

Bebr[a]s

'22

1. kārtas uzdevumi

11.-12. klasei

INFORMATĪVAIS
ATBALSTĪTĀJS

start(it)

www.startit.lv

SATURS

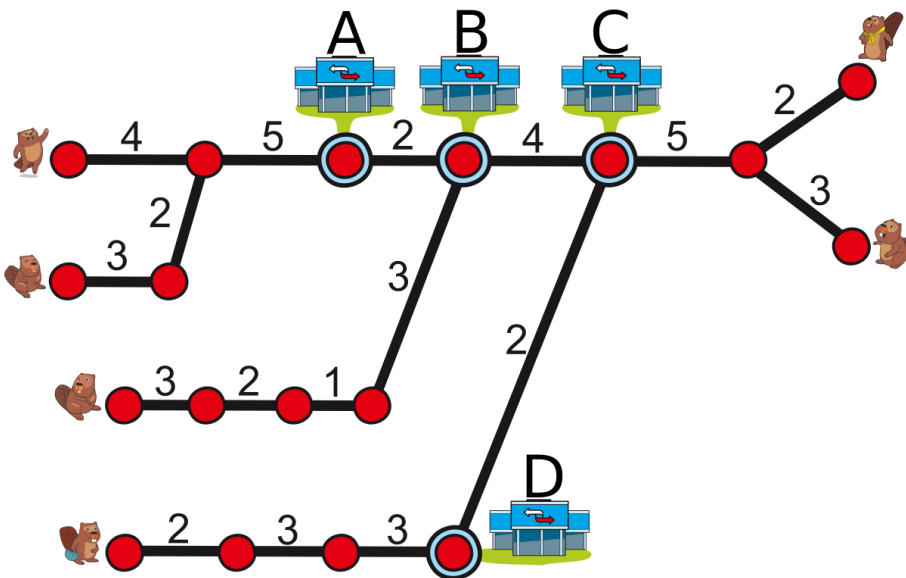
Pazemes dzelzceļa tīkls	2
Pērļu kaklarotas	4
Salu savienošana	7
Alkatīgie troļļi	9
Mistērija	12
Elektromobiļu rindas	15
Bebru datubāze	17
Izkrāso vardi!	19
Klausies un ej!	22
Vārīgie krustojumi	25
Labirints	27
Vārdi	29
Bebru spēles	31
Robota maršruts	33
Akmeņu savākšana	35

Pazemes dzelzceļa tīkls

Saūda Arābija

Seši bebri dzīvo dažādās Bebrpilsētas daļās. Viņi izmanto pazemes dzelzceļu, lai satiktos kādā no tā galvenajām stacijām - A, B, C vai D. Viņi vēlas ierasties vienā un tajā pašā stacijā iespējami īsākā laikā. Visi bebri savu braucienu sāk no atšķirīgām stacijām (skat. attēlu) vienā un tajā pašā laikā.

Kartē parādītas katra dzelzceļa atzara stacijas (sarkanie aplīši) un laiks minūtēs, kāds nepieciešams braucienam starp divām blakus stacijām. Pasažieru iekāpšanai un izkāpšanai katrā stacijā nepieciešama viena minūte.



Uzdevums

Kurā no stacijām visi bebri var satīties visagrāk?

Atbilžu varianti

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

Atbildes izskaidrojums

Pareizā atbilde: Stacijā B

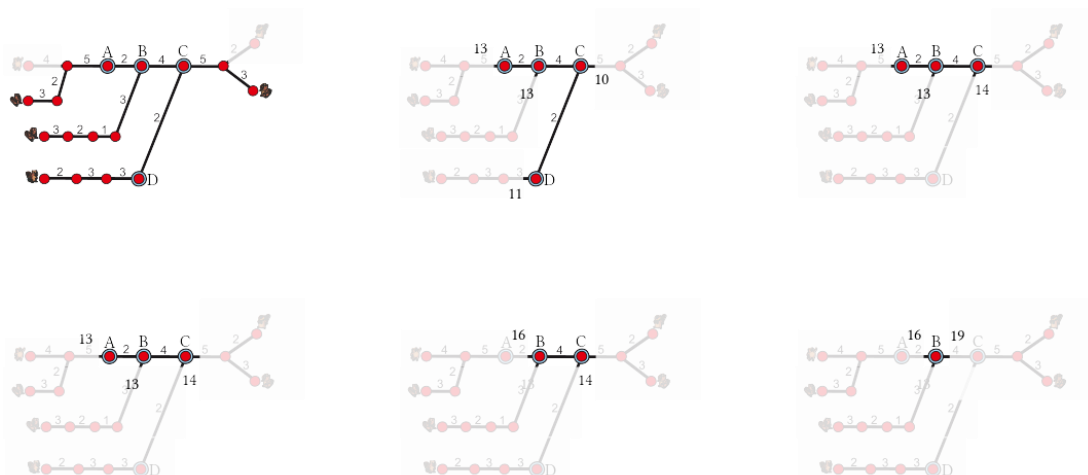
Lai atrisinātu uzdevumu, nepieciešams aprēķināt katra bebra brauciena laiku minūtēs līdz katrai no stacijām (A, B, C, D).

Protams, var rēķināt šo laiku katram bebram līdz katrai no stacijām, veicot $4 \cdot 6 = 24$ ceļa garumu aprēķinus. Tomēr šādi aprēķini būs nogurdinoši un aizņems pārāk daudz laika.

Tāpēc labāk izmantosim dekompozīciju (uzdevuma sadalīšanu mazākos) un abstrakciju (izlemt, kuri atzari ir aprēķināšanai svarīgāki, un ignorēt pārējos). Aplūkosim tikai ilgākos

brauciena laikus, jo svarīgs ir tikai tā bebra brauciena ilgums, kurš tikšanās vietā ieradīsies pēdējais.

Zīmējumā zemāk ir parādīti iespējamie uzdevuma atrisināšanas soļi.



Veicot aprēķinus par diviem bebrim, kuri stacijā A ieradīsies no kreisās puses, mēs iegūsim, ka pirmajam bebram būs nepieciešamas 11 minūtes ($1+4+1+5$), bet otrajam – 13 minūtes ($1+3+1+2+1+5$). Tātad turpmāk varam aplūkot tikai otro bebru, jo pirmais jebkurā no pārējām stacijām ieradīsies agrāk par otro. Līdzīgi, divi bebru no labās puses līdz stacijai C brauks 10 minūtes ($1+3+1+5$), bet, tā kā trešajam bebram (attēlā apakšējais) nepieciešamas 14 ($1+11+2$) minūtes, tad turpmāk varam ņemt vērā tikai šīs 14 minūtes.

