplanko; atverti failą; įrašyti failą į dokumentų aplanką; siųsti failą el. paštu;	skaityti failą iš sistemos aplanko; atverti failą; įrašyti failą į dokumentų aplanką; siųsti failą el. paštu;
B	D
skaityti failą iš dokumentų aplanko; atverti failą įtraukti failą į automatinio paleidimo programą	skaityti failą iš automatinio paleidimo aplanko; atverti failą; įrašyti failą į dokumentų aplanką; siųsti failą el. paštu;

Tai informatika!

Saugumo programinė įranga kuriama norint aptikti ir pašalinti pavojingus failus (ypač vykdomuosius) ar programas, kurios gali sutrikdyti kompiuterio darbą. Kartais pavojingą failą, pavyzdžiui, virusą, galima nustatyti vien iš failo pavadinimo. Antivirusinė sistema visada skenuoja kodą ir, naudodama informaciją iš savo duomenų bazės, stengiasi aptikti įtartinas programinio kodo dalis.

Kartais saugumo programinė įranga naudoja apsaugotą aplinką (angl. *sandbox*), kurioje ji vykdo programinį kodą ir tikrina, ar rezultatas skiriasi nuo tikėtino.

Naujausiose antivirusinėse sistemose naudojamas dirbtinis intelektas, šios sistemos „mokosi“, plečia įtartinų frazių duomenų bazę, kad lengviau ir greičiau aptiktų virusą.

22. Siuvimo mašina

Lina namuose turi programuojamą siuvimo mašiną, kuri gali siuvinėti: arba siūlėmis. Norint sukurti tokią sudėtingesnę siūlę , reikia šių abiejų dygsmių , . Tarp jų audinys turi būti pastumtas atgal vienu dygsniu.

Lina gali programuoti siuvinėjimo mašiną šiais trimis mygtukais:

	Atitinka siūlę.
	Atitinka .
	Audinys vienu dygsniu pastumiamas atgal.

Siuvimo mašinai pateikus šią programą:

, gaunama tokia siūlė:

Kurią iš šių programų Lina įvedė į siuvinėjimo mašiną, kad gautų tokią siūlę?

- A)
- B)
- C)
- D)

Paaiškinimas

Teisingas atsakymas: C.

Norėdami rasti teisingą atsakymą, pirmiausia turime atpažinti pasikartojantį šabloną

XX*XX*. Tai reiškia, kad pirmoji ir antroji komanda (mygtukas) turi būti: **X**. Dėl to galima atmesti D atsakymą. Tuomet siuvinėjama žvaigždutė, tam panaudojus **X** ir **+** komandas. Eilės tvarka nesvarbi, tačiau audinys turi būti pastumtas vienu dygsniu atgal: **←**. Tam reikia šių komandų: **X←+** arba **+←X**. Dėl to B ir D atsakymai netinka. Teisingoje programoje C yra šios komandos: **XXX←+XX+←X**.

Programa A **XXX←+X+←XXX** sukuria tokią siūlę: **XX*XX***

Programa B **XXX←+XX** išsiuvinėja tokį raštą: **XX*XX**.

Programos D pradžioje jau išsiuvinėtos dvi žvaigždutės:

+←X+←XX+←XX.

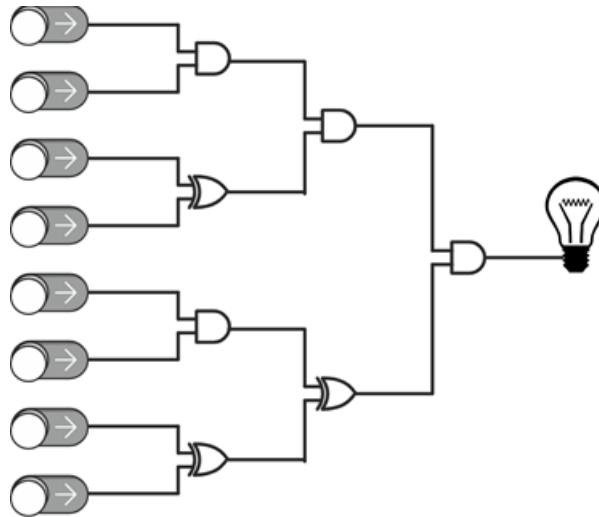
Tai informatika!


Atliekant šią užduotį pasikartojantis šablonas sukuriamas naudojant mygtukų (instrukcijų) seką. Kompiuterių moksle dideli ir sudėtingi uždaviniai taip pat dažnai skaidomi į mažesnius, kuriuos lengviau suprasti, spręsti ir programuoti. Svarbus šio proceso įgūdis – atpažinti pasikartojančių šablonų sekas. Tai galima padaryti, pavyzdžiui, naudojant ciklus.

Siuvinėjimo mašinos sukuriama programa atitinka programavimo kalba parašytų instrukcijų seką. Iš esmės programuojamoji siuvinėjimo mašina yra tik robotas (arba kompiuteris), kuris vykdo instrukcijas. Kaip siuvinėjimo mašina siuvinėja tikslus dygsnius, taip kompiuteris vykdo tikslias programos komandas. Tikslus instrukcijų vykdymas yra svarbi informatikos sąvoka. Ne mažiau svarbi ir instrukcijų vykdymo tvarka. Jei pakeičiame tvarką, paprastai pasikeičia ir programos išvestis. Mūsų atveju tai reiškia, kad kitokia instrukcijų seka lems kitokią dygsnių seką, taigi ir kitokį raštą (siūlę).

23. Įjunk šviesą!

Žaidime „Įjunk šviesą“ yra 8 jungikliai, kuriuos galima valdyti. Iš šių jungiklių išvesti laidai eina pro tam tikrus komponentus ir galiausiai pasiekia lemputę.



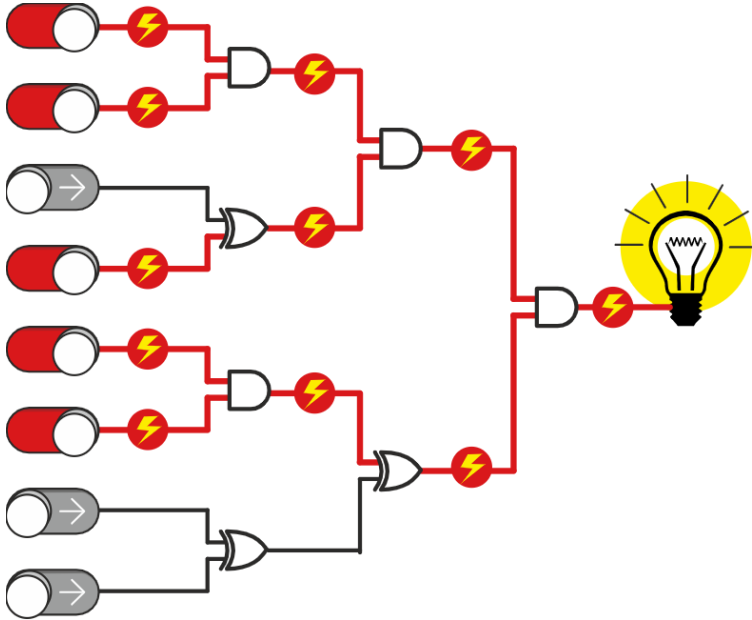
Komponentas  įsijungia tik tada, kai įjungti ABU įeinantys laidai.

Komponentas  įsijungia, kai įjungtas tik VIENAS iš įeinančių laidų.

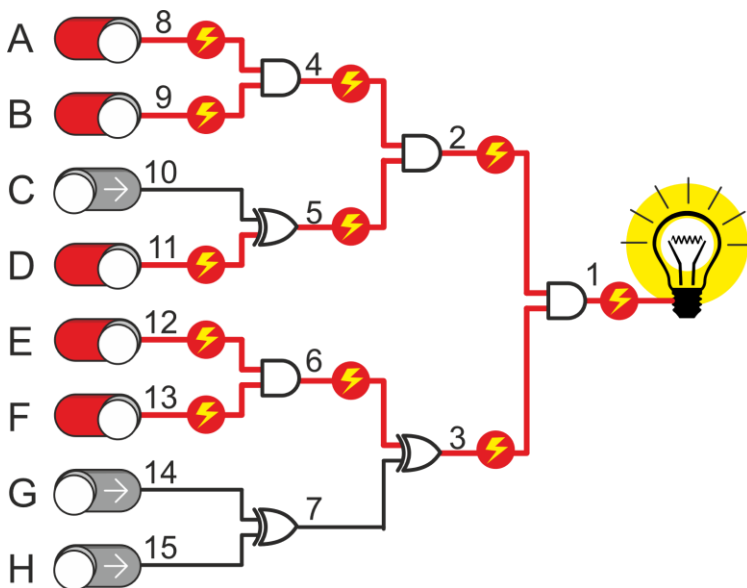
Kurie jungikliai turi būti įjungti, kad lemputė šviestų?

Paaiškinimas

Yra 16 galimų 8 mygtukų kombinacijų, kurios galiausiai įjungia lemputę. Vienas iš sprendimų pavaizduotas pateiktame paveikslėlyje: laidai, jei jais teka srovė, paryškinti raudona spalva.



Šią užduotį galima spręsti ir atvirkštine tvarka.



Dešinėje pusėje esanti lemputė yra prijungta prie laido, einančio iš dešiniojo komponento AND (1). Kad šis laidas būtų įjungtas, du laidai (2 ir 3), vedantys į komponentą, taip pat turi būti įjungti. Šie laidai yra prijungti prie AND komponento ir jo OR sudedamosios dalies.

Žinome, kad komponentas AND turi būti įjungtas, todėl abu į jį vedantys laidai (4, 5) turi būti įjungti.

Žinome, kad komponentas \mathcal{D} turi būti įjungtas, todėl vienas iš prie jo prijungtų laidų (6, 7) turi būti įjungtas, o kitas – išjungtas.

Atskirkime viršutinės ir apatinės pusės laidus:

Viršutinė pusė:

Abu laidai (4, 5) turi būti įjungti. Todėl laidai (8, 9) turi būti įjungti, t. y., turi būti įjungti 2 viršutiniai jungikliai (A ir B), ir vienas iš laidų (10, 11) turi būti įjungtas, t. y., turi būti įjungti jungikliai C ir D.

Apatinė pusė:

Jei komponentas \mathcal{D} yra įjungtas, vienas iš laidų (6, 7), vedančių į komponentą \mathcal{D} , turi būti įjungtas, o kitas – išjungtas. Todėl arba komponentas \mathcal{D} , kuris veda į kitą komponentą \mathcal{D} , turi būti įjungtas, o komponentas \mathcal{D} išjungtas, arba atvirkščiai. Tai galima padaryti keliais būdais, pirmiau pateiktas sprendimas rodo vieną iš galimų būdų. Šiuo atveju komponentas \mathcal{D} yra įjungtas, todėl jungikliai E ir F turi būti įjungti, o tai reiškia, kad komponentas \mathcal{D} turi būti išjungtas, o du apatiniai jungikliai (G ir H) yra arba abu įjungti, arba abu išjungti (šiuo atveju abu išjungti).

Visas galimybes galite pamatyti šioje lentelėje:

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

10								
11								
12								
13								
14								
15								
16								

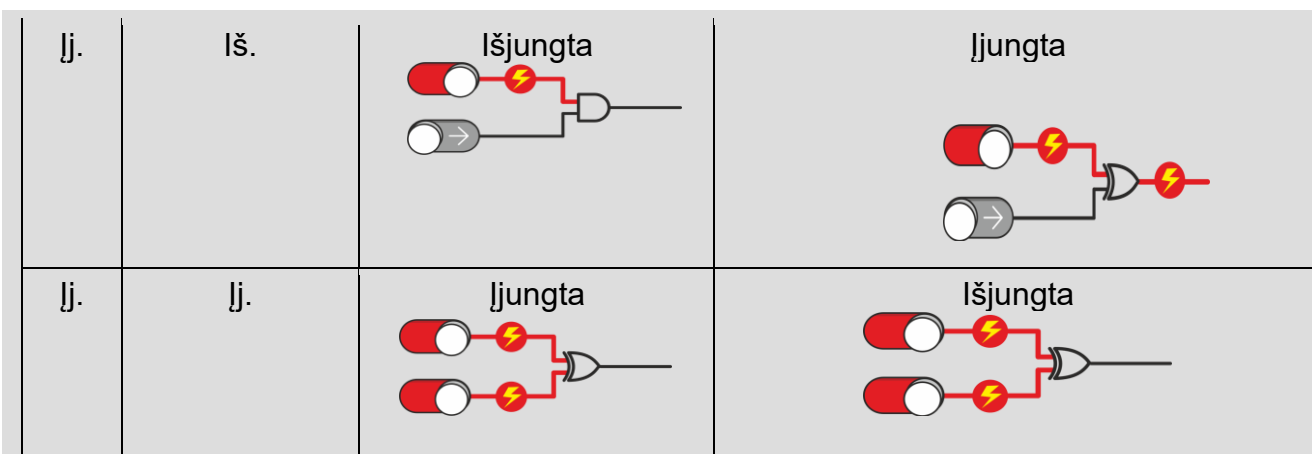
Tai informatika!

Laidais gali tekėti arba netekėti srovė, todėl jungikliai yra arba įjungti, arba išjungti. Kompiuterių moksle tokios būsenos atitinka loginio kintamojo reikšmę. Jos paprastai vadinamos *true* arba *false*, atitinkamai 1 arba 0.

Šiuolaikiniai kompiuteriai paprastai veikia tik su šiomis dviem būsenomis. Viena iš priežasčių yra ta, kad kompiuterio branduolyje yra milijardai tranzistorių, kurių galimi įėjimai ir išėjimai yra tik įjungti arba išjungti.

Logines grandines galima sudaryti iš kelių tranzistorių. Šioje užduotyje naudojamos dvi tokios grandinės: vienas iš komponentų yra AND loginis elementas, kurio išėjimas įjungiamas tik tada, kai įjungti abu įėjimai. Kita sudedamoji dalis yra išskirtinis XOR loginis elementas, kurio išėjimas yra įjungtas, kai įjungtas lygiai vienas iš dviejų įėjimų. Tai taip pat galite pavaizduoti lentele:

A Iš.	B Iš.	Išvestis AND Išjungta 	Išvestis XOR Išjungta
Iš.	Ij	Išjungta 	Ijungta



Kitas įprastas loginis elementas yra OR loginis elementas, kurio išėjimas bus įjungtas, kai įjungtas bent vienas iš dviejų įėjimų, ir išvesties XOR loginis elementas, kurio išėjimas yra įjungtas būtent tada, kai įėjimas nėra įjungtas. Dažnai naudojamas AND loginio elemento ir XOR loginio elemento derinys, nes jį galima sukurti naudojant labai nedaug tranzistorių.

Rezultatų lentelės yra tokios:

Įvestis A	Įvestis B	Išvestis AND	Išvestis XOR
Iš.	Iš.	Iš.	Iš.
Iš.	Iš.	Iš.	Iš.
Iš.	Iš.	Iš.	Iš.
Iš.	Iš.	Iš.	Iš.

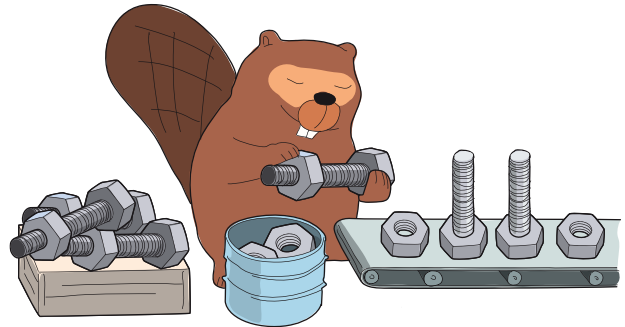
Įvestis	Išvestis be loginių elementų
Iš.	Iš.
Iš.	Iš.

Kompiuteris, sumaniai derindamas loginius elementus, gali labai greitai atlikti sudėtingus skaičiavimus. Loginiai elementai naudojami programavime: jei programos dalies vykdymas grindžiamas keliomis sąlygomis, šias sąlygas galima sujungti loginiais operatoriais, kurie veikia lygiai taip pat. Kartais programa turi priimti sprendimą, ką daryti toliau priklausomai nuo to, ar vienas dalykas (o kartais ir keli dalykai) įvyko anksčiau. Programuotojai tai įgyvendina naudodami savo programose sąlyginius (*if*) sakinius.

24. Veržlės ir varžtai

Bebras Benas dirba gamykloje „Bebro konstrukcijos“ prie veržlių ir varžtų surinkimo linijos.



Benas dirba taip:

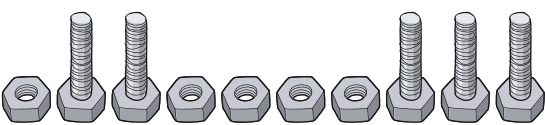
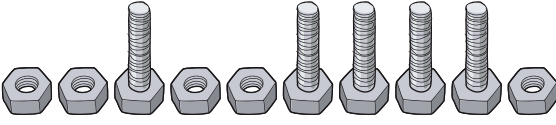
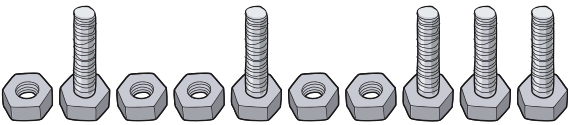
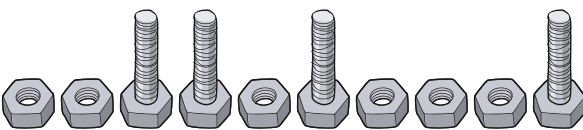


- Benas stovi ilgos konvejerio juostos, kurioje yra veržlių ir varžtų eilė, pradžioje .
- Beno užduotis – nuiminėti nuo konvejerio visas veržles ir varžtus po vieną.
- Jei Benas paima veržlę nuo konvejerio, jis ją įdeda į šalia esantį kibirą.
- Jei Benas paima varžtą nuo konvejerio, jis paima veržlę iš šalia esančio kibiro, užsuka veržlę ant varžto ir junginį padeda ant didelės dėžės.

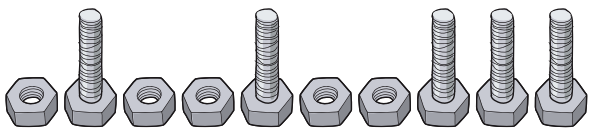
Dirbant Benui gali nutikti dvi nemalonios situacijos:

1. Jis paima varžtą nuo konvejerio juostos, o kibire nėra nė vienos veržlės.
2. Konvejerio juosta jau tuščia, o kibire vis dar yra veržlių.

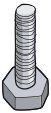
Pradžioje kibiras yra tuščias. Kurioje veržlių  ir varžtų  eilėje Benas, atlikdamas darbą iš kairės į dešinę, nuims visas detales ir nepatirs nemalonios situacijos?

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

Paaiškinimas

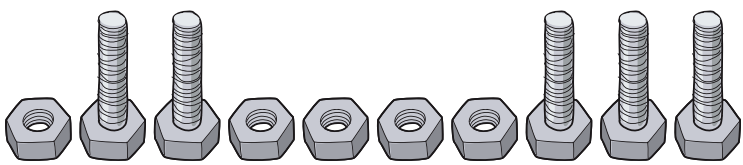
Teisingas atsakymas yra .