Tālāk varam aprēķināt tikšanās laiku stacijā (B) un iegūsim mazāko iespējamo laiku no braucienā sākuma – 19 minūtes.

Lai apstiprinātu šo rezultātu, tabulā parādīsim ilgākā brauciena laiku līdz katrai stacijai. Tas apliecina, ka labākā tikšanās vieta ir stacija B.

Brauciens līdz A	Brauciens līdz B	Brauciens līdz C	Brauciens līdz D
No kreisās: 13	No kreisās: 16	No kreisās: 21	No kreisās: 24
No labās: 22	No labās: 19	No labās: 10	No labās: 11

Tā ir informātika!

Datorzinātnē dinamiskā programmēšana nozīmē sarežģītas problēmas vienkāršošanu, sadalot to vienkāršākās apakšproblēmās. To plaši izmanto optimizācijas problēmās (kaut kāda lieluma maksimuma vai minimuma atrašanai).

Šāda veida diagrammas, ko izmanto šajā uzdevumā, sauc par grafu. Stacijas ir virsotnes, un ceļi ir šķautnes. Konkrētāk, šis ir svarota grafa piemērs, jo šķautnēm ir piešķirtas noteiktas skaitliskas vērtības (ceļojuma ilgums). Tas nozīmē, ka grafam ir šķautnes ar svāriem. Svāri grafos var attēlot tādas lietas kā izmaksa, garums, svārs vai ietilpība.

Pērļu kaklarotas

Maķedonija

Savā brīvdienų ceļojumā Ohridā Monika un Veronika nopirka Ohridas pērles un izveidoja no tām attēlā redzamās kaklarotas.

Monika	
Veronika	

Viņas nolēma samainīt pērles, veicot šādus soļus:

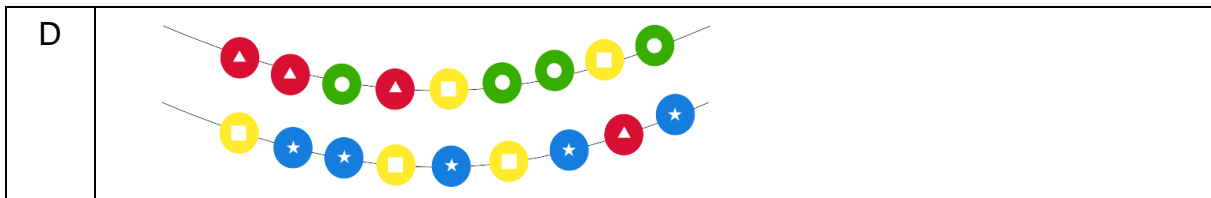
1. solis. Monika un Veronika katra paņem vienu pērli no savas kaklarotas labā gala.
2. solis. Ja pērle ir dzeltena (kvadrāta simbols) vai sarkana (trīsstūra simbols), tad to pievieno otras meitenes kaklarotas kreisajā galā. Ja pērle ir kādā no citām krāsām, tad to pievieno savas kaklarotas kreisajā galā. Vispirms šo soli izpilda Monika, un tad – Veronika.
3. solis. Ja katra no meitenēm ir atdevusi otrai trīs pērles, viņas apstājas. Pretējā gadījumā maiņu turpina, atkal sākot ar 1. soli.

Uzdevums

Kā izskatīsies Monikas un Veronikas kaklarotas pēc aprakstīto soļu izpildes?

Atbilžu varianti

A	
B	
C	



Atbildes izskaidrojums

Pareizā atbilde: A

Atbilde B nav pareiza, jo Monikai un Veronikai vajadzēja apmainīties tikai ar trim pērlēm, bet šajā variantā samainītas ir četras. Atbilde C nav pareiza, jo soļi nav izpildīti līdz galam. Atbilde D nav pareiza, jo, lai gan dažādo krāsu pērļu skaits ir pareizs, tās atrodas nepareizā secībā. Lai pārliecinātos, ka atbilde A ir pareiza, varam izpildīt aprakstītos soļus, sākot ar sākuma stāvokli:

Monika	
Veronika	

Mēs veiksīm darbības, lai pārmaiņus, sākot ar Moniku, noņemtu pērles pa vienai no katras kaklarotas labā gala. Tālāk paskaidrojumu tekstā izmantosim terminu "iterācija", lai apzīmētu vienu gājienu – 1. līdz 3. soļa izpildi. Attēlā zemāk parādīts katras iterācijas rezultāts. Šajā attēlā pirmais krāsaino rindu pāris attēlo kaklarotu sākuma stāvokli, bet nākamie – kaklarotas pēc katras iterācijas.

1. iterācija.

Tā kā abas pērles labajā pusē ir dzeltenas vai sarkanas, tās tiek apmainītas un pievienotas otrai kaklarotai kreisajā pusē. Katra meitene līdz šim ir iedevusi otrai vienu pērli.

2. iterācija.

Tā kā Monikas kaklarotas labajā pusē ir dzeltena pērle, viņa to pievieno Veronikas kaklarotas kreisajā pusē. Veronikai ir zila pērle (kura netiks apmainīta), tāpēc tā tiek pievienota viņas pašas kaklarotas kreisajā pusē. Monika iedevusi divas un Veronika – vienu pērli.

3. iterācija.

Nākamā Monikas pērle ir zaļa, tāpēc viņa to pievieno savas kaklarotas kreisajā pusē. Nākamā Veronikas pērle ir sarkana, tāpēc viņa to pievieno Monikas kaklarotas kreisajā pusē. Abas meitenes ir iedevušas otrai divas pērles.

4. iterācija.


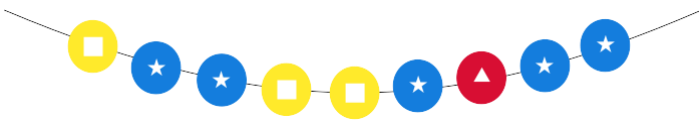
Nākamā Monikas pārle ir zaļa, tāpēc viņa to pievieno savas kaklarotas kreisajā pusē.