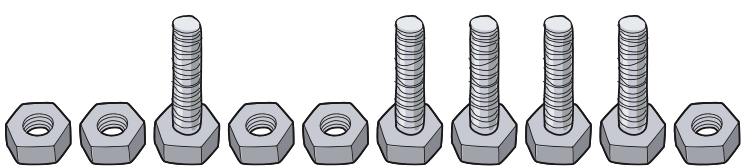
Galime stebėti varžtų ir veržlių judėjimą konvejeriu iš kairės į dešinę. Pažymėkime N = , B

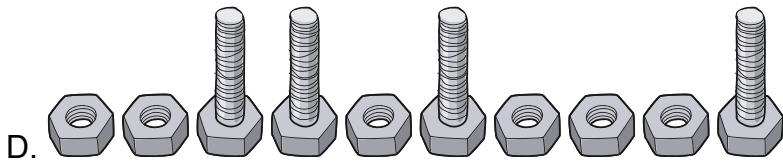
= 

Kibiras	Konvejeris
tuščias	N B N N B N N B B B
N	B N N B N N B B B
tuščias	N N B N N B B B
N	N B N N B B B
N N	B N N B B B
N	N N B B B
N N	N B B B
N N N	B B B
N N	B B
N	B
tuščias	tuščias

Panagrinėkime kitus atsakymus. Sutarkime nemalonią situaciją, kai Benas negali atlikti jokio veiksmo, vadinti klaida.:

A.  Nuėmus 3-ą detalę (varžtą) bus klaida, nes kibiras bus tuščias.

B.  Nuėmus paskutinį penktą varžtą bus klaida, nes nuimtos 4 veržlės bus panaudotos ir kibiras bus tuščias.



D. Konvejeriye yra 6 veržlės ir 4 varžtai, todėl bus klaida, nes kibire liks nepanaudotos 2 veržlės, nors visos detalės nuo konvejerio bus nurinktos.

Tai informatika!

Šioje užduotyje akcentuojamas nuleidžiamo automato (PDA) naudojimas. PDA – tai būdas aprašyti algoritmą, kuris remiasi dabartine būseną, bet taip pat turi neribotą atmintį – dėklą (angl. *stack*). Šioje užduotyje būseną yra arba veržlė, arba varžtas ant konvejerio juostos, o dėklas – kibiras, kuriame laikomos veržlės.

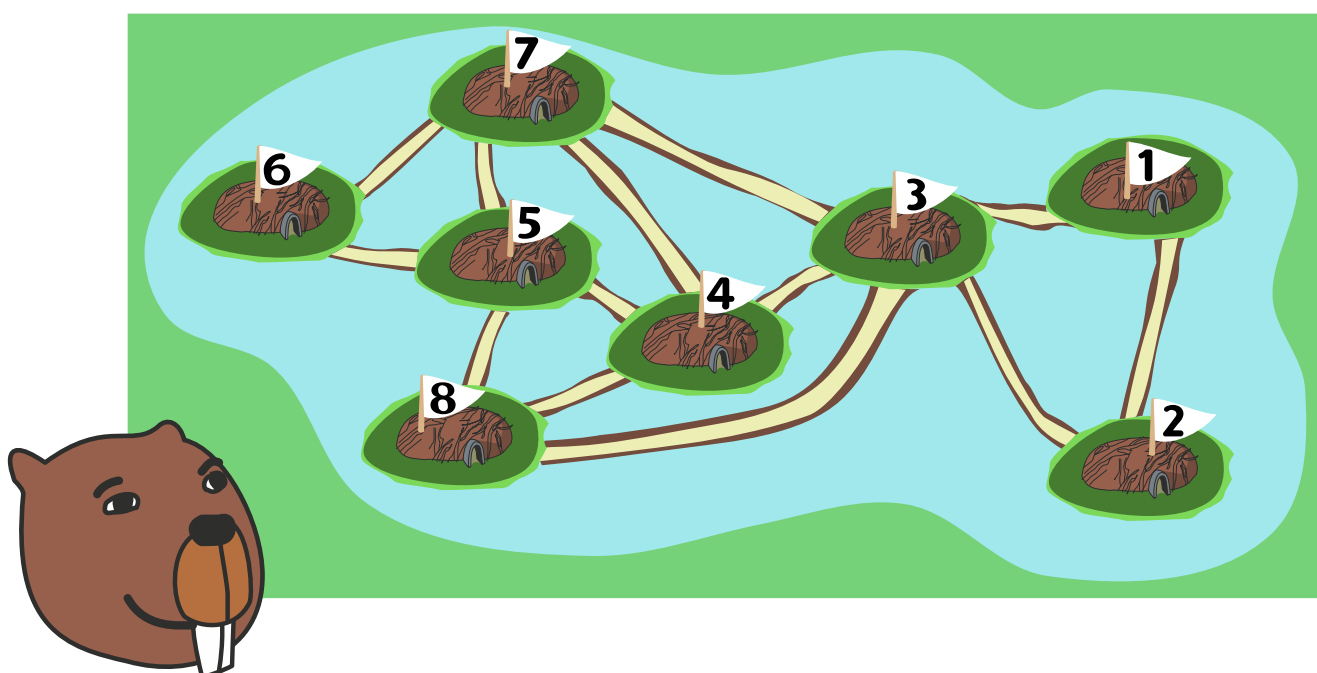
PDA gali būti naudojamas atpažinti arba analizuoti laisvo konteksto kalbas. Atpažinti arba analizuoti kalbą reiškia nustatyti, ar tam tikra simbolių seka priklauso kalbai. Šiuo atveju veržlės ir varžtus galime įsivaizduoti kaip subalansuotų skliaustų atvaizdą, čia $N=($ ir $B=)$. Paprastai subalansuoti skliaustai naudojami aritmetiniuose reiškiniuose. Nesubalansuotų skliaustų sekos gali būti $(((($ arba $)))($. Subalansuotų skliaustų aptikimas svarbus kompiliatoriams, nes daugelyje programavimo kalbų skliaustai naudojami įterptinėms sritims ir aritmetiniams reiškiniams nurodyti.

25. Marijos kaimynai

Bebras nori aplankyti savo draugę Mariją, tačiau nežino, kur ji gyvena. Laimei, jis turi žemėlapij ir šiek tiek informacijos:

- Du bebrai yra kaimynai, jei jų namus jungia takas.
- Kiekvienas iš šių trijų bebrų – Marija, Zenius ir Pijus – turi keturis kaimynus.
- Nikita turi tik du kaimynus – Zenių ir Pijų.

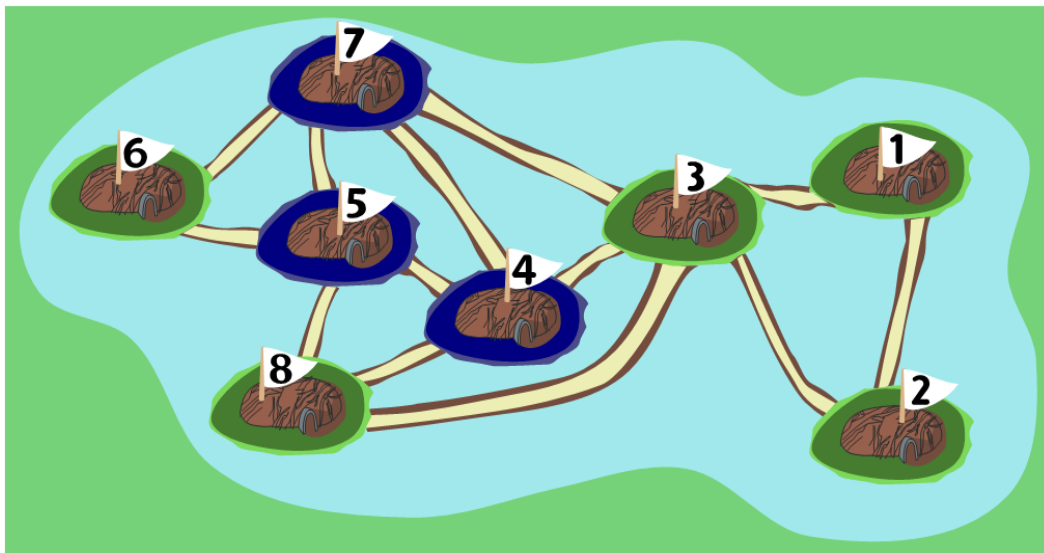
Kuris namas yra Marijos?



Paaiškinimas

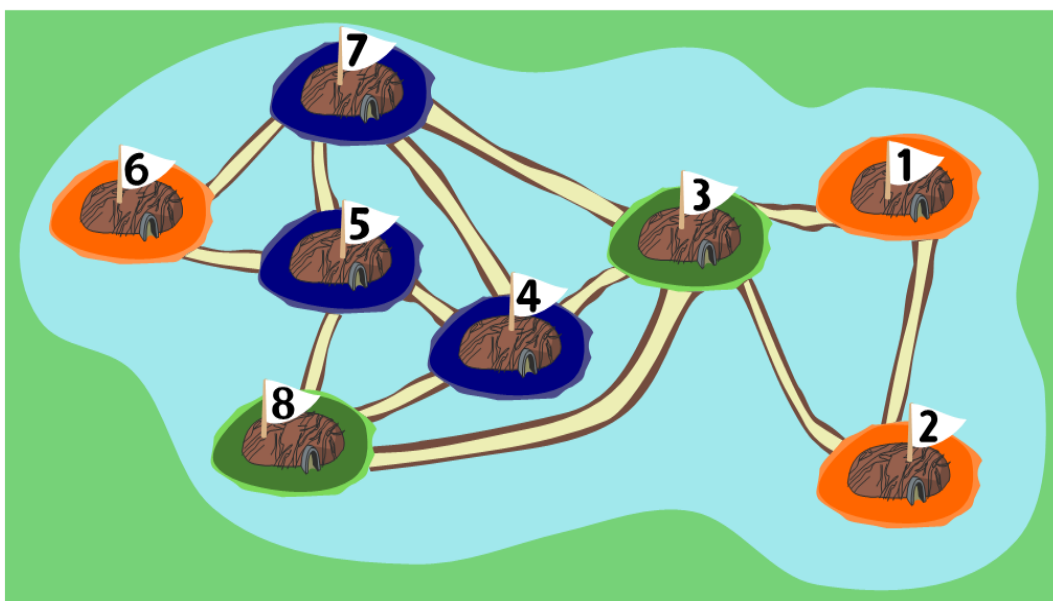
Teisingas atsakymas yra 4.

Sprendžiant šį uždavinį turime atkreipti dėmesį į kelius, einančius iš kiekvieno namo. Pirma, nustatome namus, į kuriuos veda keturi keliai. Tokie namai yra trys: 4, 5 ir 7.



Taigi Marija, Zenius ir Pijus gyvena viename iš šių trijų namų, tačiau turime išsiaiškinti, kuriame name gyvena Marija.

Kitos dvi sąlygos apibūdina Nikitos namus. Vadinasi, yra tik du keliai, kurie veda pas Nikitą. Taigi, Nikita gyvena viename iš namų, pažymėtų skaičiais 1, 2 arba 6.



Kadangi Nikita yra Pijaus ir Zeniaus kaimynė, tai:

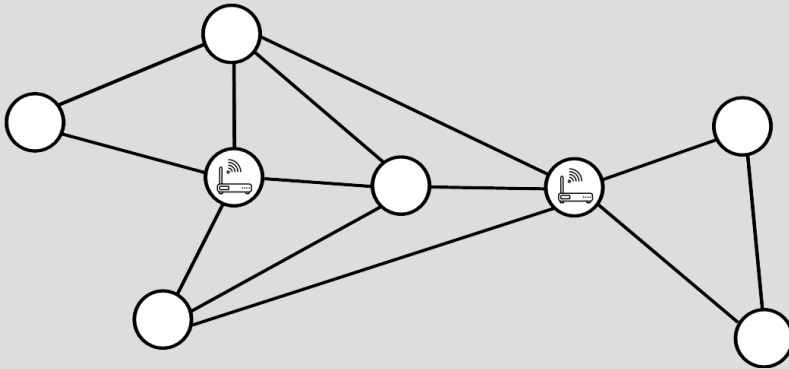
- Nikita gyvena name nr. 6.

- Zenius ir Pijus gyvena namuose, kurie pažymėti skaičiais 5 ir 7 (arba atvirkščiai).

Taigi, yra tik vienas namas su keturiais keliais, kuris gali būti Marijos namas. Tai yra namas, pažymėtas skaičiumi 4.

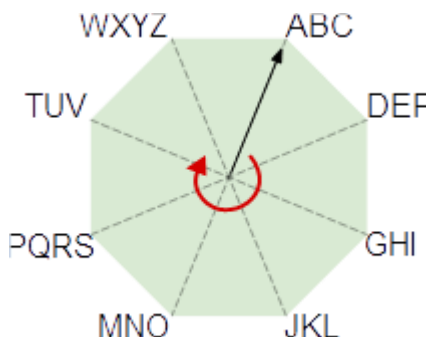
Tai informatika!

Grafų teorija – tai grafų, naudojamų objektų tarpusavio ryšiams modeliuoti, tyrimas. Grafa galima vaizduoti kaip mazgų (dar vadinamų viršūnėmis arba taškais), sujungtų briaunomis (dar vadinamomis jungtimis), rinkinį. Šiame uždavinyje namai yra viršūnės, o keliai yra briaunos. Grafi gali būti naudingi aprašant ir sprendžiant tinklo problemas, pavyzdžiui, ieškant tinkamos vietos maršrutizatoriui pastate arba užtikrinant, kad kiekvienas kaimynystėje esantis namas turėtų stiprų belaidžio interneto ryšio signalą.



26. Šifras: 8

Paveiksle pavaizduotas specialus ratas, kurio pagalba užšifruojami žodžiai.



Pradžioje rato rodyklė nustatoma ABC raidžių kryptimi. Vieną kartą pasukus rodyklę, ji pasisuka kampu, lygiu aštuntadaliui apskritimo.

Kiekviena raidė užšifruojama dviem skaitmenimis.

Pirmasis skaitmuo rodo, kiek kartų pasukama rodyklė (laikrodžio rodyklės judėjimo kryptimi), kad ji rodytų raidžių grupelę, kurioje yra reikiama raidė.

Antrasis skaitmuo nurodo, kelinta grupelėje yra užšifruojama raidė.

Pavyzdžiui, žodis PAAR užšifruojamas taip: 51-31-81-53.

Koks tekstas čia užšifruotas: 22-61-62-74?

- A) HANS
- B) HAUS
- C) HALLO
- D) HALS
- E) HAUT

Paaiškinimas

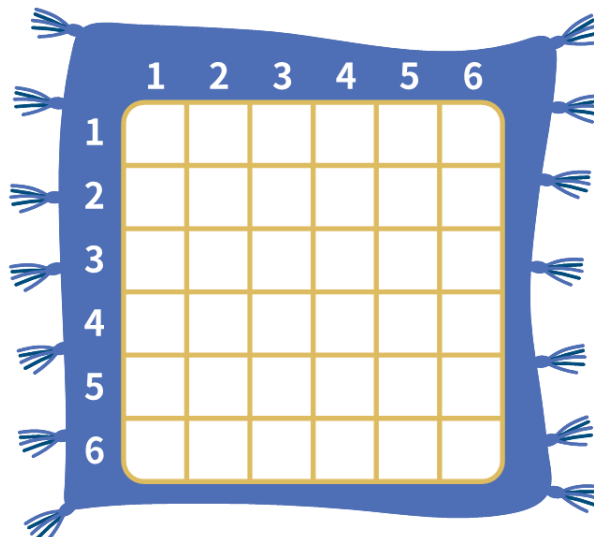
Teisingas atsakymas: B (HAUS). 22 reiškia, kad rodyklė pasukta nuo grupės ABC į GHI (pirmasis skaitmuo 2) ir paimta antroji raidė H (antrasis skaitmuo 2). 61 reiškia, kad rodyklė nuo grupės GHI pasukta į grupę ABC (pirmasis skaitmuo 6) ir paimta antroji raidė A (antrasis skaitmuo 1). 62 reiškia, kad rodyklė nuo grupės ABC pasukta į TUV (pirmas skaitmuo 6) ir paimta antroji raidė U (antras skaitmuo 2). 74 reiškia, kad dabar rodyklė nuo grupės TUV pasukta į PQRS (pirmas skaitmuo 7) ir paimta ketvirtoji raidė S (antras skaitmuo 4). Tai reiškia, kad atsakymas B (HAUS) yra teisingas. Šį sprendimą buvo galima rasti ir greičiau. Atsakymas C (HELLO) netinka, nes jį sudaro penkios raidės, o šifrą sudaro tik keturios raidės. Kadangi paskutinė raidė koduojama antruoju skaitmeniu 4, ji gali būti tik S arba Z. Tai atitinka tik A, B ir D atsakymus. Prieš ją esanti raidė turi būti iš raidžių bloko, esančio septyniais posūkiiais prieš laikrodžio rodyklę, t. y., iš bloko TUV. Tai reiškia, kad gali būti tik atsakymas HAUS.

Tai informatika!