Nākamā Veronikas pārle ir zila, tāpēc viņa to pievieno savas kaklarotas kreisajā pusē. Bez izmaiņām: abas meitenes ir iedevušas otrai divas pārles.

5. iterācija.

Abas nākamās pārles ir dzeltenas vai sarkanas, tāpēc tās tiek apmainītas un pievienotas otras kaklarotas kreisajā pusē. Abas meitenes ir iedevušas otrai trīs pārles.

Tā kā viņas abas ir iedevušas otrai trīs pārles, maiņa beidzas, un kaklarotas izskatās kā parādīts attēlā zemāk, kas ir atbildes variants A.

Monika	
Veronika	

Tā ir informātika!

Mēs varam izmantot šo uzdevumu, lai izskaidrotu datorzinātņu jēdzienu "rinda". Rinda ir saraksts, kurā elementi tiek pievienoti tikai vienā galā un izņemti tikai no otra gala. To var saukt par "caurules" tipa datu struktūru, kurā realizēts "pirmais iekšā - pirmais ārā" jeb FIFO (first in – first out) princips. Tā ir noderīga datu struktūra, jo tā automātiski seko secībai, kādā vienumi tika pievienoti rindai.

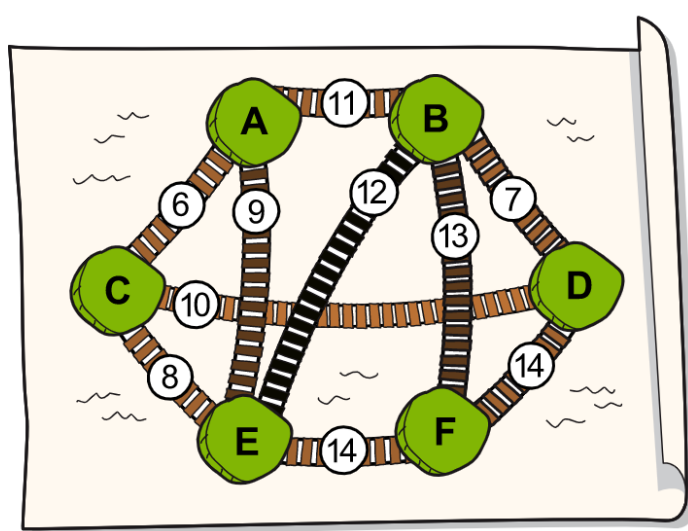
Monikas un Veronikas kaklarotas var uzskatīt par rindām. Lai gan mēs nepaskaidrojam, kā pārles tika savērtas kaklarotā, noteikumos ir paskaidrots, ka to gali būtiski atšķiras – vienā (kreisajā) galā var tikai pievienot pārles, bet otrs (labais) gals ir paredzēts tikai pārļu noņemšanai.

Salu savienošana

Lietuva

Džungļu kopiena, kas apdzīvo sešas salas, vēlas savienot šīs salas, izveidojot tiltu tīklu. Tiltu iespējamā savienojuma plāns parādīts zīmējumā. Tilti nekrustojas viens ar otru. Skaitļi parāda tiltu būvniecības izmaksas kāda salu pāra savienošanai.

Kopiena vēlas savienot visas salas tā, lai no jebkuras salas uz jebkuru citu salu varētu nokļūt vai nu tieši, vai netieši caur vienu vai vairākām salām. Tajā pašā laikā kopiena vēlas uzbūvēt tiltu tīklu pēc iespējas lētāk.



Uzdevums

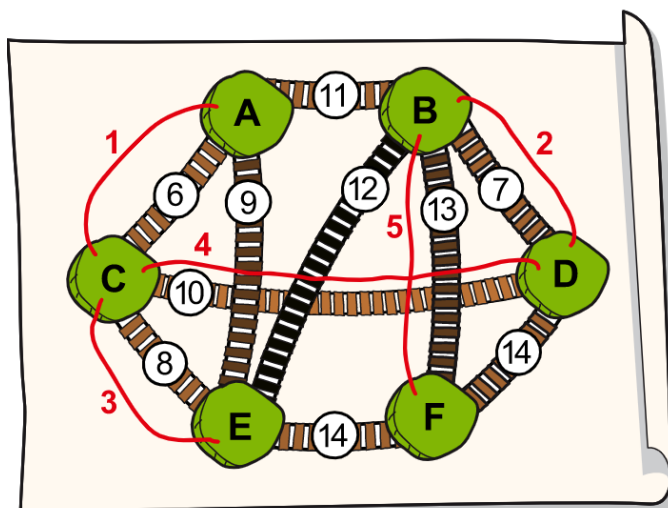
Cik izmaksās vislētākā tiltu tīkla uzbūvēšana?

Atbilžu varianti

Jāievada naturāls skaitlis.

Atbildes izskaidrojums

Pareizā atbilde ir: $6 + 7 + 8 + 10 + 13 = 44$



Algoritms ir vienkāršs un intuitīvs (to sauc par Kruskala algoritmu):

1. Sāciet no lētākā tilta AC – tas maksās 6.
2. Tad izvēlamies otru lētāko tiltu – BD (7).
3. Trešais lētākais tilts ir CE (8).
4. Nākamais lētākais tilts maksā 9. Tomēr, ja mēs to uzbūvēsim, mēs iegūsim ciklu AECA. Tas nozīmē, ka šis tilts mums nav nepieciešams – mēs varam sasniegt salas A, C un E, nebūvējot tiltu AE.
5. Ceturtais tilts, kas jābūvē, ir CD (10).
6. Nākamais lētākais tilts ir AB (11), atkal ir cikls: ABDCA
7. Līdzīgā veidā tilts BE (12) veido ciklu: ACDBEA
8. Piektais tilts, kas mums jābūvē, ir BF (13).

Tagad mums ir savienotas visas sešas salas, un tas ir lētākais veids, kā uzbūvēt piecus savienojuma tiltus: katrā posmā esam izvēlējušies lētāko iespējamo tiltu no vēl neuzbūvētajiem.

Tā ir informātika!




Šis ir minimālā pārklājošā koka atrašanas noteiktam grafam uzdevums. Pārklājošais koks ir noteikta grafa apakšgrafs, kuram ir tieši tādas pašas virsotnes un dažas šī grafa šķautnes.