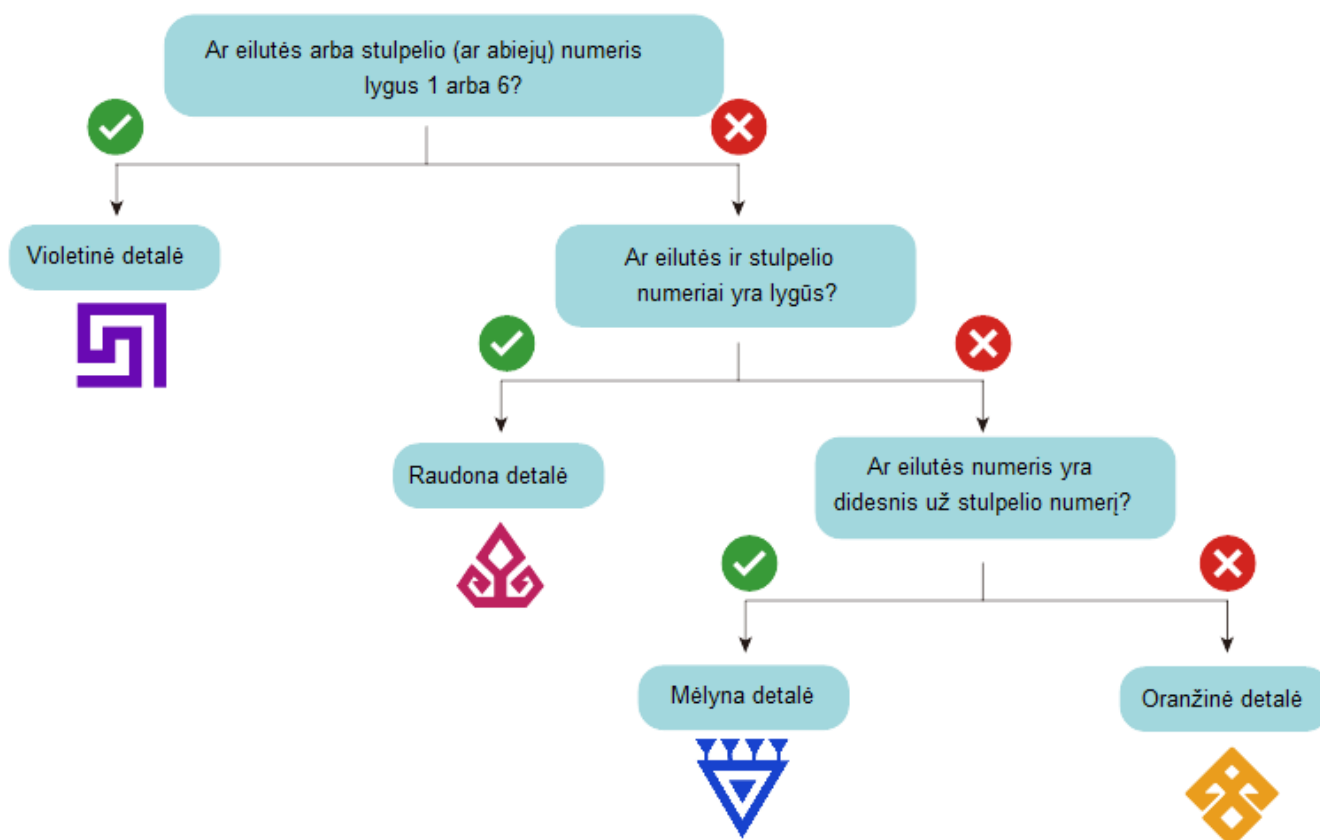
Vienas iš būdų apsaugoti duomenis nuo pašalinių asmenų yra šifravimas. Kriptografija pradėta naudoti jau prieš 3500 metų. Vienas paprasčiausių šifravimo būdų – kiekvieną raidę pakeisti kita raide. Tūkstančius metų žmogus stengėsi paslėpti informaciją taip, kad ją iššifruoti galėtų tik gavėjai. Tai, kas prasidėjo nuo popieriaus juostelių, apvyniotų aplink lazda, tęsiama naudojant transpozicijos šifrus (pvz., Cezario kodą) ir polialfabetinius šifravimo metodus (pvz., Vigenero metodą), peraugo į šiuolaikinę viešojo rakto kriptografiją (pvz., GnuPG, kurioje, be kita ko, naudojamas RSA metodas). Šios užduoties šifravimo metodas yra polialfabetinis šifravimo metodas, nes ta pati raidė nebūtinai užšifruojama tuo pačiu šifro tekstu: pavyzdyje raidė A pradžioje užšifruojama kaip 31, o pabaigoje – kaip 81. Iš esmės visus šiuos šifravimo būdus šiandien galima greitai ir lengvai iššifruoti naudojant kompiuterius. Tačiau šiuo atveju iššifravimas yra kuo paprasčiausias: tekstui užšifruoti yra tik vienas raktas. Net jei būtų galima nustatyti, kad rodyklės pradinė padėtis prasidėtų ne nuo ABC, o nuo kokio nors bloko, turėtumėte tik aštuonis skirtingus raktus. Net Cezario kodas, kuriam jau daugiau nei 2000 metų, yra saugesnis. Galima teigti, kad esmė yra ne raktas, o šifravimo metodas. Tačiau Kerckhoffo principas, suformuluotas 1883 m. ir naudojamas iki šiol, aiškiai parodo, kad kriptosistemos saugumas neturi būti grindžiamas šifravimo metodo laikymu paslapyje, nes jį pernelyg lengvai gali sužinoti kiti.

27. Kilimo audimas

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



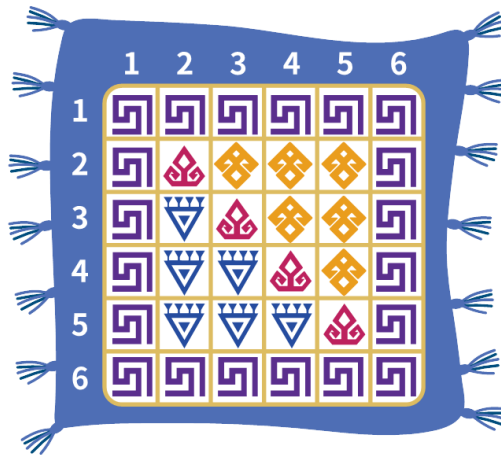
Kiekviename langelyje audžiama tokia ornamento detalė, kuri gaunama atsakius į šiuos klausimus:



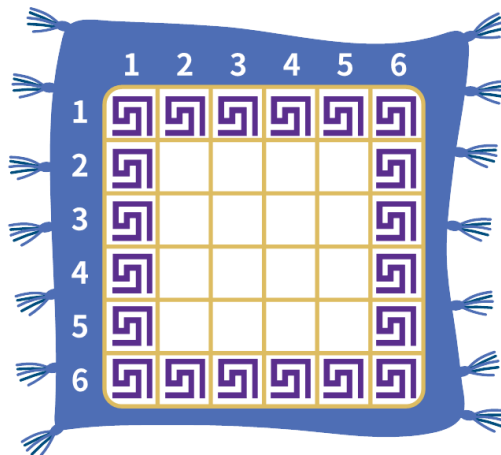
Kaip atrodys išaustas kilimas?

Paaiškinimas

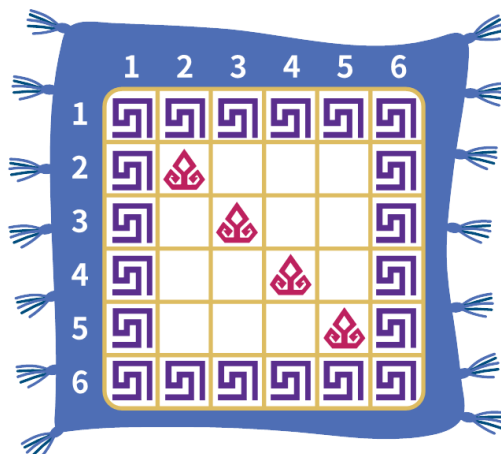
Teisingas atsakymas:



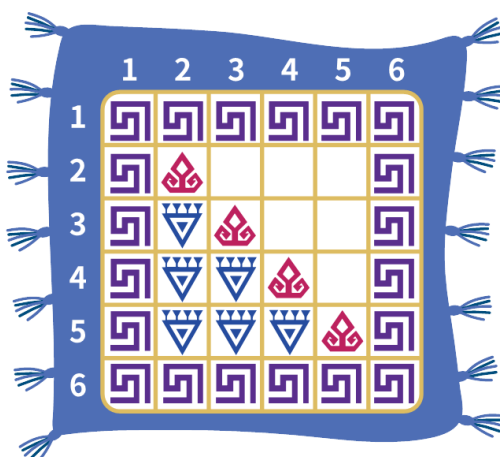
Pirmuoju klausimu nustatoma, kad 1-osios ir 6-osios eilučių bei 1-ojo ir 6-ojo stulpelių langeliuose audžiamos violetinės detalės. Taip gaunamas toks simbolių išdėstymas:



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



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



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

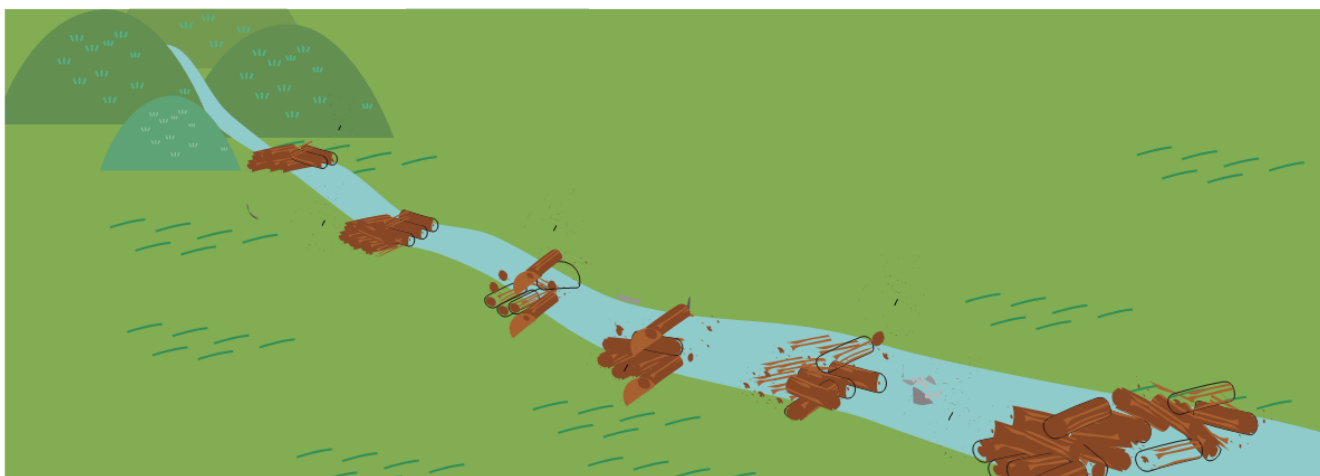
Tai informatika!

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

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

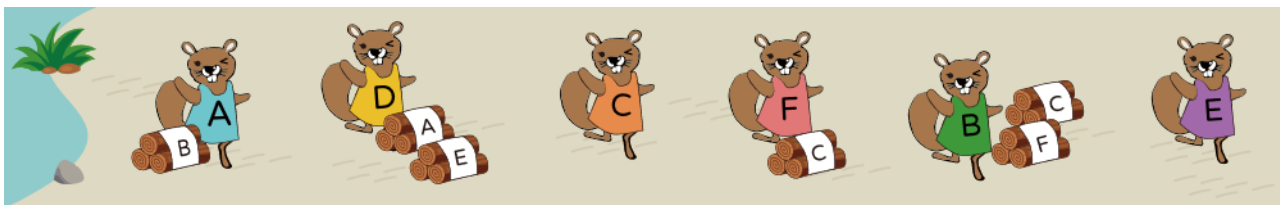
Pasinaudodami dirbtiniu intelektu, algoritmai gali atlikti automatinius išvedimus (vadinamus automatiniu samprotavimu) ir naudoti matematinius ir loginius tikrinimus, kad nukreiptų programos atlikimą įvairiais keliais (tai vadinama automatiniu sprendimų priėmimu).

28. Bebriukių užtvanka



Šešios bebrukės (A, B, C, D, E, ir F) Bebrų upelyje pastatė po savo užtvanką. Netikėtai vieną dieną buvo liūtis: kai kurie rąstai buvo nuplauti nuo užtvankų ir srovės nunešti upeliu. Laimei, visi rąstai buvo pažymėti užtvankų statytojų raidėmis. Pavyzdžiui, ant rąstų, kuriuos panaudojo bebrukė A, buvo raidė A.

Po audros bebrukės surinko nuplautus rąstus (žr. paveikslėljį):



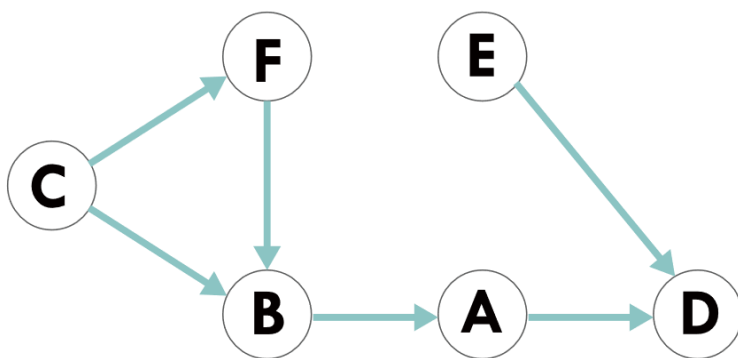
Sprendžiant iš rąstų, kuriuos kiekviena bebrukė surinko, kokia tvarka upėje pasroviui buvo pastatytos užtvankos?

- A. $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$
- B. $C \rightarrow B \rightarrow F \rightarrow A \rightarrow D \rightarrow E$
- C. $C \rightarrow F \rightarrow B \rightarrow D \rightarrow A \rightarrow E$
- D. $E \rightarrow C \rightarrow F \rightarrow B \rightarrow A \rightarrow D$

Paaiškinimas

Teisingas atsakymas yra (D): $E \rightarrow C \rightarrow F \rightarrow B \rightarrow A \rightarrow D$

Kadangi raštai plaukė pasroviui ir bebrūkė A rado bebrūkės B raštus, galime numanyti, kad bebrūkės B užtvanka stovi pirmiau nei bebrūkės A. Jei rodyklėmis nubrėšime ryšį tarp dviejų gretimų bebrūkių, tai galėsime nustatyti kiekvienos užtvankos padėtį:



Kaip matome, nėra rodyklės, jungiančios C ir E taškus, todėl C arba E užtvanka gali būti pirma. Todėl atsakymo A variantas neteisingas.

Atsakymo B variantas neteisingas, nes F turi būti pirmiau nei B, ką rodo rodyklė iš F į B.

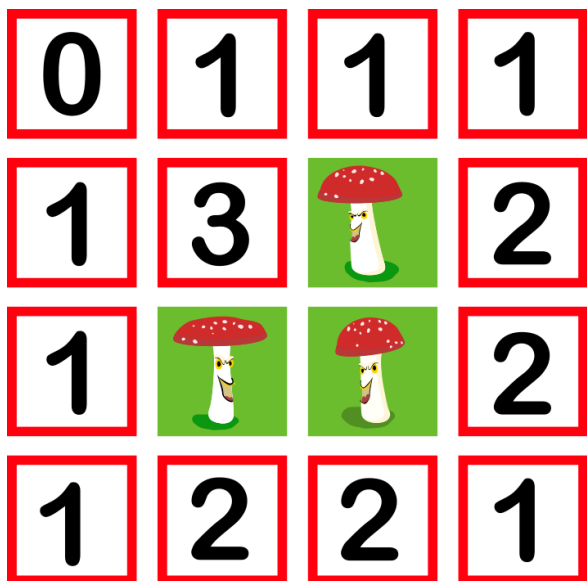
Atsakymo C variantas neteisingas, nes D turi būti pirmiau nei E, ką rodo rodyklė iš C į B, tada į A ir į D. Teisingas atsakymas yra D, nes tik jis atitinka pavaizduotų rodyklių ryšį.

Tai informatika!

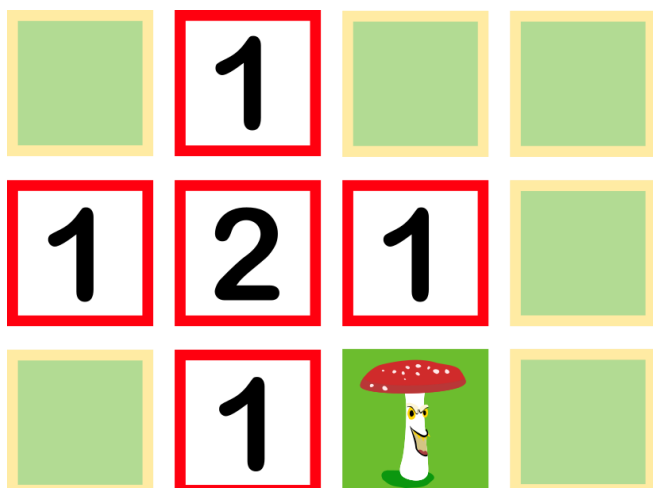
Orientuotasis nekontūrinis grafas susideda iš viršūnių ir nukreiptų briaunų be ciklų. Šioje užduotyje galime kiekvieną bebrūkę laikyti viršūne. Kiekvieno rašto plaukimo kryptį – orientuota briauna. Bebrūkių rikiavimas pagal upės srovę teisinga tvarka vadinamas topologiniu rikiavimu: taip viršūnės rikiuojamos pagal briaunų kryptis. Gali būti įvairių rikiavimo būdų, atitinkančių topologinę tvarką. Vienas iš būdų rasti topologinį rikiavimą yra Kahno algoritmas. Pirmiausia pasirenkame viršūnę S, kuri neturi įeinančių briaunų. Tada ištrinkime viršūnę S ir visas briaunas, išeinančias iš jos. Kartojame procesą, kol grafas tampa tuščias. Kompiuterių moksle topologinis rikiavimas naudojamas procesams planuoti, duomenų keitimui į nuosekliają formą (angl. *data serialization*) ir sprendžiant simbolių priklausomybę jungtyse.

29. Saugokitės šungrybių

Žaidimo „Saugokitės šungrybių“ pradžioje matoma lygiai viena musmirė. Visi kiti lentos laukai yra uždengti. Paspaudus ant lauko, pasirodo arba kita musmirė, arba gretimuose laukuose esančių musmirių skaičius. Jei atidengsite visus laukus, kuriuose nėra paslėptų musmirių, laimėsite. Štai visiškai atidengtos lentos pavyzdys:

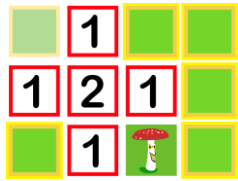


Pradėjote naują žaidimą (paveikslukas žemiau):



Kuriuose laukuose tikrai nėra musmirių?

Paaiškinimas

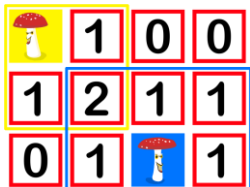


Teisingas atsakymas:

Norėdami paaiškinti teisingą atsakymą, uždengtus laukus pažymime raidėmis. Be to, sakome, kad skaičius N lauke yra „išnaudotas“, jei N kaimyninių šio skaičiaus laukelių jau yra neuždengta musmirėmis; tuomet kituose kaimyniniuose laukeliuose nebegali būti daugiau musmirių.



Laukelyje D nėra musmirės, nes dešinėje pusėje esantis skaičius 1 yra išnaudotas. Laukeliuose B, C, E ir F nėra musmirės, nes bendras šių laukų kaimyninis skaičius 1 yra išnaudotas. Laukelyje A yra musmirė, nes priešingu atveju kaimyniniai skaičiai 1, 2 ir 1 neteisingai rodytų musmirių skaičių jų kaimyniniuose laukeliuose.



Taigi, A lauke slepiasi musmirė. B, C, D, E ir F laukai gali būti atidengti.

Tai informatika!

Tam, kad išspręstumėte tokius galvosūkius, kaip šioje „Bebro“ užduotyje, tereikia mąstyti logiškai. Dažnai reikia pradėti nuo prielaidos ir vadovautis logika. Jei randate klaidą (pavyzdžiui, kad „1“ reiškia elementą ir dėl to negalite turėti kito elemento), grįžkite ir ištaisykite, kadangi prielaida buvo klaidinga. Tai labai populiarus algoritmas, dar vadinamas grįžimo algoritmu. Grįžimo algoritmo trūkumas yra tas, kad kartais tenka daug kartų grįžti atgal, o tai reikalauja laiko ir išteklių. Ir, žinoma, reikia nustatyti aiškią pabaigą, jei sprendinys neegzistuoja.

30. Spalvota dėlionė

Kristupas turi trijų spalvų šešiakampės dėlionės detales. Trys detalės sudėtos į trikampio formos figūrą (žr. paveikslėlį). Jos turi būti tos pačios spalvos arba visos skirtingų spalvų.



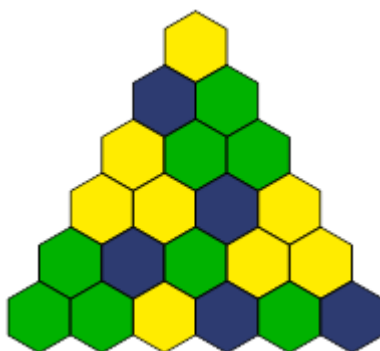
Kristupas nori sudėti didelę trikampę dėlionę, pavaizduotą paveiksle. Kokios spalvos turi būti viršutinė detalė?