Zināmais Kruskala algoritms vienmēr atrod minimālo pārklājošo koku sakarīgam svarotam grafam. Algoritms veic šķautņu ar mazāko svaru secīgu atlasīšanu, izvairoties no cikliem, ko veido jau izvēlētas šķautnes.

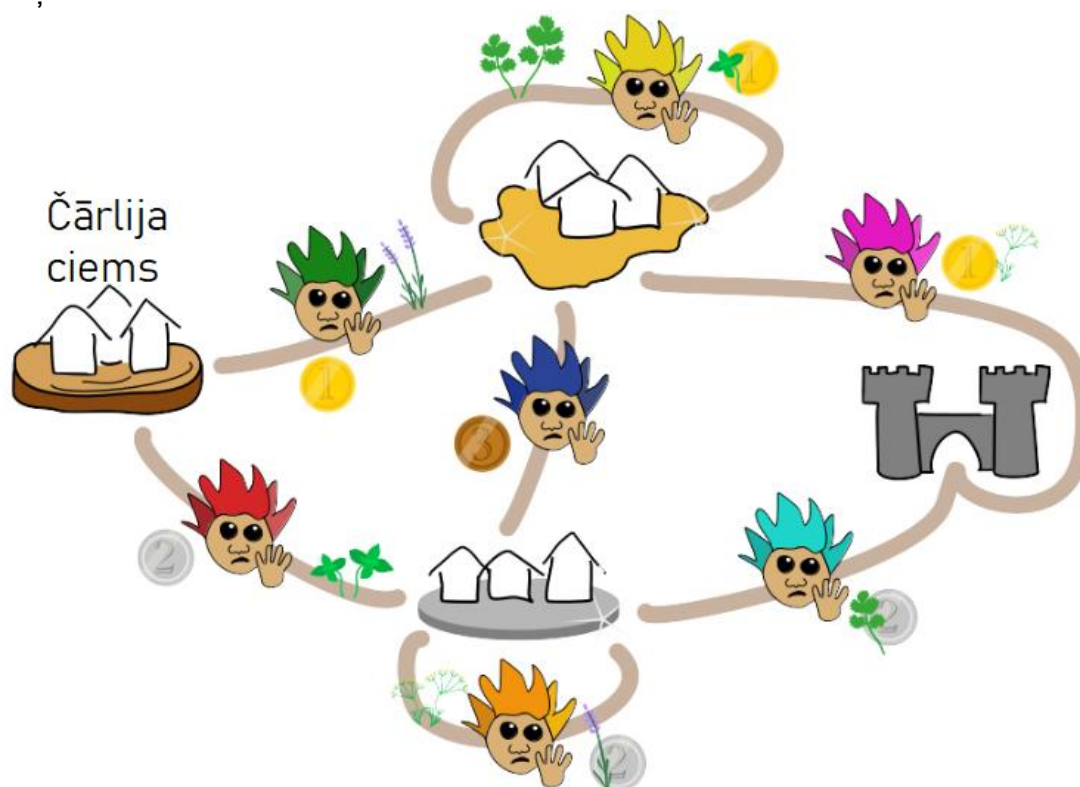
Alkatīgie troļļi

Šveice

Čārlijs vēlas no sava ciema nokļūt pilī, bet uz meža ceļiem, kas atrodas starp ciemu un pili, dežūrē alkatīgie troļļi. Katrs no tiem no jebkura ceļinieka kā nodevu pieprasa noteiktu

monētu ( ,  vai ). Daži no troļļiem papildus pieprasa arī kādu noteiktu augu

( ,  ,  vai ). Par laimi, šos augus var atrast meža ceļu ceļmalās. Zīmējumā parādīts, kāda monēta un kāds augs (ja attiecīgais troļlis to pieprasa) nepieciešami, lai varētu veikt attiecīgo ceļa posmu. Ja uz ceļa ir atrodams augs, tad to var savākt tikai pēc šī ceļa nodevas nomaksas.



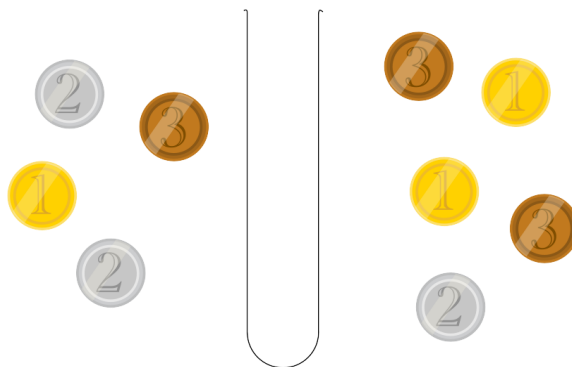
Čārlijam ir caurulīte ar vienu slēgtu galu, kurā viņš glabā monētas un uzpilda to pirms ceļojuma. Tā kā ceļojuma laikā viņš monētām varēs piekļūt secīgi (katrā brīdī tikai virsējai monētai), tad caurulīte ar monētām jāuzpilda pareizā secībā.

Čārlijam līdz būs arī neliela mugursoma, kurā noglabāt savāktos augus. Tā kā mugursoma ir gauži šaura, viņš piekļūt vienmēr varēs tikai vienam - virspusē esošajam augam. Čārlijs ceļojumu sāks ar tukšu mugursomu.

Uzdevums

Čārlijam caurulīte jāpiepilda ar piecām monētām pareizā secībā tā, lai viņš varētu pēc kārtas samaksāt visiem troļļiem savā ceļā līdz pilij. Neaizmirsti par augiem, kurus nepieciešams savākt!

Cik pareizas monētu kombinācijas iespējamas?



Atbilžu varianti

Ierakstiet veselu skaitli

Atbildes izskaidrojums

Pareizā atbilde: 2

Uzdevumam ir iespējami divi atrisinājumi:

	<p>Čārlijs vispirms dodas uz Zelta ciematu. Pa ceļam viņš samaksā vienu zelta monētu un savāc lavandu. Tad viņš dodas uz Sudraba ciematu un pa ceļam samaksā vienu vara monētu. Tad viņš dodas pa ceļu, kur jāsamaksā viena sudraba monēta un lavanda, bet viņš var ievākt dilles. Pēc tam viņš atkal dodas uz Zelta ciematu, samaksājot vienu vara monētu. No turienes viņš var doties uz pili, jo viņam ir zelta monēta un dilles.</p>	
	<p>Čārlijs vispirms dodas uz Sudraba ciematu. Pa ceļam viņš samaksā vienu sudraba monētu un savāc baziliku. Tad viņš dodas uz Zelta ciematu un pa ceļam samaksā vienu vara monētu. Pēc tam viņš dodas pa ceļu, kur viņam jāsamaksā viena zelta monēta un baziliks, bet viņš var savākt koriandru. Pēc tam viņš atkal dodas uz Sudraba ciematu, samaksājot vienu vara monētu. No turienes viņš var doties uz pili, jo viņam ir sudraba monēta un koriandrs.</p>	

Tie ir vienīgie atrisinājumi, jo, lai nokļūtu pilī, viņam vajag vai nu dilles, vai koriandru. Lai iegūtu dilles, viņam vispirms jāsavāc lavanda. Ar piecu monētu ierobežojumu paliek tikai 1. risinājums. Lai iegūtu koriandru, vispirms jāsavāc baziliks. Un tad paliek tikai 2. risinājums. Jebkurš cits veids viņam izmaksātu vairāk nekā piecas monētas.

Tā ir informātika!

Ciemos, pili un celiņus starp tiem var uzlūkot kā grafu. Katram ceļam ir piesaistīta noteikta cena (monēta), un tā veikšanas laikā var kaut ko ielikt vai paņemt no mugursomas, kas darbojas kā kaudze.

Steks ir datu struktūra, kas atbilst principam “pēdējais iekšā – pirmais ārā”, un kas atbalsta šādas darbības: ievietot objektu steka augšpusē (push) un izņemt steka augšpusē esošo objektu (pop).








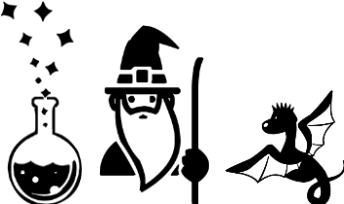
Tāpēc šo grafu var uzskatīt par magazīnas automātu jeb automātu ar steku. Magazīnas automāts sastāv no dažādiem stāvokļiem (šeit tos attēlo ciemati un pils), ievades alfabēta (šeit – monētas), kaudzes alfabēta (šeit – augi), pārejas funkciju kopas (šeit attēlotas kā ceļi), sākuma stāvoklis (šeit – Čārlija ciemats) un akceptēto stāvokļu kopums (šeit attēlots kā pils). Šeit visi ceļi ir veicami abos virzienos, kas nav nepieciešams magazīnas automātu gadījumā.

Magazīnas automāti ir saistīti ar īpašu formālo valodu klasi (bezkonteksta valodas).

Mistērija

Kanāda

Mistērijas pilī dzīvo vientuļš burvis. Burvis var pats sevi pārvērst par feju, vai arī uzburt feju sev blakus (labajā pusē no skatītāja skatpunkta). Feju var pārvērst par burvju dziru kreisajā un pūķi labajā pusē vai arī par trijotni: burvju dziru kreisajā pusē, burvi centrā un pūķi labajā pusē. Visas četras iespējamās burvestības ir attēlotas tabulā, parādot attiecīgajai burvestībai nepieciešamo pirms un rezultātu pēc tās:





Pirms	Pēc
	
	
	
	

Šīs burvestības var izpildīt neierobežotu reižu skaitu un jebkādā secībā. Tas ir, jebkurš burvis un jebkura feja var pārvērsties jebkurā brīdī.

Uzdevums

Ja sākumā ir viens pats burvis, tad kuru no attēlotajām objektu virknēm Mistērijā iegūt **nav** iespējams?



Atbilžu varianti







A	
B	
C	
D	

Atbildes izskaidrojums

Pareizā atbilde: B

Pieņemsim, ka visas burvestības ir sanumurētas ar skaitļiem no 1 līdz 4:

Numur s	Pirms	Pēc
1		

2		
3		
4		

Atbildi A var iegūt no viena burvja sākumā, pielietojot burvestības šādā secībā: 1, 4, 2 un 3.

Atbildi C var iegūt no viena burvja sākumā, pielietojot burvestības šādā secībā: 2, 2, 3, 4 un 1.

Atbildi D var iegūt no viena burvja sākumā, pielietojot burvestības šādā secībā: 2, 1, 3 un 3.

Viens no variantiem kā ieraudzīt, ka atbildē B dotais variants nav iespējams, ir pamanīt, ka burvestības vienmēr vienlaicīgi rada burvju dziru un pūķi. Tādējādi, virknē vienmēr jābūt vienādam skaitam burvju dziras trauku un pūķi. Tā tas nav atbildē B.

Tā ir informātika!

Burvestību var uzskatīt par noteikumu kopumu, ko izmanto, lai ģenerētu Mistērijas objektu virknes.

Datorzinātnēs bezkonteksta gramatika ir viens no rīkiem, ko izmanto, lai aprakstītu noteikumus, kas ģenerēs virknes. Bezkonteksta gramatikas var aprakstīt valodas (gan formālas, gan dabiskas), un, atkārtoti piemērojot gramatikas noteikumus, jūs varat ģenerēt vārdus (vai virknes), kas veido valodu.

Šajā uzdevumā jums bija nepieciešams noteikt, kurš no dotajiem "vārdiem" (objektu virknēm) neietilpst Mistērijas "valodā" (visu derīgo objektu virkņu kopā).