Paaiškinimas

Kai žinome dviejų šalia esančių detalių spalvas, tada aišku, kokia turi būti virš jų esančios detalės spalva. Pavyzdžiui, panagrinėkime, kokios spalvos turėtų būti antroje eilėje esanti kairiausia dėlionės detalė. Kadangi dvi detalės, esančios po šia nežinoma detale, yra žalios, tad ir ši detalė taip pat turi būti žalia. Greta dešinėje esanti detalė turi būti mėlyna, nes dvi po ja esančios detalės yra žalios ir geltonos spalvos.

Šį metodą, eidami nuo apačios iki viršaus, galime taikyti visai dėlionei. Rezultatas turėtų būti toks:



Atkreipiame dėmesį, kad kiekvienoje dėlionės eilėje galima nustatyti bet kurioje vietoje esančių detalių spalvas.

Tai informatika!

Ši užduotis yra gera iliustracija informacinių technologijų įgūdžio, vadinamo modelio atpažinimu, kuris specifiškai naudojamas rasti panašius arba bendrus elementus. Norint sudėti korteles eilės tvarka reikia lyginti kortelių poras ir surasti tas, kurios turi panašiausias modelius. Supratus, kaip paveikslukas yra kuriamas (vis pridėdant baltų geometrinių formų), galima sudėti visus paveikslukus eilės tvarka.

31. Sąrašai

Duotas skaičių sąrašas X:

X

1	2	3	4	5
3	5	2	4	1

Sąrašo X pozicijose nuo 1 iki 5 yra šie skaičiai: 3, 5, 2, 4, 1.

Tam tikroje pozicijose esantį skaičių nurodome skliausteliuose įrašydami sąrašo pavadinimą ir skaičiaus poziciją. Pavyzdžiui, X sąrašo 2 pozicijoje esantį skaičių nurodome taip: (X 2). Šiuo atveju (X 2) = 5.

Pozicija gali būti nurodyta netiesiogiai, pavyzdžiui, (X (X 2)) = (X 5) = 1.

Duoti kiti trys sąrašai: A, B ir C.

A

3	2	4	1	5
---	---	---	---	---

B

5	4	1	3	2
---	---	---	---	---

C

2	5	4	3	1
---	---	---	---	---

Kokį skaičių nurodo užrašas (A (B (C 3)))?

Paaiškinimas

Atsakymas: 4.

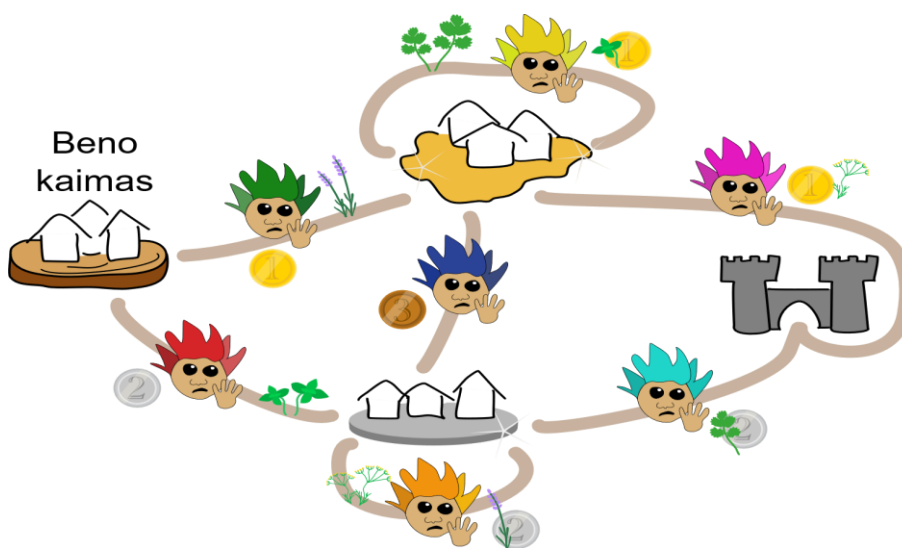
$(C\ 3) = 4$, todėl $(B\ (C\ 3)) = (B\ 4) = 3$, taigi $(A\ (B\ (C\ 3))) = (A\ (B\ 4)) = (A\ 3) = 4$

Tai informatika!

Duomenų struktūros yra labai svarbios programavimui. Duomenų struktūros, kaip ir daugelis kitų informatikos ir programavimo objektų, gali būti tarpusavyje sujungtos, todėl vieno sąrašo elementas gali apibrėžti poziciją kitame sąrašo. Šis netiesioginis pozicijų aprašymo būdas yra galinga sąvoka. Iš pradžių ji gali atrodyti paini, tačiau nėra sudėtinga, jei reikšmes skaičiuosite žingsnis po žingsnio.

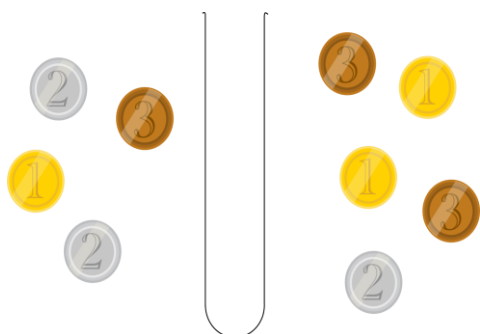
32. Gobšūs troliai

Benas nori iš savo kaimo nueiti į pilį, bet keliuose tarp kaimų laukia gobšūs troliai. Visi jie reikalauja vienos konkrečios monetos (☉, ☽, ☾) iš kiekvieno praeivio. Kai kurie iš jų papildomai nori ir vienos konkrečios žolelės (☘, ☙, ☚, ☛). Laimei, šių žolelių galima rasti miškuose ant kai kurių kelių tarp kaimų. Paveiksle parodyta, kurios monetos nori kiekvienas trolis ir, jeigu jam reikia, žolelės. Neįmanoma nuskinti žolelės prieš tai nesumokėjus tame kelyje esančiam troliu.



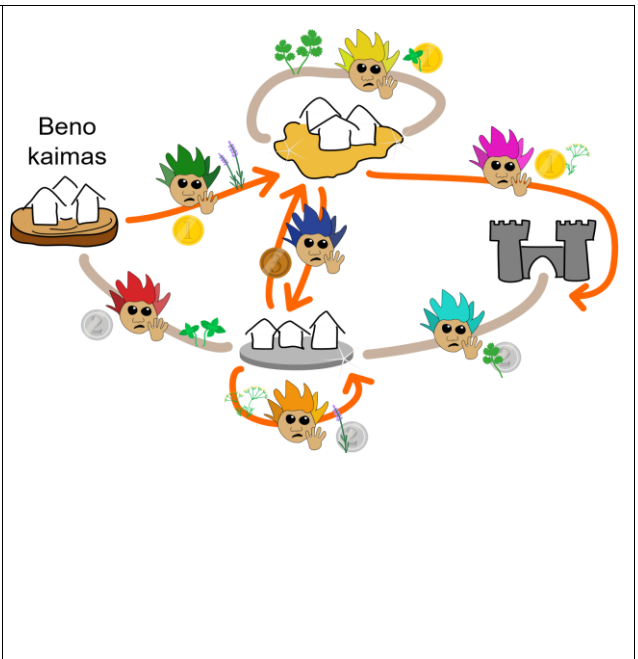
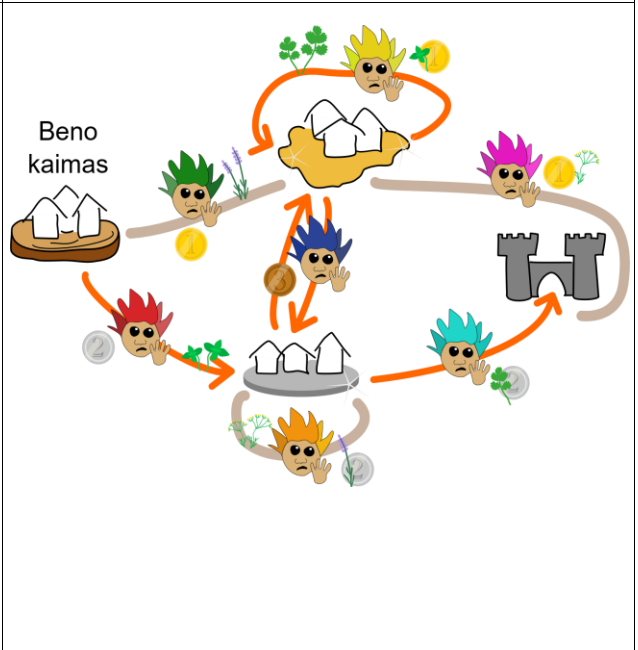
Benas turi vamzdelio formos dėtuvę monetoms laikyti ir nori ją užpildyti prieš išvykdamas. Kadangi jis gali imti tik vieną monetą po kitos, jis turi užpildyti monetų dėtuvę tiksliai ta tvarka, kokia reikės imti monetas. Benas keliauja su nedidele kuprine, kurioje laiko miške rastas žoleles. Kuprinė yra labai siaura, todėl jis gali pasiekti tik pačiame kuprinės viršuje esančią žolelę. Išvykstant Beno kuprinė tuščia.

Į Beno monetų dėtuvę sudėkite penkias monetas reikiama tvarka, kad pakeliui į pilį jis galėtų susimokėti gobšiams troliams. Nepamirškite žolelių, kurių jis turi pririnkti.



Paaiškinimas

Galimi du sprendimo variantai:

<p>1 3 2 3 4</p>	<p>Benas pirmiausia vyksta į Aukso kaimą. Pakeliui jis sumoka vieną auksinę monetą ir prisirenka levandų. Tada jis eina į Sidabro kaimą ir pakeliui sumoka vieną varinę monetą. Tada jis keliauja per mišką, kur turi sumokėti vieną sidabrinę monetą ir levandų, bet gali rinkti krapus. Po to jis vėl keliauja į Aukso kaimą, už kelią sumoka vieną varinę monetą. Iš ten jis eina į pilį, nes turi vieną auksinę monetą ir krapų.</p>	
<p>2 3 4 3 2</p>	<p>Benas pirmiausia vyksta į Sidabro kaimą. Pakeliui jis sumoka vieną sidabrinę monetą ir prisirenka bazilikų. Tada jis eina į Aukso kaimą ir pakeliui sumoka vieną varinę monetą. Tuomet jis keliauja per mišką, kur turi sumokėti vieną auksinę monetą ir bazilikų, tačiau ten jis gali rinkti kalendrą. Po to jis vėl keliauja į Sidabro kaimą, pakeliui sumoka vieną varinę monetą. Iš ten jis eina į pilį, nes turi vieną sidabrinę monetą ir kalendrą.</p>	

Tai vienintelės sprendimo galimybės, nes norint patekti į pilį Benui reikia arba krapų, arba kalendros. Norėdamas gauti krapų, jis pirmiausia turi surinkti levandų. Kadangi monetos yra tik penkios, šiuo atveju jam tinka tik 1-as sprendimas. Norėdamas gauti kalendros, jis pirmiausia turi surinkti bazilikų, tokiu atveju jam tinka tik 2-as sprendimas. Kiekvienas kitas būdas jam kainuotų daugiau nei penkias monetas.

Tai informatika!

Kaimus, pilį ir kelius tarp jų galima pavaizduoti grafo struktūra. Kiekvienas kelias turi tam tikrą kainą (monetą) ir jame galima ką nors įdėti arba paimti iš kuprinės, kuri veikia kaip stekas (angl. *stack*, liet. atitikmuo – dėklas). Stekas yra duomenų struktūra, veikianti pagal principą „paskutinis į – pirmas iš“, palaikanti šias operacijas: objekto padėjimas į steko viršų (*push*) ir objekto pašalinimas iš steko viršaus (*pop*).

Šio uždavinio grafą galima laikyti „stumiamuoju“ automatu (angl. *push-down automata*, PDA). Stumiamąjį automata sudaro skirtingos būsenos (čia vaizduojami kaimai ir pilis), įvesties abėcėlė (monetos), steko abėcėlė (žolelės), perėjimo funkcijų rinkinys (keliai), pradinė būsena (Beno kaimas) ir priėmimo būsenų rinkinys (pilis). Išskirtina tai, kad visi keliai eina į abi puses – tai nėra būdinga stumiamiesiems automatams.

Teorinės informatikos moksle stumiamasis automatas yra automatų tipas, kuriame naudojamas stekas. Stumiamieji automatai yra susiję su specialia formalių kalbų klase – bekontekstinėmis kalbomis.

33. Klausyk ir eik

Neregė Toma vaikšto miesto gatvėmis naudodamasi kalbančiais akiniais. Kalbantys akiniai turi kameras ir išmaniąją objektų atpažinimo sistemą. Jie atpažįsta žemėlapiuose pavaizduotus keturių tipų objektus: namą, medį, kelią ir veją. Kai Toma įžengia į kelio langelį, kalbantys akiniai pasako, kas yra Tomos kairėje, po to – kas priešais ją ir, galiausiai, kas dešinėje. Pavyzdžiui: „medis, kelias, namas“.

Toma pradeda kelionę starto langelyje su ruda trikampiui, ji žiūri žemėlapiu eilutės kryptimi.

Toma klausosi, ką jai kalba akiniai
(pradedant starto langeliu):

medis kelias namas

kelias kelias veja

medis kelias medis

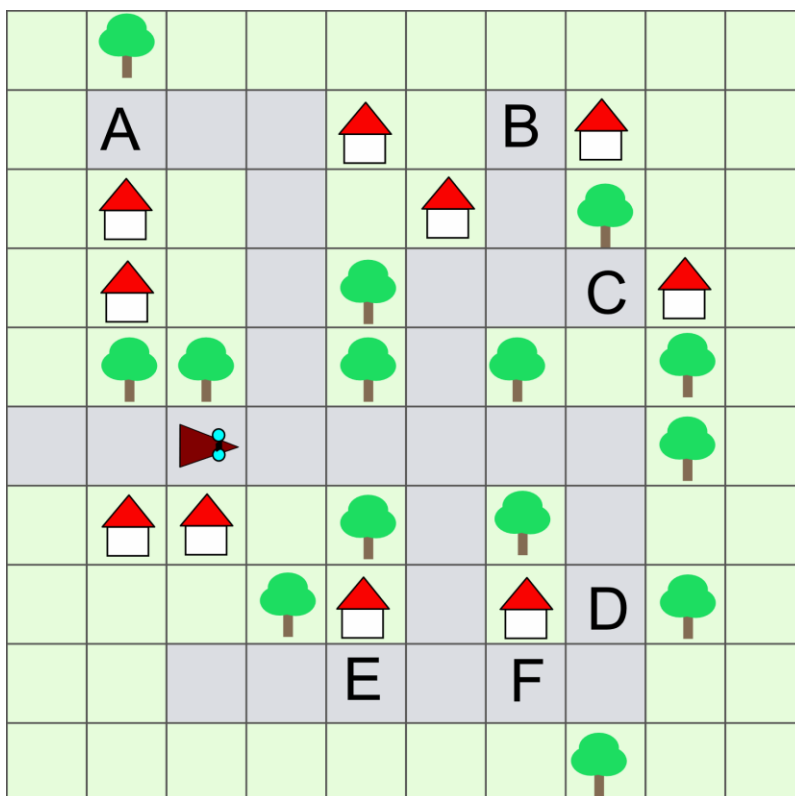
kelias kelias kelias

medis kelias medis

medis namas kelias

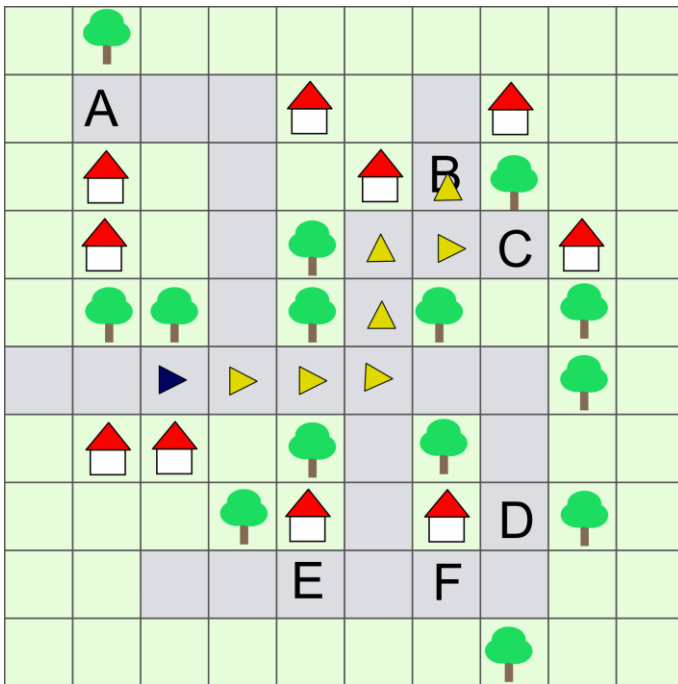
kelias kelias medis

namas kelias medis



Į kurį langelį, pažymėtą raide,

įžengė Toma, išklausiusi paskutinius akinių žodžius?





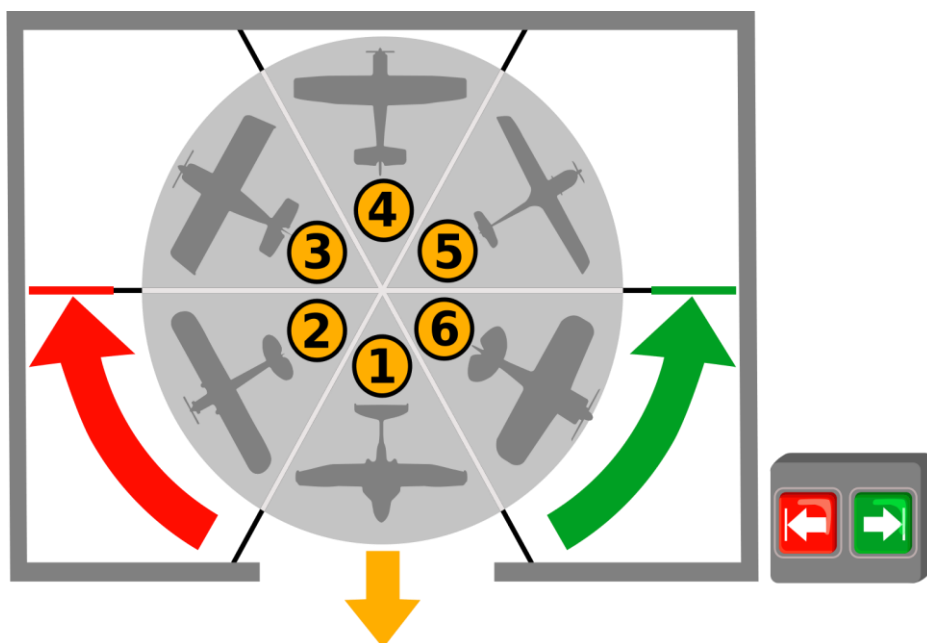
Tai informatika!


Užduotis parodo, kad pakanka žinoti tik kalbančių akinių Tomai pasakytų žinučių registrą, kad būtų galima atkurti galutinį jos pasiektą langelį iš kelių galimų variantų. Apskritai galima manyti, kad šis registras nėra pakankamai informatyvus, kad būtų galima pasakyti, kur Toma yra šiuo metu, tačiau papildomų žinių apie aplinką, kurioje ji yra, turime daug ir galime atsekti jos kelionės tikslą. Visada sunku spręsti, kiek jautri gali būti asmeninė informacija, ir tai ypač pasakytina apie erdvinius ar geografinius duomenis.

Objektų atpažinimas yra svarbi informatikos mokslo tema. Dirbtinio intelekto sistemą galima išmokyti atpažinti tokius objektus, kaip namai, medžiai ir kelio danga. Tikriausiai jūsų išmaniojo telefono fotoaparatas gali atpažinti veidus ir pažymėti juos rėmeliu. Autonominis automobilis turi atpažinti asmenis, kitus automobilius, kelio ženklus ir kitus aplinkos objektus, kurie yra svarbūs jo priimamiems sprendimams. Siekiant padėti akliesiems ir silpnaregiams, šiuolaikinėje saityno technologijoje naudojamas automatinis alternatyvusis tekstas (angl. *automatic alt text*, AAT). Tam taikoma objektų atpažinimas, kad automatiškai sukurtų žodinius vaizdų aprašymus („alt“ tekstus). Kalbantys akiniai vis dar yra mokslinė fantastika, tačiau esama realių projektų, kuriais bandoma įgyvendinti šią idėją.

34. Karuselė

Aerodromo angare ant besisukančių apvalių grindų stovi šeši lėktuvai. Grindys pasukamos į kairę arba į dešinę naudojantis valdymo pultu su dviem rodyklėmis  . Vienu mygtuko paspaudimu grindys pasukamos lygiai viena pozicija į kairę arba į dešinę. Pro angaro vartus gali išvažiuoti vienas lėktuvas. Grindys sukasi labai lėtai, todėl mažiau mygtukų paspaudimų padeda išvengti lėktuvų vėlavimo.



Paveiksle pavaizduota sunumeruotos lėktuvų stovėjimo vietos (1 stovėjimo vieta visada yra prie vartų). Norint, kad visi lėktuvai kuo greičiau išvažiuotų iš angaro, pulto rodyklių mygtukus reikia paspausti penkis kartus. Pavyzdžiui, jei norime, kad lėktuvai išvažiuotų šitokia tvarka: 1, 6, 5, 4, 3, 2, penkis kartus paspaudžiame .

Bet jei norime, kad lėktuvai išvažiuotų kita tvarka, gali prireikti ir daugiau mygtuko paspaudimų.

Kokia tvarka lėktuvai turėtų išvažiuoti iš angaro, kad pulto mygtukų paspaudimų skaičius būtų didžiausias?

Pateikite vieną iš tokių blogiausių atvejų įrašydami į langelius lėktuvų stovėjimo angare vietų numerius.

Paaiškinimas

Yra dvi blogiausios lėktuvų išleidimo eilės: 4 1 3 6 2 5 ir 4 1 5 2 6 3.

Blogiausią atvejį, taigi ir teisingą atsakymą, galima rasti pasirinkus kitą vietą, kuri visada yra toliausiai nuo vartų. Šiuo atveju iššūkis – įsivaizduoti posūkio rezultatą ir kaip atsilaisvina vieta. Kadangi grindų plokštumą galima pasukti abiem kryptimis, yra daugiau nei vienas teisingas sprendimas; esant šešioms plokštumoms yra du teisingi sprendimai.

4 1 3 6 2 5 :

Pirmiausia, norint pasiekti 4 padėtį, reikia paspausti tris mygtukus (į kairę arba į dešinę).

Paskui, norint pasiekti 1 padėtį, reikia paspausti tris mygtukus (kairę arba dešinę)

Toliau, norint pasiekti 3 padėtį, reikia paspausti du kartus į dešinę

Toliau, norint pasiekti 6 padėtį, reikia paspausti tris kartus (kairę arba dešinę)

Tada, norint pasiekti 2 padėtį, reikia paspausti du kartus į dešinę

Galiausiai, norint pasiekti 5 padėtį, reikia paspausti tris kartus (kairę arba dešinę)

4 1 5 2 6 3:

Pirmiausia, norint pasiekti 4 padėtį, reikia paspausti tris mygtukus (į kairę arba į dešinę).

Paskui, norint pasiekti 1 padėtį, reikia paspausti tris mygtukus (kairę arba dešinę)

Paskui norint pasiekti 5 padėtį, reikia paspausti du kartus į kairę

Toliau norint pasiekti 2 padėtį, reikia paspausti tris kartus (į kairę arba į dešinę)

Tada, norint pasiekti 6 padėtį, reikia paspausti du kartus į kairę

Galiausiai, norint pasiekti 3 padėtį, reikia paspausti tris kartus (kairę arba dešinę)

Tai informatika!

Apvalaus angaro privalumas – jame galima statyti lėktuvus taip, kad būtų sutaupyta daug vietos. Tačiau kaip dėl laiko? Jei lėktuvai naudojami stovėjimo vietomis pagal pilotų atvykimo tvarką, pilotai gali greitai parkuotis ir kilti vienas po kito, negaišdami laiko. Lėktuvų pakilimo procesui tai yra geriausias atvejis. Tačiau, kaip parodė ši užduotis, daugumai procesų būna ir blogiausių atvejų. Blogiausiu atveju prarandamas laikas, todėl tarp vietos ir laiko reikia siekti balanso.

Kompiuterių moksle labai svarbus vaidmuo tenka proceso efektyvumui įvertinti. Nors geriausius problemos atvejus yra gerai žinoti, efektyvumui apibūdinti svarbesnis yra blogiausias atvejis. Efektyvumas reiškia, kad išteklių – laikas, energija, saugojimo vieta – naudojami ekonomiškai.











































Efektyvaus sandėliavimo pavyzdys – paštu siunčiamų užsakytų prekių verslas. Norint efektyviai pasiekti prekes, jos dažnai nebetvarkomos pagal abėcėlę ar prekės numerį. Vietoj to jos sandėliuojamos pagal sudėtingus algoritmus, kuriuose atsižvelgiama, pavyzdžiui, į tai, kaip dažnai prekės perkamos.

35. Mėgstamiausias filmas

Grupė draugų nori kartu pažiūrėti filmą. Kiekvienas asmuo kiekvieną iš septynių filmų įvertina:

puikus , vidutiniškas  blogas .

Žiūrėkite lentelę:

	1	2	3	4	5	6	7
Ada							
Dorotėja							
Tomas							
Jonas							
Rasa							
Rožė							

Filmas laikomas mėgstamiausiu, jei kiekvienas žmogus jam skyrė savo geriausią įvertinimą. Pavyzdžiui, filmas Nr. 1 nėra mėgstamiausias, nes Tomas geriausiai įvertino filmą Nr. 4. Deja, dabar mėgstamiausio filmo nėra. Tad Ada nori įtikinti kuo mažiau draugų pakeisti savo vertinimą, kad mėgstamiausias filmas vis dėlto būtų.

Padėkite Adai ir kitiems pakeisti kuo mažiau įvertinimų, kad filmas būtų įvertintas kaip mėgstamiausias.

Paaiškinimas

2 yra teisingas atsakymas. Nėra galimybės pakeisti tik vieno filmo įvertinimą, nes apklausoje bent du draugai geriau įvertino kitą filmą:

Filmas	Tiek daug draugų kitus filmus įvertino geriau
1	4: Dorotėja, Tomas, Jonas, ir Rožė
2	3: Tomas, Rasa, ir Rožė
3	3: Tomas, Rasa, ir Rožė
4	3: Dorotėja, Rasa, ir Rožė
5	3: Dorotėja, Jonas, ir Rasa
6	2: Tomas ir Rožė
7	3: Tomas, Jonas, ir Rožė

Adai reikia įtikinti Tomą ir Rožę pakeisti po vieną įvertinimą, kad 6-as filmas taptų mėgstamiausiu. Tomas ir Rožė turi pagerinti savo 6-o filmo įvertinimus arba jie gali sumažinti savo vertinimus tiems filmams, kuriuos įvertino geriau nei 6-ą, t. y., 4 (Tomas) ir 5 (Rožė) filmams. Tačiau nesunku pastebėti, kad Ada turi atsižvelgti į reitingų pagerinimą tik panašiose būsimose situacijose.

Tai informatika!

Ką padarėte, kad išspręstumėte šią užduotį? Vienas iš būdų – patikrinti kiekvieną filmą po vieną ir kiekvieną asmenį po vieną, ar kuris nors iš kitų filmų turi geresnį įvertinimą. Galiausiai gausite tokią lentelę, kokia pateikta. Tuomet žinosite, kurių asmenų prašyti pakeisti savo vertinimus, kad būtų atlikta mažiausiai keitimų. Šį algoritmą Ada gali naudoti efektyviai spęsdama savo problemą.

Tačiau ar pats algoritmas yra veiksmingas, t. y., ar Ada galėtų veikti greičiau? Tarkime, yra M filmų ir F draugų. Ada turės peržiūrėti visus $M \times F$ lentelės įrašus ir kiekvienam iš jų turės peržiūrėti $M-1$ kitų to paties asmens įvertinimų. Iš viso ji turės peržiūrėti $M \times (M-1) \times F$ įvertinimų. Tačiau iš tikrųjų, norėdama sužinoti, ar asmens konkretaus filmo įvertinimas kelia problemų, Ada turi žinoti tik geriausią bendrą šio asmens įvertinimą. Jei konkretus

įvertinimas yra mažesnis už bendrą geriausią įvertinimą, įvertintas filmas negali būti mėgstamiausias. Tai reiškia, kad jei Ada pirmiausia sužino kiekvieno asmens bendrą geriausią įvertinimą (peržiūrėjusi visus $M \times F$ įvertinimus), ji gali patikrinti visus $M \times F$ įvertinimus, ar jie yra mažesni už atitinkamo asmens geriausią įvertinimą. Pagal šį alternatyvų algoritmą, kuriame iš anksto apskaičiuojami geriausi įvertinimai, Ana turi peržiūrėti $2 \times M \times F$ įvertinimų. Kai $M=7$ ir $F=6$, bus 84 peržiūros, o pagal ankstesnį algoritmą atliekamos 252 peržiūros. Antrasis algoritmas taip pat gerai sprendžia Ados problemą, tačiau jis yra efektyvesnis už pirmąjį.

Kompiuterių specialistų viena iš pagrindinių veiklų yra ne tik išspręsti problemą, bet ir atlikti tai kuo efektyviau. Greitesnė techninė įranga leidžia kompiuteriams greičiau spręsti problemas, tačiau jei nėra veiksmingo problemos sprendimo algoritmo, net ir greita techninė įranga nelabai padeda.

37. Nuliukai ir kryžiuikai

„Kryžiuikai ir nuliukai“ yra žaidimas, skirtas dviem žaidėjams, naudojamas popierius ir rašiklis.

Taisyklės. Du žaidėjai žaidžia paeiliui, pažymėdami po vieną 3×3 dydžio lentelės langelį nuliuku (0) arba kryžiuoku (X). Laimi tas, kuris pirmas trimis savo ženklais užpildo lentelės eilutę, stulpelį arba įstrižainę. Jei to nepavyksta padaryti, kol užpildoma visa lentelė – žaidimas baigiamas lygiosiomis.

Pavyzdys. Paveikslėliuose pavaizduoti keli pirmieji žaidimo ėjimai ir paskutinis žaidimo ėjimas (vėliausias ėjimas paryškintas):

		X

O		X

O		X
X		

O		X
	O	
X		

...

O		X
	O	O
X	X	X

Dešinėje esantis paveikslėlis vadinamas baigto žaidimo rezultatu. Ne visos lentelės, kuriose atsitiktine tvarka įrašyta X arba O, yra galiojantys rezultatų lapai pagal pirmiau pateiktas taisykles.

Kuriame iš šių paveikslėlių pavaizduotas vienintelis teisingas baigto žaidimo rezultatas pagal aprašytas taisykles?

A.

X	O	X
O	X	O
O	O	X

B.

X	O	X
O	X	
O	X	X

C.

X	X	O
	O	X
O	O	X

D.

X	O	X
O	X	O
O	X	

Paaiškinimas

X	X	O
	O	X
O	O	X

Teisingas atsakymas: , nes žaidimą laimėjo nuliukų žaidėjas.

Atsakymas A yra neteisingas. Kryžiuokų žaidėjas laimėjo žaidimą, bet nuliukų lentelėje yra daugiau, negu kryžiuokų, ko būti negali. Kadangi paskutinį ženklą visada padeda laimėjęs žaidėjas, jo ženklų gali būti tik daugiau arba tiek pat, kiek priešininko, bet ne mažiau.

Atsakymas B yra neteisingas, nes jame yra 5 kryžiuokai ir tik 3 nuliukai. Taip būti negali. Skirtumas tarp kryžiuokų ir nuliukų skaičiaus visuomet bus 0 arba 1.

Atsakymas D yra neteisingas, nes laimėtojas nebuvo išaiškintas, o lentelė neužpildyta.

Tai informatika!

Uždavinį išsprendėme tikrindami kiekvieną paveikslėlį, ar šis yra tikėtinas.

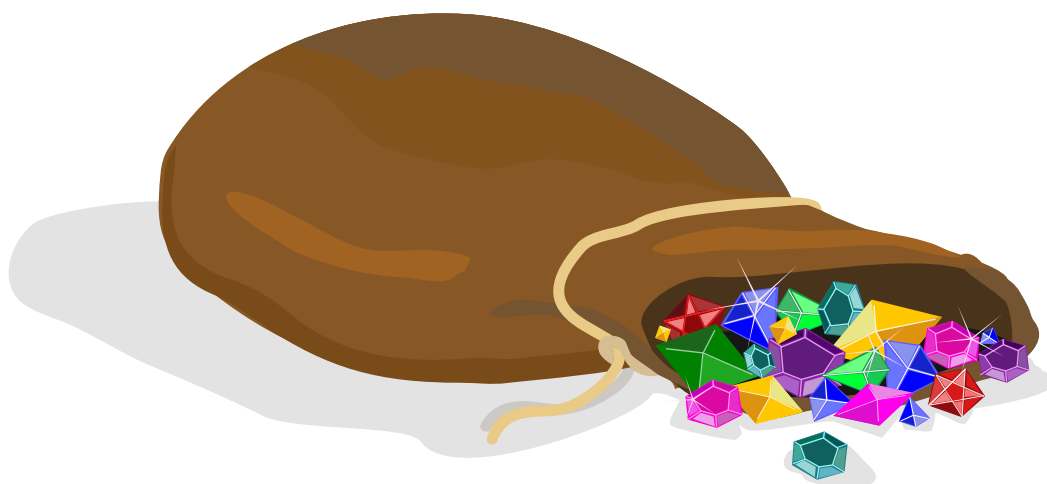
Pagal žaidimo taisykles galima nustatyti keletą taisyklių, kurias turi atitikti baigto žaidimo rezultatas, pavyzdžiui:

- 1) skirtumas tarp nuliukų ir kryžiuokų skaičiaus visada bus 0, 1 arba -1;
- 2) jei nė vienas žaidėjas nelaimėjo žaidimo, turi būti užpildyta visa lentelė;
- 3) laimėjusio žaidėjo ženklų lentelėje visuomet bus tiek pat arba vienu daugiau, negu pralaimėjusio žaidėjo ženklų;
- 4) vieno žaidimo rezultate gali būti tik viena laimėjimą reiškianti trijų vienodų ženklų seka.

Jei paveikslėlis neatitinka bent vienos iš šių taisyklių, jis negali būti teisingas (tikėtinas) žaidimo rezultatas.

Kai duomenys apdorojami kompiuteriu, taisyklės yra labai svarbios. Pavyzdžiui, yra taisyklės, aprašančios paveikslų failų formatus, mokėjimo kortelių numerius ar net telefono numerius. Kai bandote atverti paveikslo failą grafikos programa, ši pirmiausia patikrina, ar failo turinys atitinka nurodyto paveikslo formato taisykles.

37. Gražiausias brangakmenis



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

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

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

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

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

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

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

Paaiškinimas

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

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

Tai informatika!

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

Didžiausio brangakmenių skaičiaus radimo strategija turi tam tikrų apribojimų:

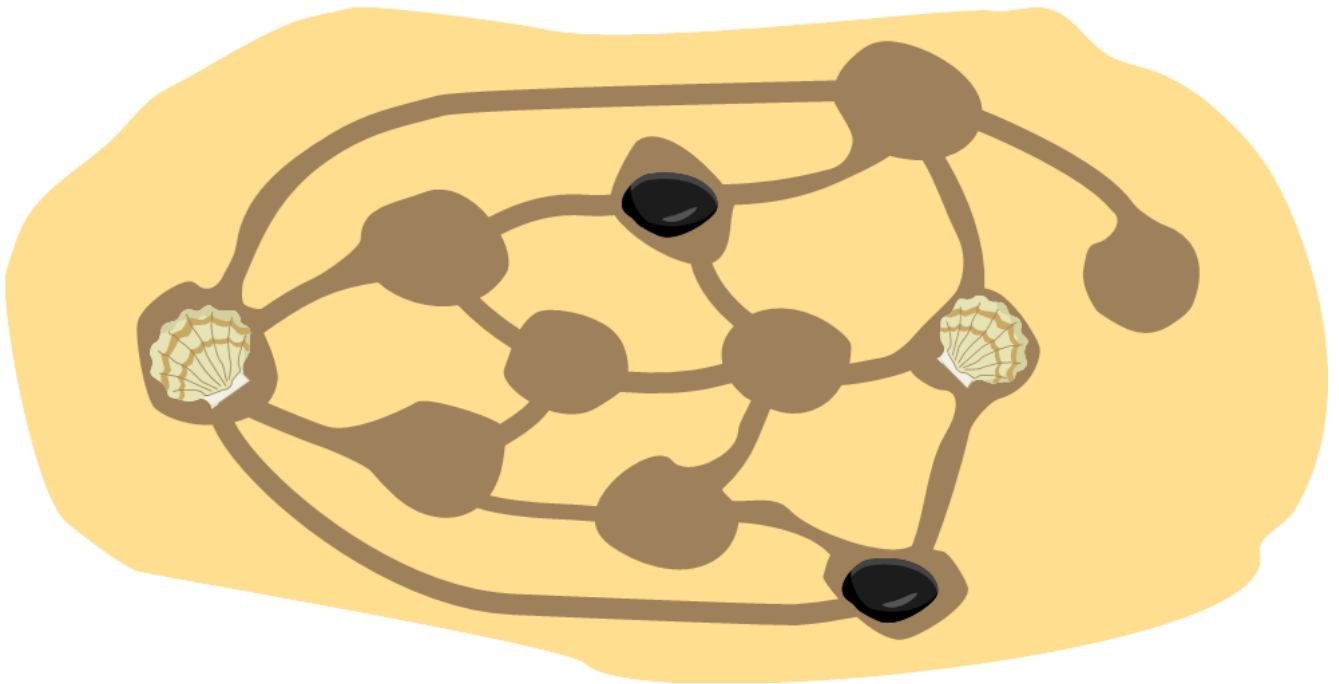
1. Nustatyti mėgstamiausią iš pasirinktų keturių brangakmenių.
2. Galima tik tris kartus nagrinėti pasirinktus brangakmenių rinkinius iš keturių elementų.