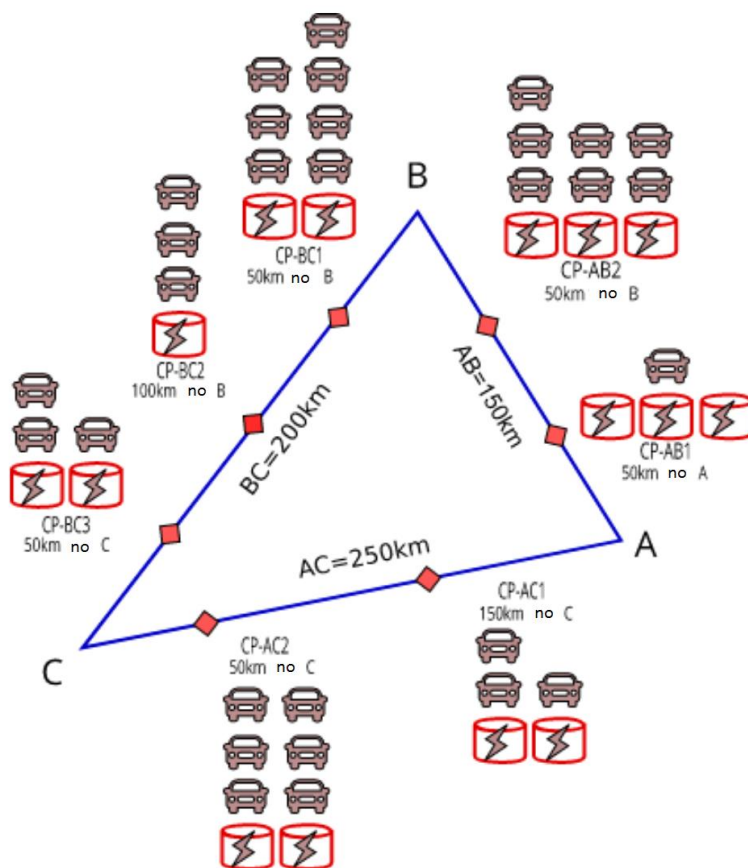
Elektromobiļu rindas

Īrija

Jums ir elektromobilis, kas pārvietojas ar ātrumu 100 km stundā un, ja ir pilnībā uzlādēts, tas var nobraukt 200 km. Elektromobili var uzlādēt jebkurā brīdī, pirms tas ir pilnībā izlādējies. Katrā uzlādē elektromobilis tiek uzlādēts pilnībā un uzlāde aizņem 1 stundu neatkarīgi no tā, cik izlādēts bija elektromobilis uzlādes sākuma brīdī.

Zemāk esošajā kartē redzami ceļi, kas savieno A, B un C. Jūs ceļojumu sākat punktā A un vēlaties apmeklēt punktus B un C pēc iespējas īsākā laikā. Varat izvēlēties jebkuru no diviem maršrutiem — $A \rightarrow B \rightarrow C$ vai $A \rightarrow C \rightarrow B$. Braucot cauri punktam B vai punktam C, jums jāapstājas uz 30 minūtēm, taču šajā laikā elektromobilis neizlādējas.

Maršrutā atrodas uzlādes stacijas, katrā ir noteikts uzlādes vietu skaits. Katrā uzlādes vietā jau var rindā būt citi elektromobiļi, tāpēc jums var nākties gaidīt, līdz tie pabeigs uzlādi, lai varētu izmantot šo uzlādes punktu. Piemēram, uzlādes stacijā CP-AB1 ir trīs uzlādes vietas, no kurām divas ir brīvas un vienā rindā stāv viens elektromobilis. Katra rindā stāvoša elektromobiļa uzlāde aizņem 1 stundu. Pirmais elektromobilis katrā rindā sāk uzlādi tieši tajā brīdī, kad dodaties ceļā no punkta A. Jūs ceļojumu sākat ar pilnībā uzlādētu automašīnu.



Uzdevums

Kāds ir īsākais laiks, kādā iespējams apciemot punktus B un C, sākot ceļojumu punktā A?

Atbilžu varianti

- A. 5 stundas
- B. 5 stundas 30 minūtes
- C. 6 stundas
- D. 6 stundas 30 minūtes
- E. 7 stundas

Atbildes izskaidrojums

Pareizā atbilde: B (5 stundas 30 minūtes)

To var panākt, izvēloties maršrutu A→B→C. Nobraukt visu posmu AB (150 km, 1 stunda 30 minūtes), nogaidīt prasītās 30 minūtes un nobraukt vēl 50 km līdz CP-BC1 (vēl 30 minūtes). Tātad elektromobilis nonāk CP-BC1 2 stundas un 30 minūtes kopš izbraukšanas no A un uzlādes vieta atbrīvosies pēc 30 minūtēm (3 stundas kopš ceļojuma sākuma). Elektromobiļa uzlādei būs nepieciešama viena stunda. Tas nozīmē, ka no CP-BC1 pilnībā uzlādēts elektromobilis izbrauks 4 stundas pēc ceļojuma sākuma. No CP-BC1 līdz C ir 150 km, kuru veikšanai nepieciešama 1 stunda un 30 minūtes - viss ceļojums ir ildzis 5 stundas 30 minūtes.

Ja dodaties no A uz C caur B, jūs ceļojat 350 km (3 stundas 30 minūtes) un pavadāt 30 minūtes punktā B. Turklāt jums ir jāveic vismaz viena uzlāde (1 stunda) posmā BC. Ja tā notiek CP-BC2 vai CP-BC3, tad pirms nepieciešama vēl viena uzlāde (vēl plus 1 stunda), kas nozīmē teorētiski mazāko ceļojuma laiku 6 stundas. Savukārt, ātrākais ceļojums ar uzlādi CP-BC1 (5 stundas 30 minūtes) jau tika aplūkots sākumā. Tas nozīmē, ka 5 stundas maršrutā A→B→C nav iespējamās.

Ja dodaties no A uz B caur C, jūs ceļojat 450 km (4 stundas, 30 minūtes) un pavadāt 30 minūtes punktā C. Turklāt jums ir jāveic uzlāde vismaz divas reizes (2 stundas), tāpēc minimālais teorētiskais ceļojuma laiks ir 7 stundas (pieņemot, ka uz uzlādi nebūs jāgaida).

Tā ir informātika!

Īsākā maršruta/ceļa atrašana punktu kopā ir zināms informātikas uzdevums. Šādu ceļu atrašana ļauj efektīvi transportēt cilvēkus un fiziskas preces no vienas vietas uz otru, kā arī Tīmeklī pēc iespējas ātrāk maršrutēt informācijas paketes līdz to galamērķiem .

Bebru datubāze

Itālija

Bebru ciematā dzīvo vairākas bebru ģimenes. Datorzinātnieks Bebrs Džefrijs ir izveidojis ciema iedzīvotāju datubāzi, ierakstot tajā datus par katru bebru 16 bitu virknes formā. Biti nozīme no b15 (pa kreisi) līdz b0 (pa labi) ir šāda:

b15 _ b12: četri biti ģimenes numuram;

b11: viens bits dzimumam (0 = mātīte, 1 = tēviņš);

b10 _ b4: septiņi svara biti (naturāls mārciņu skaits);

b3 _ b2: divi biti "strādnieka kvalifikācijai" (00 = namiņu celtniecība, 01 = dambju celtniecība, 10 = barības krātuvju ierīkošana, 11 = jauno bebru izglītošana);

b1 _ b0: divi biti iecienītākajam ēdienam (00 = koku miza, 01 = ūdensaugi, 10 = zāles, 11 = grīši).