Yra daug algoritmų, kurie gerai veikia tik tam tikrais atvejais. Pavyzdžiui, rikiavimo algoritmas gali rikiuoti teigiamus sveikuosius skaičius, bet ne neigiamus sveikuosius skaičius. Kai kurie maršrutų paieškos algoritmai gali rasti trumpiausią maršrutą tarp dviejų objektų, tačiau tai daro per lėtai, kai naudojami žemėlapiai su tūkstančiais susijusių objektų. Tokiu atveju dažnai pasirenkami greitesni algoritmai, kurie randa vieną iš trumpiausių maršrutų, bet nebūtinai patį trumpiausią. Geras greitas sprendimas yra geriau nei jokio sprendimo! Apskritai yra daugybė būdų, kaip įvertinti algoritmus, kurie naudojami kompiuterinėse sistemose. Pavyzdžiui, paprastai vertinama programos (algoritmo) veikimo trukmė arba kiek kompiuterio atminties programa gali užimti, kol yra vykdoma užduotis.

38. Kriauklės ir akmenukai

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

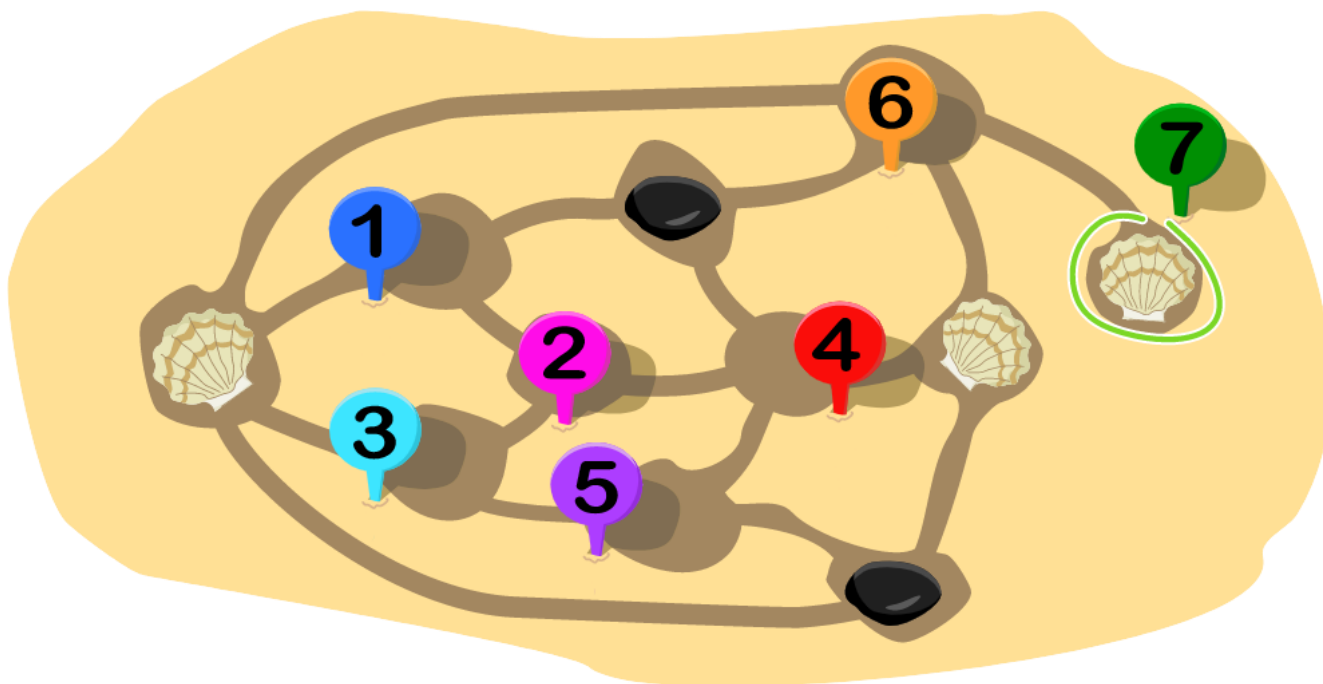
Pralaimi tas žaidėjas, kuris pirmas įdeda dvi savo figūrėles į dvi duobutes, sujungtas vienu grioveliu.



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

Paaiškinimas

Teisingas atsakymas yra 7.



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

Jei Ana būtų pasirinkusi 2-ą duobutę, tai Bobui lieka 7 duobutė. Toliau Anai tinka tik 5 duobutė, o Bobui tinka 3 duobutė ir Ana pralaimi.

Jei Ana būtų pasirinkusi 5-ą duobutę, o Bobas – 7, po to Ana – 2, Bobas – 3 ir Ana vėl pralaimi.

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

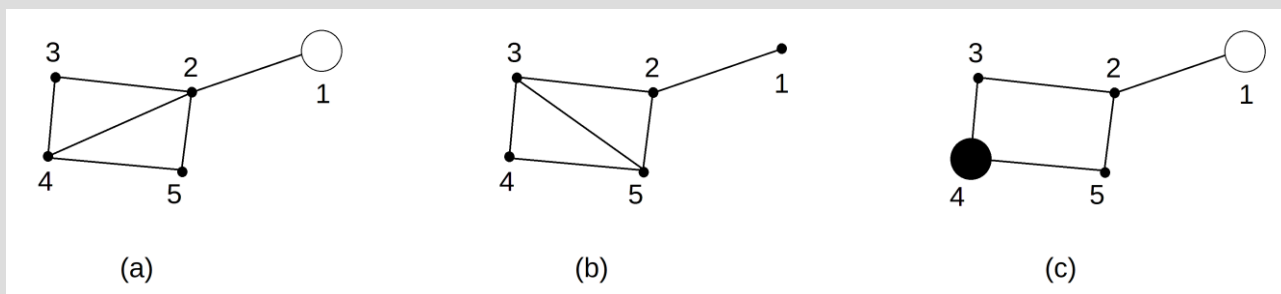
Tai informatika!

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

nuliukų žaidimas (angl. *tic-tac-toe*) visada baigiasi lygiosiomis, jei abu žaidėjai žaidžia geriausiai. Kalbant apie spalvinimo žaidimus *Col* ir *Snort* (neabejotinai baigtinius), rezultatas priklausys nuo žaidimo lentos, t. y., nuo pateikto grafo, kurio viršūnės yra duobutės, o briaunos – grioveliai (tarp skirtingų duobučių).

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

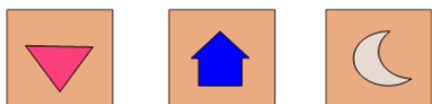
Pateiktame paveikslėlyje pavaizduotos trys galimos baigtys, naudojant tris paprastas skirtingas lentas: a) pirmasis žaidėjas (baltieji) turi laiminčią strategiją ir privalo pradėti 1 (kitaip pralaimės); b) antrasis žaidėjas (juodieji) turi laiminčią strategiją; c) partija lygiosiomis, jeigu baltieji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



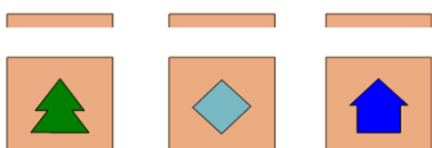
Col žaidimo tam tikrose situacijose, pavyzdžiui, simetrinėse pozicijose, yra laimėjimo strategijų.

39. Lobio dėžutė

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



1. Viena figūra yra teisinga ir tinkamoje vietoje.



2. Nei viena figūra į atrakinančią kombinaciją neįeina.



3. Dvi figūros teisingos, bet yra netinkamoje vietoje.

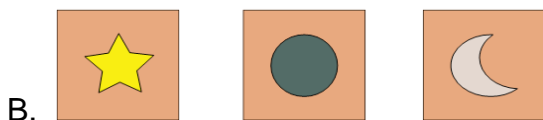


4. Viena figūra teisinga, bet netinkamoje vietoje.






5. Viena figūra teisinga, bet netinkamoje vietoje.


Kuri iš pateiktų kombinacijų atrakins lobio dėžutę?



Paaiškinimas

Teisingas atsakymas: B.

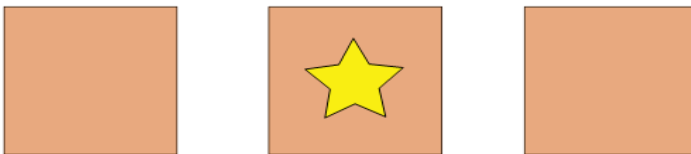
Pirmiausia eliminuosime figūras, kurios netinka kombinacijai, atrakinančiai dėžutę. Antroje užuominų rinkinio eilutėje matome, nei viena jos figūra į atrakinančią kombinaciją neįeina. Tai reiškia, kad eglutė , rombas  ir rodyklė  nepadės atrakinti dėžutės.

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

Galimos žvaigždutės pozicijos:



ARBA



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



ARBA



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



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

Pirmoje eilutėje turėjome

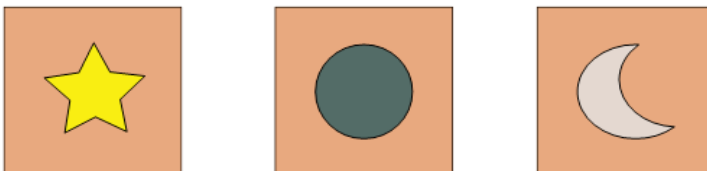


ARBA



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

Ketvirtoje eilutėje matome, kad teisinga viena figūra, tačiau ji yra neteisingoje vietoje. Trikampis ▼ jau yra atmestas, todėl vienintelė galima vieta yra viduryje. Širdelė ♥ negali būti tinkama figūra, kadangi ji yra viduryje. Taigi darome išvadą, kad viduryje turi būti skritulys ●.

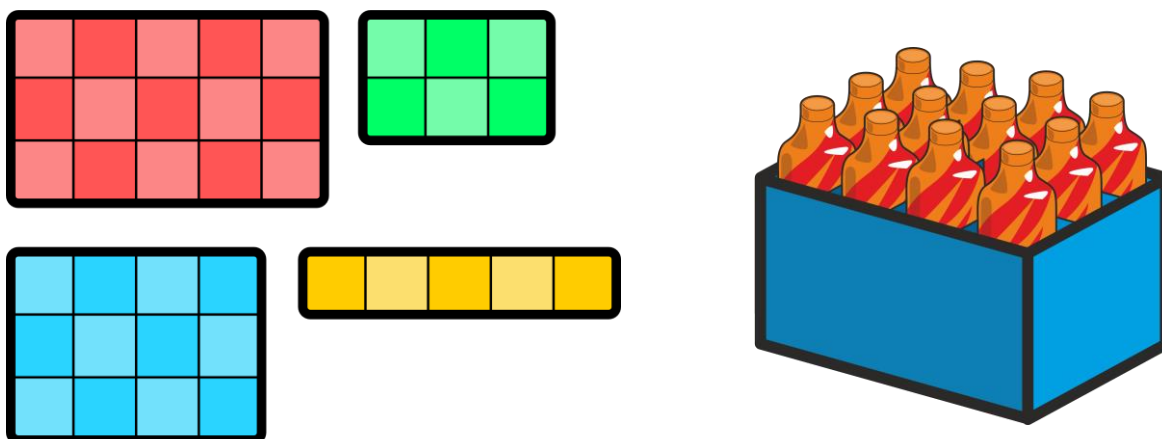


Tai informatika!

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

40. Dovana

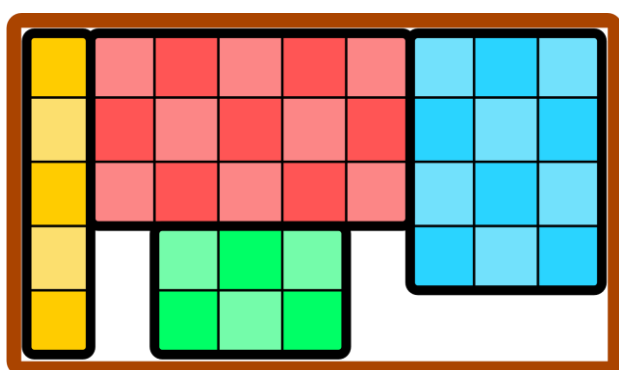
Įmonė gamina 4 rūšių sultis. Visų rūšių sultys pilstomos į vienodo dydžio butelius, bet kiekviena rūšis pakuojama į skirtingo dydžio dėžes. Paveikslėlyje šios dėžės vaizduojamos iš viršaus ir, pavyzdžiui, didžiausioje dėžėje telpa 15 butelių.



Dešinėje esančiame paveikslėlyje parodyta, kaip dėžė, kurioje telpa 12 butelių, atrodo iš šono.

Sporto varžyboms reikia paruošti specialią dovaną, kurioje būtų po vieną dėžę kiekvienos rūšies sulčių. Šios keturios dėžės turi būti supakuotos į stačiakampį konteinerį. Dėžės negali būti sukrautos viena ant kitos ir reikia palikti kuo mažiau tarpų, kurie bus užpildomi pavieniais sulčių buteliais.

Pavyzdys.



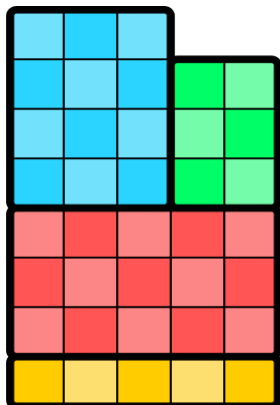
Jei būtų naudojamas šiame paveikslėlyje pavaizduotas konteineris, tai jį tektų papildyti 7 sulčių buteliais, kad konteineris būtų pilnas.

Kiek pavienių butelių reikia sudėti į stačiakampį konteinerį, kuriame su kuo mažesniais tarpais telpa keturios dovanų dėžės, kad šis konteineris būtų pilnas?

Paaiškinimas

Teisingas atsakymas: 2.

Paveiksle pateikiamas galimas dėžių išdėstymas.



Visose 4 dėžėse kartu sudėjus butelių skaičius yra $12 + 15 + 6 + 5 = 38$. Talpyklos, kurioje telpa 38 buteliai su 0 tarpų, matmenys turi būti 1×38 arba 2×19 . Į tokį konteinerį niekada netilps 3×5 dėžė (arba 3×4 dėžė).

Į konteinerį su 1 tarpų tilptų 39 buteliai. Yra dvi galimybės: 1×39 (neįmanoma) arba 3×13 . Dvi didžiosios dėžės tokiam konteineriui užimtų 9 eiles. Likusių 4 eilių nepakanka mažiausiai 1×5 dydžio dėžei įdėti.

Vadinasi, 2 tarpai yra mažiausias galimas skaičius konteineriui.

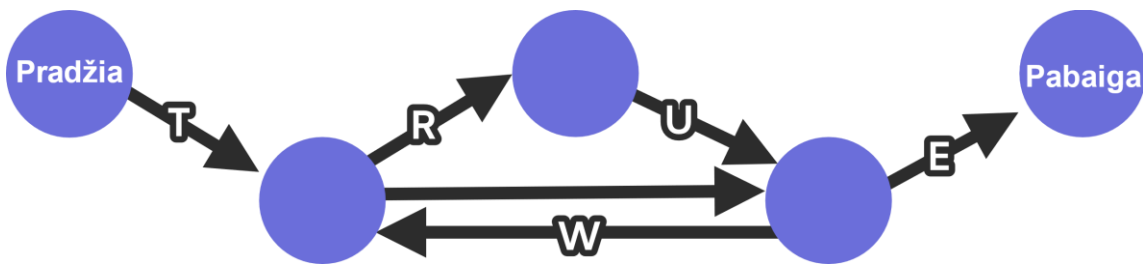
Tai informatika!

Ši užduotis yra konteinerių pakavimo uždavinio pavyzdys. Pakavimo uždaviniai aktualūs realiame gyvenime. Pavyzdžiui, svarbu pakrauti kuo daugiau dėžių į sunkvežimį arba sužymėti automobilių stovėjimo aikšteles, kad jose galėtų stovėti kuo daugiau automobilių.

Tai optimizavimo uždavinio pavyzdys, kai norime rasti algoritmą, kuris greitai pateiktų geriausią sprendimą. Deja, gali būti labai sudėtinga rasti geriausią pakavimo sprendimą, kai dėžių skaičius yra didelis, net jei naudojame labai galingus kompiuterius. Vietoj to galime naudoti algoritmus, kurie pateikia tik apytikslus sprendinius, bet yra pakankamai geri praktinėse situacijose.

41. Slaptažodžiai

Brigita kuria slaptažodžius naudodamasi paveiksle pavaizduota schema. Ji visada pradeda nuo skritulio su užrašu Pradžia ir keliauja rodyklėmis, kol atsiduria skritulyje su užrašu Pabaiga. Kiekvieną raidę, kurią ji mato ant rodyklės, užsirašo.



Pavyzdžiui, naudojantis šia schema, galima sukurti slaptažodžius TWWWE ar TRUE.

Kiek skirtingų slaptažodžių iš 8 raidžių gali sukurti Brigita, naudodama šią schemą?

- A. 8
- B. 10
- C. 12
- D. 14

Paaiškinimas

Kiekvienas slaptažodis turi prasidėti raide T, o paskutinė raidė visada bus E.

Reikia pagal schemą išnagrinėti galimas 6 raidžių kombinacijas iš šių raidžių: R, U ir W.

Panagrinėkime galimus slaptažodžius, kai viduryje yra nuo vienos iki šešių raidžių:

- jei 1 raidė, tai 1 galimybė (W)
- jei 2 raidės, tai 2 galimybės (WW ir RU)
- jei 3 raidės, tai 3 galimybės (WWW; WRU; RUW)
- jei 4 raidės, tai 4 galimybės (WWWW; WWRU; RUWW; WRUW)
- jei 5 raidės, tai 6 galimybės (WWWWW; WWWRU; WWRUW; WRUWW; RUWWW; RUWRU)
- jei 6 raidės, tai 8 galimybės (WWWWWW, WWWWRU, WWWRUW, WWRUWW, WRUWWW, RUWWW, WRUWRU, RUWRUW)

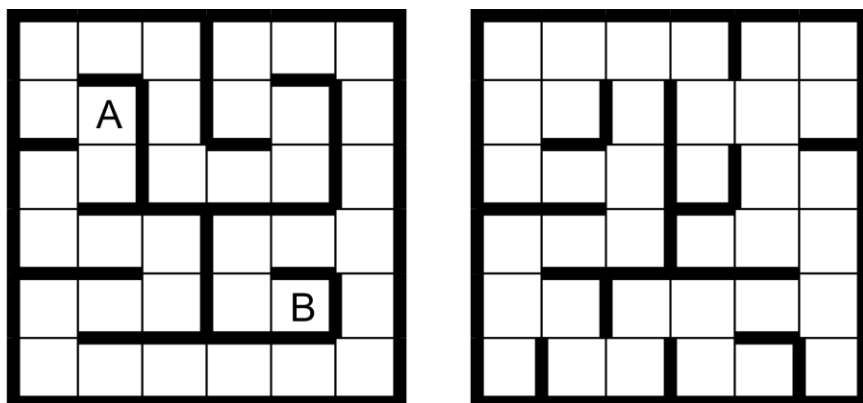
Tai informatika!

Kompiuteris iš tiesų puikiai geba surasti visas įmanomas galimybes.