Piemēram, virkne 0100 0 0100101 10 01 norāda, ka bebrs pieder pie 4. ģimenes, ir mātīte, sver 37 mārciņas, ir kvalificēta strādniece pārtikas krātuvju ierīkošanā un viņai patīk ūdensaugi.

Uzdevums

Bebrs Džefrijs veido vaicājumus - Būla izteiksmes, izmantojot loģiskās operācijas "un" (and), "vai" (or) un negāciju (not). Katras izteiksmes vērtība ir 0(aplama) vai 1(patiesa). Kādu bebru kopu apraksta sekojošā izteiksme?

b11 **and not** (b10) **and** b9 **and** b7 **and not** (b3 **and** b2)

Atbilžu varianti

A) Mātītes, kuras sver vismaz 16 mārciņas, ir kvalificētas strādnieces pārtikas krātuvju ierīkošanā.

B) Tēviņi, kuri sver vismaz 64 mārciņas, ir kvalificēti strādnieki namiņu vai dambju celtniecībā.

C) Tēviņi, kuri sver no 40 līdz 63 mārciņām, ir kvalificēti strādnieki kaut kā celtniecībā vai barības krātuvju ierīkošanā.

D) Tēviņi, kuru svars nepārsniedz 39 mārciņas, ir kvalificēti strādnieki dambju celtniecībā.

Atbildes izskaidrojums

Pareizā atbilde: C

Jo: $b_{11} = 1$ nozīmē "tēviņš", $b_{10} = 0$ nozīmē "maksimāli sver 63 mārciņas", $b_9 = 1$ un $b_7 = 1$ nozīmē "sver vismaz (32 + 8) mārciņas" un visbeidzot $b_3 = 0$ vai $b_2 = 0$ neietver tikai "kvalificētu darbinieku jauno bebru izglītošanā" ($b_3 = 1$ un $b_2 = 1$).

Ņemiet vērā, ka saskaņā ar De Morgana likumiem **not** (b_3 and b_2) atbilst **not** (b_3) or **not** (b_2).

A atbilde ir nepareiza: $b_{11} = 1$ nozīmē "tēviņš", bet atbildē minētas mātītes. Izteiksme, kas apraksta 1. atbildē doto bebru kopu, ir: **not** (b_{11}) and b_8 and b_3 and **not** (b_2).

B atbilde ir nepareiza: $b_2 = 0$ nozīmē arī "kvalificēts strādnieks barības krātuvju ierīkošanā"; turklāt $b_9 = 1$ un $b_7 = 1$ nosaka maksimālo svāra ierobežojumu. Izteiksme, kas apraksta 2. atbildē doto bebru kopu, ir: b_{11} and b_{10} and **not** (b_3).

D atbilde ir nepareiza: $b_3 = 0$ nozīmē arī "kvalificēts strādnieks namiņu celtniecībā"; turklāt $b_{10} = 0$ nosaka svāra augšējo robežu. Izteiksme, kas apraksta 4. atbildē doto bebru kopu, ir: b_{11} and **not** (b_{10}) and **not** (b_3) and b_2 .

Ņemiet vērā, ka kopa atbildē 1. nešķēļas ar kopu katrā no pārējām trim atbildēm, bet tā nav taisnība attiecībā uz pārējiem trim kopu pāriem (2. un 3.; 2. un 4.; 3. un 4. .) un neviena no kopām 2., 3. un 4. atbildē nav citas atbildes kopas apakškopa.

Tā ir informātika!

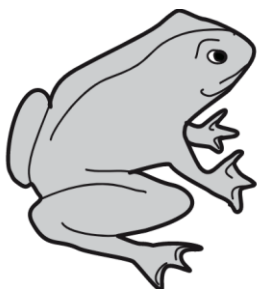
Iepriekš aprakstītā uzdevuma kontekstā Būla izteiksmi (vai, labāk, propozīciju loģikas formulu) var izmantot, lai veidotu vienkāršas datubāzes vaicājumus – tas ir, to var uzskatīt par "meklēšanas atslēgu". Parasti šāda formula identificē datu apakškopu (un šīs situācijas var grafiski attēlot, izmantojot Venna diagrammas). Šeit piedāvātā veida piemēri ir atrodami Konrāza Cūzes 1943.–1944. gada rakstos par propozīciju aprēķinu un to pielietojumu releju shēmās. Konrāds Cūze bija vācu būvinženieris, bet viņa lielākais sasniegums bija pirmais programmadītais elektromehāniskais dators, kas sāka darboties 1941. gadā.

Lielākajā daļā mūsdienu programmēšanas valodu ir Būla unārais operators not un binārie operatori and, or. Operatoriem not, and (tāpat kā not, or) pietiek ar tā sauktajiem De Morgana likumiem.

Būla shēmas, kas sastāv no atbilstošajiem loģiskajiem vārtiem, ir daudzu datoru inženierijā izmantoto digitālo komponentu (multiplekseri, summatori, aritmētiskās loģikas vienības utt.) pamatā. 1913. gadā (jau 1880. gadā to atklāja Čārlzs S. Pīrss) Henrijs M. Šefers parādīja, ka visus loģiskos operatorus var atvasināt no viena operatora nand: A nand B ir līdzvērtīgs not (A and B), tādējādi A nand A atbilst not (A). Līdzīgi nand ventiļiem, nor ventiļi (A nor B atbilst not (A or B), tādējādi A nor A atbilst not (A)) ir "universāli ventiļi" – tos kombinējot var izveidot jebkuru citu loģisko ventili, un ļoti daudzas elektroniskas sistēmas ir būvētas, izmantojot tikai nand vai nor loģiskos ventiļus.

Izkrāso vardi!

Itālija



Datora ekrānā parādīsies vārdes attēls. Nav zināms, kādā krāsā varde būs. Vārdes krāsas maiņai jums būs pieejamas četras datorprogrammas, bet izpildīt varēs tikai vienu no tām. Katra programma ir “**ja-tad-citādi**” vai “**ja-tad**” instrukciju virkne. Katras instrukcijas, izņemot pirmo, izpilde sākas tikai tad, kad iepriekšējās instrukcijas izpilde ir beigusies.