Tai vadinamoji algoritminė mašina, kuri generuoja raidžių seką su tam tikrais apribojimais. Tokios sistemos kartais naudojamos generuoti slaptažodžius mokiniams, dalyvaujantiems „Bebro“ konkurse, ar gaminant automobilių numerius. Tokiais atvejais reikalingos unikalios simbolių sekos, tačiau jos turi atitikti nustatytas sąlygas.

42. Labirintas

Magijos mokykla yra dviejų aukštų pastate. Abu aukštai yra tiksliai vienas ant kito, jie suskirstyti į kvadratus, tarp kai kurių kvadratų yra sienos:



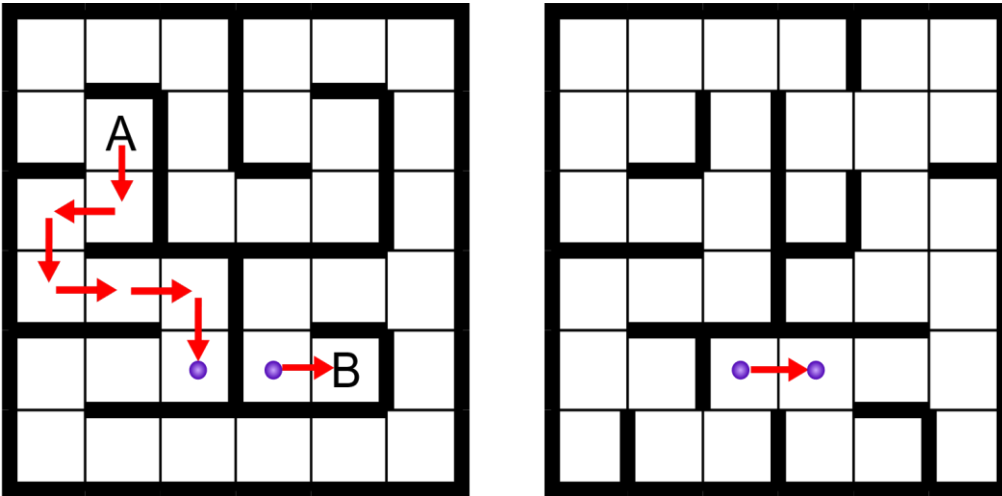
Magijos mokinys per 1 sekundę pereina iš vieno kvadrato į kitą tame pačiame aukšte. Deja, Robertas pamiršo, kaip vaikščioti kiaurai sienas. Tačiau jis gali pereiti iš vieno aukšto į atitinkamą kvadratą kitame aukšte – tam reikia 5 sekundžių.

Robertas nori kuo greičiau patekti iš A kvadrato į B kvadratą.

Kiek sekundžių reikia Robertui, kad tai padarytų?

Paaiškinimas

Teisingas atsakymas: 18. Robertas gali nuvykti iš taško A į tašką B per 18 sekundžių:



Ar tai greičiausias būdas? Panagrinėkime, kaip galima apskaičiuoti trumpiausią laiką, per kurį Robertas gali nuvykti iš kvadrato A į bet kurį kitą kvadratą. Akivaizdu, kad trumpiausias laikas kvadrato A yra 0 sekundžių. Toliau žingsnis po žingsnio einama taip: iš visų kvadratų, kurių trumpiausias laikas jau yra įvestas, pasirenkamas tas, kurio vertė yra mažiausia. Taigi pačioje pradžioje pasirenkate kvadratą A. Iš šio pasirinkto kvadrato peržvelgiame visus kitus galimus kvadratus ir svarstome, kaip iš pasirinkto kvadrato į juos patekti greičiausiai; kituose kvadratuose įvedate apskaičiuotus laikus. Gali atsitikti taip, kad anksčiau įvestas laikas bus pagerintas. Pasirinktas kvadratas vėliau negali būti svarstomas; jo negalima pasirinkti kituose žingsniuose. Čia pateikiami trumpiausi šiuo metodu apskaičiuoti laikai, pradedant nuo A kvadrato:

2	3	4	11	12	13
1	0	5	10	9	14
2	1	6	7	8	15
3	4	5	18	17	16
8	7	6	17	18	15
9	10	11	12	13	14

7	6	7	8	11	12
6	5	8	9	10	11
7	6	7	10	11	12
8	9	8	13	12	13
9	10	11	12	13	14
10	11	12	13	14	15

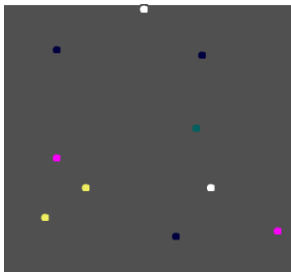
Matome, kad greičiausias kelias iš A į B yra 18.

Tai informatika!

Greičiausius arba trumpiausius kelius tenka skaičiuoti gana dažnai; akivaizdus pavyzdys – maršruto planavimas šiuolaikinėse žemėlapių programėlėse. Problema labai supaprastėja, jei kelią sudaro atskiri žingsniai tarp gretimų taškų ir žinoma, kiek visi šie žingsniai „kainuoja“ (kaina vadiname laiką, pinigus, energijos sąnaudas – bet kokį dydį, kuris yra svarbus sprendžiant konkretų uždavinį). Šiuo atveju taškus, žingsnius ir žingsnių kainą galima abstrahuoti į grafą, kuriame žingsnius galima sujungti į kelius. Informatikos moksle žinoma daug algoritmų grafams, kuriais galima efektyviai apskaičiuoti trumpiausius kelius. Vieną iš jų sugalvojo informatikas Edsgeris Dijkstra (Edsger Dijkstra); šis Dijkstros algoritmas buvo panaudotas pirmiau, aiškinant teisingą atsakymą. Trumpiausi keliai taip pat yra svarbūs projektuojant kompiuterių grandines. Perjungimo taškai turi būti sujungti laidais kuo mažesnėmis sąnaudomis. Šiuolaikinės grandinės sudaro keli lygiai, o laidų jungimas tarp dviejų lygių yra brangesnis nei to paties lygio laidų jungimas – panašiai kaip pereiti tarp aukštų šioje „Bebro“ užduotyje užtrunka ilgiau, nei žingsnis tame pačiame aukšte.

43. Grindų raštas

Kvadratiniame kambaryje ant grindų padėta dešimt 5 skirtingų spalvų mikroschemų.



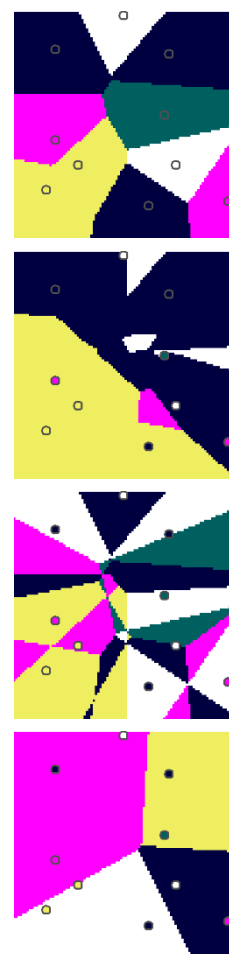
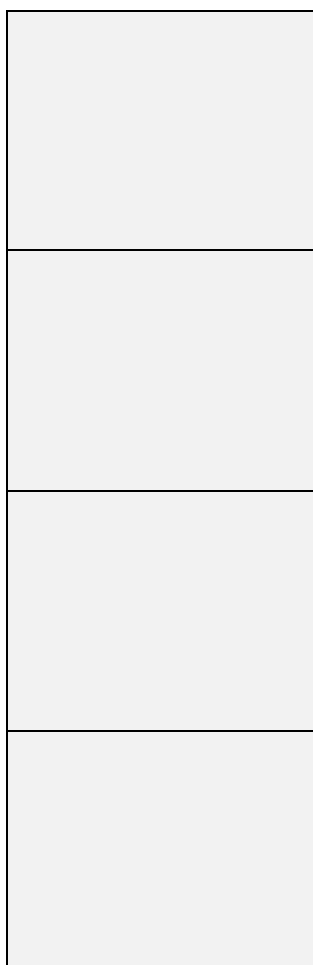
Tada grindys (išskyrus mikroschemas) nudažomos taip, kad dažų spalva kiekviename grindų plotelyje būtų nustatoma pagal specialią dažymo taisyklę. Ta pati taisyklė taikoma visoms grindims. Jei pagal taisyklę galima parinkti daugiau nei vieną vienodai gerą spalvą, galima pasirinkti bet kurią iš jų. Suderinkite pateiktas keturias skirtingas dažymo taisykles su atitinkamais grindų dažymo rezultatais. Vilkite grindų paveikslėlius į tuščias vietas šalia taisyklių. Kiekvienas grindų plotelis nudažytas:

1 taisyklė:
arčiausiai esančios
mikroschemos spalva.

2 taisyklė:
toliausiai nuo jo esančios
mikroschemos spalva.

3 taisyklė:
antros arčiausiai esančios
mikroschemos spalva.

4 taisyklė:
spalva, kuri labiausiai
paplitusi tarp 6 artimiausių
mikroschemų.



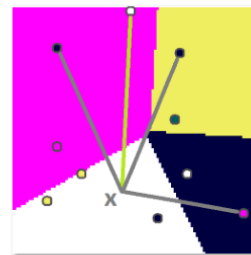
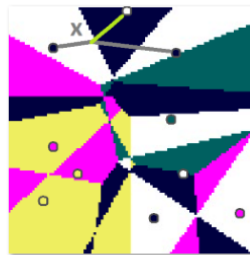
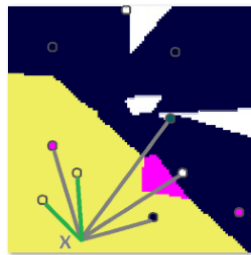
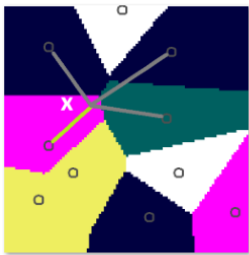
Paaiškinimas

1-A

2-D

3-C

4-B



<p>Plotelis x nuspalvintas rožine spalva, nes rožinis taškas yra arčiausiai taško x.</p>	<p>Plotelis x nuspalvintas geltonai, nes iš šešių kaimyninių plotelių geltona spalva yra dažniausia.</p>	<p>Plotelis x nuspalvintas balta spalva, nes baltas taškas yra antras arčiausiai taško x.</p>	<p>Plotelis x nuspalvintas balta spalva, nes baltas taškas yra labiausiai nutolęs nuo taško x.</p>
------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------

Tai informatika!

Plokštumos dalijimas ir jo realizavimas naudojantis algoritmais svarbus įvairiose kompiuterių mokslo srityse, pavyzdžiui, kompiuterinėje grafikoje ir modeliavime.

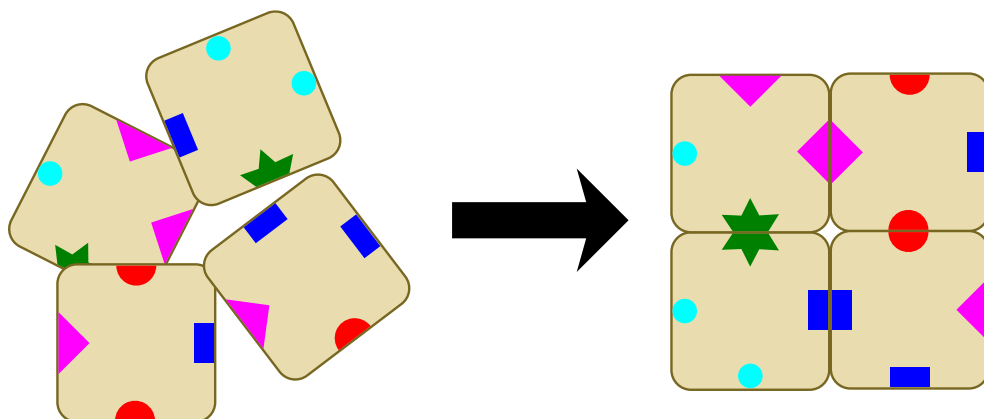
Voronojaus diagramos (A atvejo rezultatas) yra ypač įdomios ir dažnai pasitaiko realiose situacijose, pavyzdžiui, mobiliojo ryšio aprėpties srityje.

44. Keturios kortelės

Keturias korteles reikia sudėlioti į 2×2 kvadratą laikantis šios taisyklės:

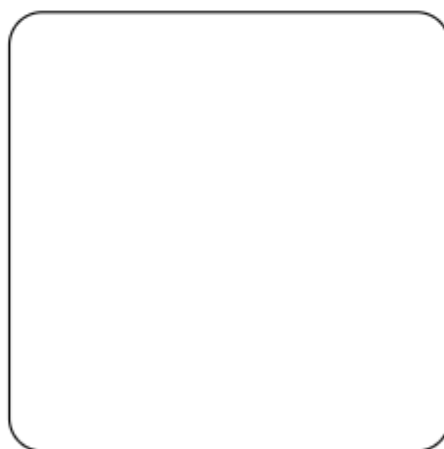
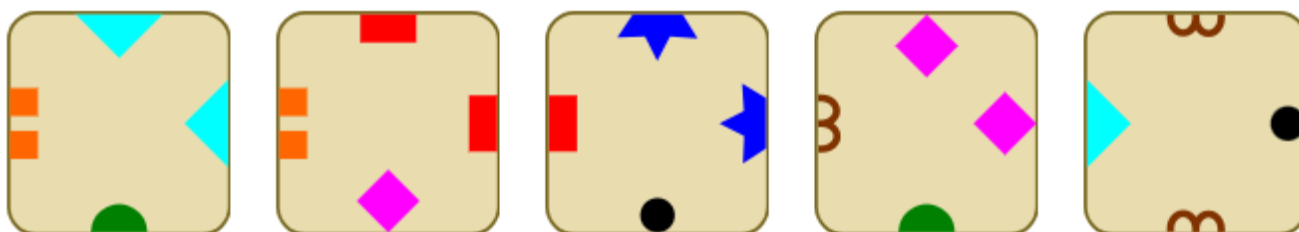
Kortelės gali liestis tik tada, jei jų šonai sutampa.

Pateikiame pavyzdį:



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

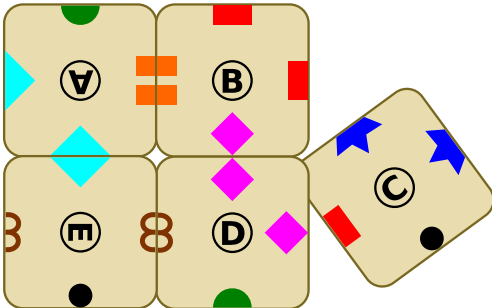
Spustelkite ant kortelių, kad jas pasuktumėte, ir nuvilkite jas į kvadratą.



Paiškinimas

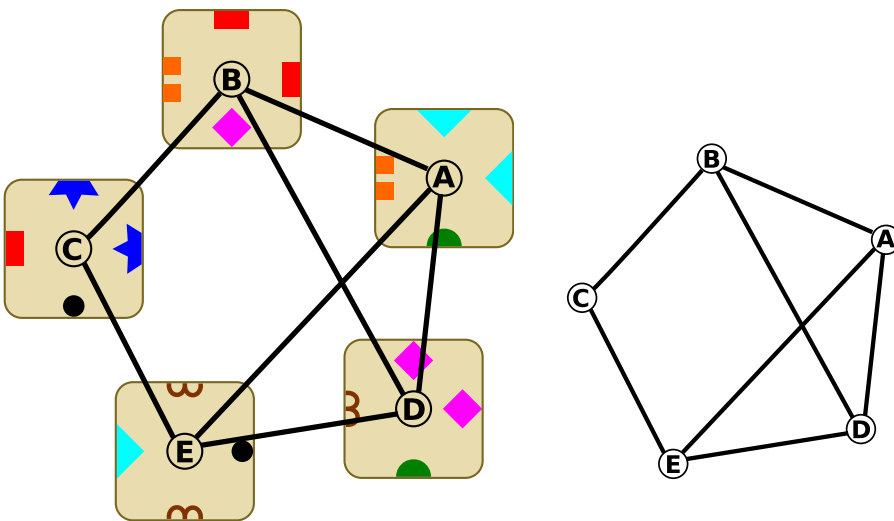
Teisingas atsakymas yra C.

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



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

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



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

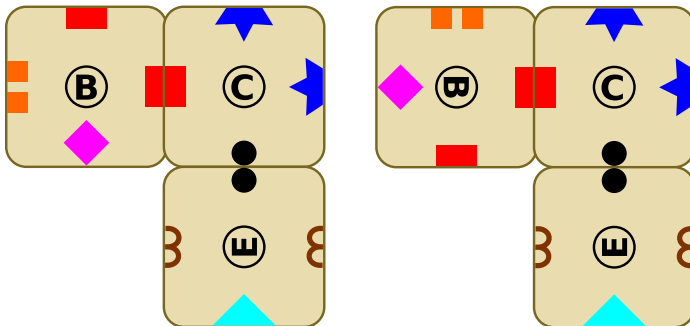
2. Naudodamiesi schema (grafu), surandame visus galimus 4-žingsnius ciklus. Čia yra trys 4-žingsniai ciklai:

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

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

c) trečiasis yra B-C-E-D-B.

Žinoma, visi cikliniai penkių raidžių perstatiniai (pradedant nuo bet kurios raidės, bet išlaikant eilės tvarką) yra ekvivalentūs, įskaitant ir atvirkštinius. Pavyzdžiui, A-B-C-E-A, B-C-E-A-B ir E-C-B-A-E yra ekvivalentūs. Ar gali būti, kad antrasis arba trečiasis 4-žingsniai ciklai gali padėti rasti kitą mūsų galvosūkių sprendimą? Abiejuose cikluose yra B-C-E ir yra tik dvi galimybės sujungti šias korteles tokia tvarka:



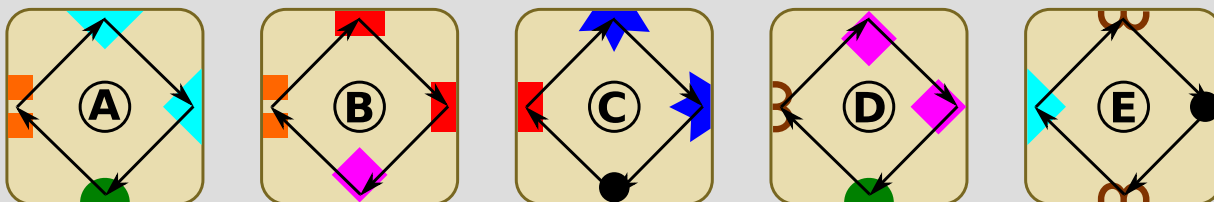
Nei A, nei D kortelės negali būti panaudotos 2x2 kvadratui užbaigti (ir, žinoma, atitikti taisyklę).

Tai informatika!

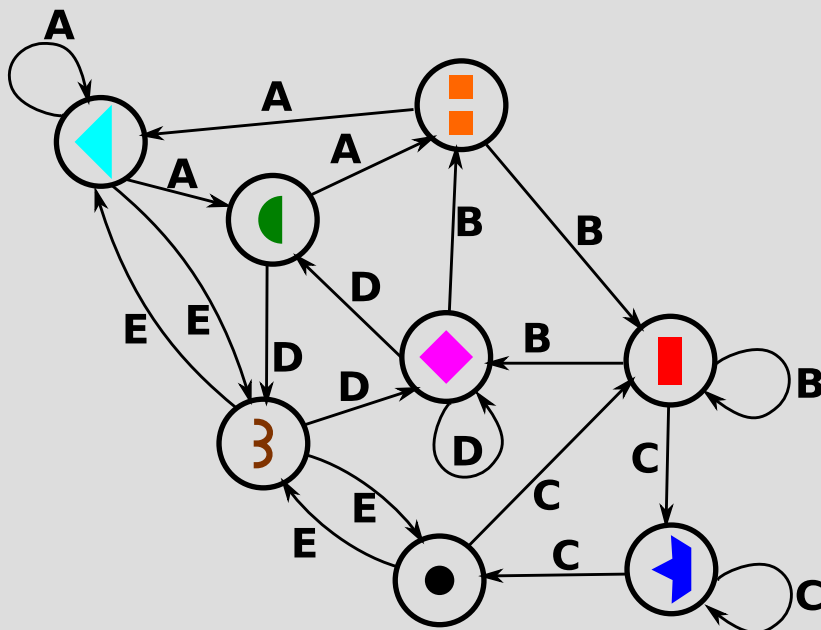
Sprendžiant sudėtingą uždavinį dažnai naudinga (laikina) nekreipti dėmesio į kai kurias detales. (Tai vadinama abstrakcija ir yra svarbus informatinio mąstymo komponentas). Pirmajame žingsnyje nekreipėme dėmesio į simbolių formą, spalvas ir jų padėtį ant kortelių šonų. Domino tik tai, ar dvi kortelės turi bendrą simbolį, ar ne. Gauta diagrama (grafas) suteikė mums pakankamai informacijos, kad sprendimo paieška būtų šiek tiek trumpesnė. Liko tik trys galimi sprendimai. Grafai dažnai naudojami informatikoje kaip apdorojamų duomenų abstrakcija. Istorijos bėgyje daugybė žmonių sugalvojo gudrių ir greitų metodų, kaip nustatyti įvairias tokių grafų savybes, pavyzdžiui, lengvai rasti visus 4-žingsnius ciklus. Norint toliau spręsti šį uždavinį reikėjo antrojo žingsnio, kuriame vėl buvo naudojamas grafas. Sudėtingo uždavinio skaidymas į atskirus žingsnius, kuriuos galima lengviau išspręsti, vadinamas dekompozicija – tai vėl svarbus informatinio mąstymo komponentas.

Alternatyvus sprendimas

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



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











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

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





45. Magiška šalis

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

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

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


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

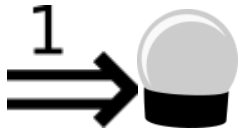
- A. 
- B. 
- C. 
- D. 

Paaiškinimas

Teisingas B atsakymas.

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

Atsakymas A: 

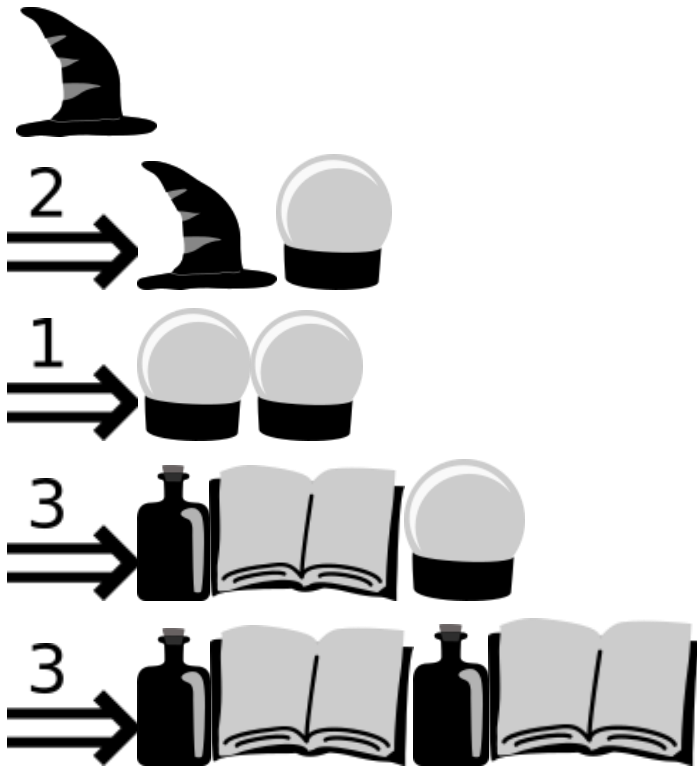


Atsakymas C:





Atsakymas D:



Tai informatika!

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

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

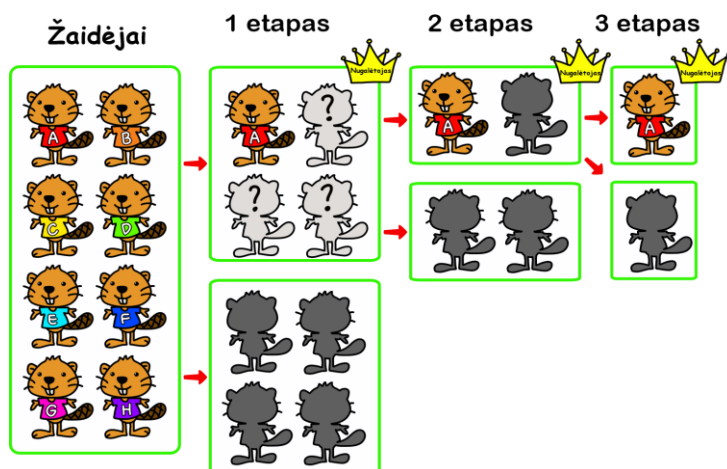
46. Bebrų žaidynės

Bebrų čempionate dalyvauja aštuoni bebrai. Varžybos organizuojamos trimis etapais. Kiekviename etape kiekvienas bebras renka taškus.

- 1 etape burtų keliu sudaromos dvi komandos po keturis bebrus. Susumuojami atskirų bebrų taškai. Daugiausia taškų surinkusi komanda laimi ir patenka į antrąjį etapą. Pralaimėjęsijei toliau žaidžia ir tarpusavyje išsiaiškina 5–8 vietas.
- Antrasis etapas vyksta pagal tas pačias taisykles. Dabar komandose yra po 2 bebrus. Nugalėtojai patenka į finalą. Pralaimėjęsijei žaidžia toliau ir tarpusavyje išsiaiškina garbingas vietas.
- 3 etape, finale, varžosi ne komandos, o du bebrai vienas prieš kitą.

Bebrų čempionato nugalėtoja tapo bebrė Ada . Kiekvieno bebro kiekviename etape surinkti taškai pateikti lentelėje.

Vardas	Ada	Bronius	Cilė	Daina	Edis	Fredis	Gedas	Herkus
1 etapas	15	16	19	18	17	20	19	19
2 etapas	20	27	30	24	28	24	30	30
3 etapas	10	14	11	15	16	13	9	12



Kurie trys bebrai buvo Ados pirmojoje komandoje? Nutempkite juos ant paveiksliukų, pažymėtų klausukais.

Paaiškinimas

Kiti trys Ados komandos nariai yra Daina , Fredis  ir Gedas .

Trečiajame etape žaidėjai dalyvauja individualiai. Gedas yra vienintelis žaidėjas, gavęs mažiau taškų už Adą, todėl jie 3 etape žaidė finale, o 2 etapo metu turėjo būti vienoje komandoje.

Judviejų taškų suma antrajame etape buvo 50. Taigi, kitos dviejų žaidėjų komandos taškų suma turėjo būti mažesnė. Daina ir Fredis yra vieninteliai kiti du žaidėjai, kurie kartu surinko mažiau nei 50 taškų. Todėl pirmojo etapo metu jie turėjo būti toje pačioje komandoje, kaip Ada ir Gedas. Kadangi mes jau žinome šios komandos narius, mums nebereikia tikrinti kitos komandos narių.

Taigi, pirmajame etape komanda, kurią sudarė Ada, Daina, Fredis ir Gedas, surinko 72 taškus, o komanda, sudaryta iš Broniaus, Cilės, Edžio ir Herkaus, surinko 71 tašką. Taigi, Ados komanda laimėjo. Antrajame etape Ada ir Gedas kartu surinko 50 taškų, o Daina ir Fredis surinko 48 taškus. Trečiajame etape Ada su 10 taškų laimėjo prieš Gedą, surinkusį 9 taškus. Taigi, Ada yra žaidynių nugalėtoja.

Tai informatika!

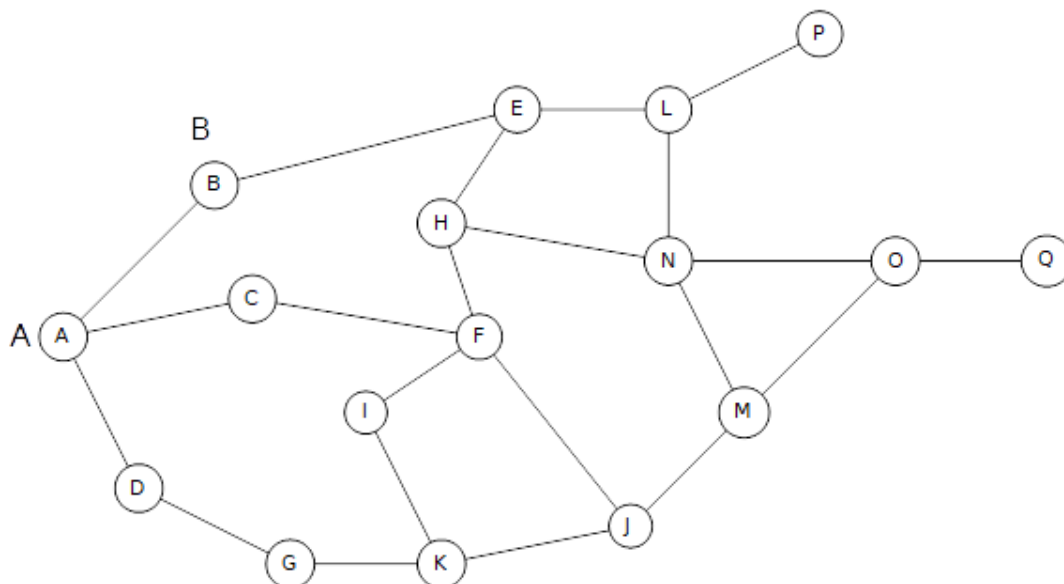
Bandydami išspręsti šią užduotį ir pradėję nuo pirmojo etapo, turėtume patikrinti visus galimus derinius ir įvertinti kiekvieno iš jų rezultatus visuose etapuose. Tai užimtų labai daug laiko.

Tokiais atvejais informatikai ieško efektyvesnių metodų uždaviniui išspręsti. Mūsų atveju užuot pradėję nagrinėti nuo pradžių, galime daryti išvadas pradėdami nuo pabaigos, ir tai labai greitai duos teisingą sprendimą, kuris ir paaiškintas.

Šis metodas vadinamas atvirkštine paieška (angl. *backward search*) ir taikomas tada, kai ieškomas sprendinys tenkina kokius nors ribojimus. Kartais tiesioginė paieška ir atvirkštinė paieška sprendžiant uždavinį yra derinamos kartu.

47. Virusai

Schemoje apskritimai žymi kompiuterius, o linijos – ryšius tarp jų.



A kompiuteris užkrėstas „StuxNet“ virusu, o B kompiuteris – „RoXX3“ virusu.

Kiekvienos dienos pradžioje abu virusai patenka į visus gretimus, tiesiogiai su užkrėstu kompiuteriu sujungtus, kompiuterius. Jei abu virusai atsiduria tame pačiame kompiuteryje, tai tas kompiuteris visiškai nustoja veikęs, tad virusai iš jo nebeplinta. Po keleto dienų kiekvienas kompiuteris jau yra arba užkrėstas, arba sugedęs.

Kiek yra veikiančių, bet užkrėstų „RoXX3“ virusu kompiuterių?

- (A) 6 (B) 7 (C) 8 (D) 9 (E) 10

Paaiškinimas

1 diena. B, C ir D kompiuteriai užkrėsti „StuxNet“ virusu, o A ir E kompiuteriai – „RoXX3“ virusu, sugadinami A ir B kompiuteriai.

2 diena. F ir G kompiuteriai užkrėsti „StuxNet“ virusus (iš C ir D kompiuterių), o H ir L (iš E kompiuterio) virusu „RoXX3“. Kompiuteriai tebeveikia.

3 diena. „StuxNet“: F ir G užkrečia H, I, J, K kompiuterius (C jau yra užkrėstas).

„RoXX3“: L užkrečia N ir P (E jau yra užkrėstas), H užkrečia F (E, N).

Taip nustoja veikę H ir F kompiuteriai.

4 diena. „StuxNet“: (I, J, K) užkrečia (F, K), (F, K) M, ir (G, I, J) – vienintelis naujai infekuotas kompiuteris – M.

„RoXX3“: N užkrečia O, M (H jau sugadintas, ir L jau turi virusą. P taip pat gali užkrėsti.)

Taip sugriaunamas M kompiuteris.

5 diena: „StuxNet“ neturi kur plisti. Vienintelis naujai infekuotas kompiuteris nustojo veikęs 4 dieną.

„RoXX3“: O gali užkrėsti Q (ir N, kuris jau užkrėstas).

6 diena: „RoXX3“: N gali užkrėsti jau sugriuvusius kompiuterius H ir M arba jau užkrėstą kompiuterį L.

Kadangi nei „StuxNet“, nei „RoXX3“ negali užkrėsti naujų kompiuterių, virusu „RoXX3“ užkrėsti kompiuteriai yra: B, A, E, H, L, N, F, P, O, M, Q (išvardyti apytiksle užkrėtimo tvarka), taigi 11 kompiuterių turi šį virusą. Iš jų A, B, F, H, M nustojo veikę (5). Lieka 6 kompiuteriai (atsakymas A), užkrėsti „RoXX3“ virusu, bet veikiantys (E, L, P, N, O, Q).

Tai informatika!

Šia užduotimi parodoma, kaip grafai gali būti taikomi nusakant ryšius tarp objektų. Grafas – tai labai dažnai informatikoje naudojama duomenų struktūra ryšiams nurodyti. Šioje užduotyje kompiuteriai – tai grafo viršūnės, o ryšio linijos – briaunos. Grafikai taip pat padeda lengviau įsivaizduoti užduotį, palyginus su tekste pateiktais ryšių aprašymais. Šiame uždavinyje viršūnės pašalinamos taip sumažinant briaunų (galimų jungčių) kiekį.

48. Rikiavimas

Jorūnas turi 11 kortelių su skaičiais nuo 1 iki 9 ir raidėmis A, B. Jis naudoja taisyklę $9 < A < B$. Jorūnas nori korteles sudėti į vieną eilutę, vadovaudamasis $a \rightarrow b$ tipo taisykle. Ši taisyklė reiškia, kad kortelė su numeriu a turi būti kažkur kairėje nuo kortelės su numeriu b. Kartu jis nori, kad kortelių išdėstymas sudarytų kuo mažesnę leksikografinę seką. Leksikografiškai mažesnė iš dviejų sekų yra ta, kurios pirmoje (iš kairės) skirtingų kortų poroje yra mažesnis skaičius. Panagrinėkime šias dvi sekas:

1) 5 6 5 7 9 9 B

2) 5 6 5 8 0 0 0

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

Išdėstykite 11 kortelių leksikografiškai mažiausia seka pagal šias taisykles:

1) $1 \rightarrow 2$

2) $1 \rightarrow 6$

3) $2 \rightarrow 3$

4) $6 \rightarrow 5$

5) $3 \rightarrow 5$

6) $3 \rightarrow 4$

7) $5 \rightarrow 7$

8) $7 \rightarrow 8$

9) $7 \rightarrow 9$

10) $1 \rightarrow B$

11) $B \rightarrow A$

12) $A \rightarrow 7$

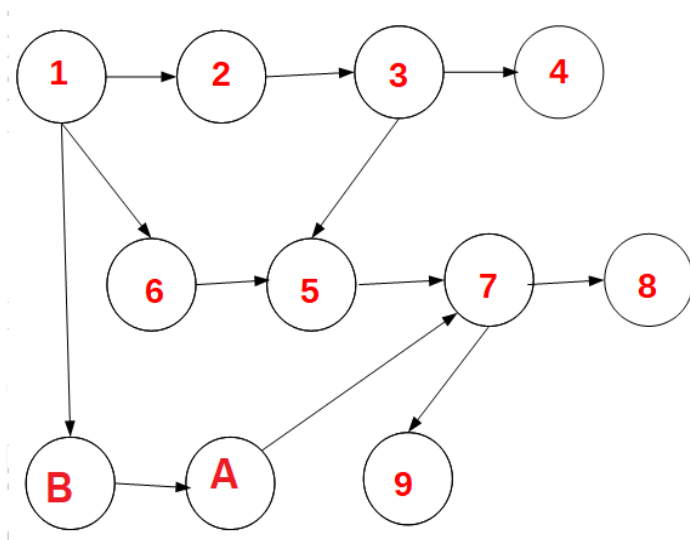
1 2 3 4 5 6 7 8 9 A B

□ □ □ □ □ □ □ □ □ □ □

Paaiškinimas

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

1 6 2 3 4 5 B A 7 8 9 taip pat tenkina visas nurodytas sąlygas, tačiau gaunama seka leksikografiškai didesnė. Šią užduotį suformuluokime grafų teorijos terminais. Kiekvieną kortelę laikysime orientuoto grafo viršūne, o kiekvieną sąlygą – orientuota briauna, nukreipta iš kairės į dešinę kortelę (žr. paveikslą):



Sudarykite į kiekvieną viršūnę įeinančių briaunų sąrašus:

1: -

2: 1

3: 2

4: 3

5: 3, 6

6: 1

7: 5, A

8: 7

9: 7

A: B

B: 1

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

Leksikografiškai mažiausia seka bus sudaryta taip:

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

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

Pirmiausia pasirinkite viršūnę 1.

Dabartinė paieškos seka: [1, ...]

Atnaujinti įvesties briaunų sąrašai:

1: -

2: -

3: 2

4: 3

5: 3, 6

6: -

7: 5, A

8: 7

9: 7

A: B

B: -

Raudonai pažymėkite viršūnes, kurios jau įtrauktos į norimą seką.

Dabar 2, 6, 11 viršūnės neturi įeinančių briaunų.

Pasirinkite 2 viršūnę (mažiausias skaičius).

Dabartinė paieškos seka: [1, 2, ...]

Atnaujinti įvesties briaunų sąrašai:

1: -

2: -

3: -

4: 3

5: 3, 6

6: -

7: 5, A

8: 7

9: 7

A: B

B: -

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

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

Atnaujinti įvesties briaunų sąrašai:

1: -

2: -

3: -

4: -

5: -

6: -

7: 5, A

8: 7

9: 7

A: B

B: -

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

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

Atnaujinti įvesties briaunų sąrašai:

1: -

2: -

3: -

4: -

5: -

6: -

7: -

8: 7

9: 7

10: -

11: -

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

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

Tai informatika!

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

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

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

Todėl toks uždavinys daug efektyviau sprendžiamas topologinio rikiavimo metodu.