Jūs vēlaties, lai pēc programmas izpildes varde būtu *zaļā* krāsā, *neatkarīgi no vārdes krāsas programmas izpildes sākumā*.

Uzdevums

Tikai vienā gadījumā, kad sākotnēji varde bija noteiktā krāsā, vienas no programmām izpildes rezultātā varde netiks pārkrāsota zaļā krāsā.

Kura no programmām pēc tās izpildes beigām negarantē vārdes pārkrāsošanu zaļā krāsā?

Atbilžu varianti

- A **ja** varde ir sarkana, **tad** nokrāso to dzeltenu, **citādi** nokrāso to zaļu;
ja varde ir dzeltena, **tad** nokrāso to sarkanu;
ja varde nav dzeltena, **tad** nokrāso to zaļu.
- B **ja** varde ir sarkana, **tad** nokrāso to dzeltenu;
ja varde nav sarkana, **tad** nokrāso to zaļu;
ja varde ir dzeltena, **tad** nokrāso to sarkanu, **citādi** nokrāso to zaļu.
- C **ja** varde ir dzeltena, **tad** nokrāso to zaļu;
ja varde nav dzeltena, **tad** nokrāso to sarkanu;
ja varde ir sarkana, **tad** nokrāso to zaļu, **citādi** nokrāso to dzeltenu.
- D **ja** varde ir dzeltena, **tad** nokrāso to zaļu, **citādi** nokrāso to sarkanu;
ja varde ir sarkana, **tad** nokrāso to zaļu, **citādi** nokrāso to dzeltenu.

Atbildes izskaidrojums

Pareizā atbilde ir programma D: ja sākumā varde ir dzeltena, ar pirmās ja-tad-citādi instrukcijas izpildi tā tiek nokrāsota zaļā krāsā, pēc kuras nosacījums “varde ir sarkana” ir nepatiess un tāpēc tiks atkal iekrāsojās dzeltenā krāsā.

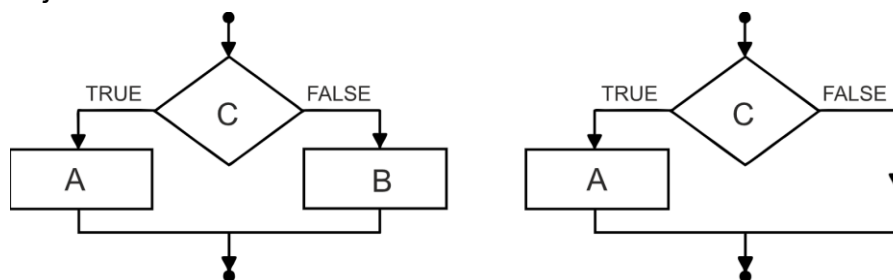
Palaižot programmu A, pēc ja-tad-citādi instrukcijas izpildes varde ir dzeltena vai zaļa; pēc otrās instrukcijas izpildes tā ir vai nu sarkana, vai zaļa, pēc kā nosacījums “varde nav dzeltena” ir patiess un tāpēc tā tiks iekrāsota zaļā krāsā.

Ja palaiž programmu B, tad, izpildot otro instrukciju, nosacījums “varde nav sarkana” noteikti būs patiess, tāpēc varde tiks iekrāsota zaļā krāsā, pēc kuras trešā instrukcija (ja-tad-citādi) to atkal iekrāsos zaļā krāsā.

Ja palaiž programmu C, tad, izpildot otro instrukciju, nosacījums “varde nav dzeltena” noteikti būs patiess, tāpēc varde tiks iekrāsota sarkanā krāsā, pēc kuras trešā instrukcija (ja-tad-citādi) to iekrāso zaļā krāsā.

Tā ir informātika!

Saskaņā ar imperatīvās programmēšanas paradigmu (un arī objektorientētā programmēšanā) varat uzrakstīt nosacījuma instrukciju virkni, kas jāizpilda norādītajā secībā. Nosacījuma priekšrakstam ir forma, ja C, tad A cits B, vai forma, ja C, tad A, atbilstoši tālāk parādītajām blokshēmām.



Jāuzsver, ka instrukcijas

ja C tad A citādi B

un

ja C tad A;

ja ne C tad B

vispārīgā gadījumā nav ekvivalentas. Tas ir, to rezultāti var atšķirties.

Patiesībā otrajā gadījumā (ja nosacījums tiek pārbaudīts divreiz) instrukcijas A izpilde varētu mainīt sistēmas stāvokli tā, ka nosacījums C kļūst aplams, izraisot arī instrukcijas B izpildi. Tā vietā pirmajā gadījumā vienmēr tiek izpildīta viena un tikai viena no divām instrukcijām.

Dotajā uzdevumā atšķirīgais rezultāts tiek izcelts, salīdzinot 4. un 3. programmu.

Tiek pieņemts, ka nosacījuma C pārbaude nemaina sistēmas stāvokli un katras instrukcijas izpilde galu galā tiek pārtraukta.

Jebkurā gadījumā instrukcija

ja C, tad A citādi B

var pārveidot par secību

```
jauns_nosacījums := C;
```

```
ja jauns_nosacījums, tad A;
```

```
ja ne jauns_nosacījums, tad B
```

kur jauns_nosacījums ir “svaigs” Būla mainīgais, kuram pirmā komanda piešķir Būla izteiksmes C vērtību.

Tīrā funkcionālās programmēšanas paradigmā nav komandu; jebkuru izskaitļojamu funkciju var norādīt, izmantojot nosacījuma izteiksmes un rekursiju. Nosacījuma izteiksmes ir formā “**ja** C, **tad** E1 **un** E2”, kur E1 un E2 ir viena veida izteiksmes. Piemēram, naturāla skaitļa n faktoriāls fakt(n) var tikt aprēķināts, aprēķinot nosacījuma izteiksmi

ja n = 0, **tad** 1 **citādi** n * fact(n – 1);

levērojiet, ka izmantota rekursija. Parasti **citādi** gadījumu nevar izlaist, jo rezultāts ir jānorāda arī tad, ja nosacījums ir aplams.

Klausies un ej!

Vācija

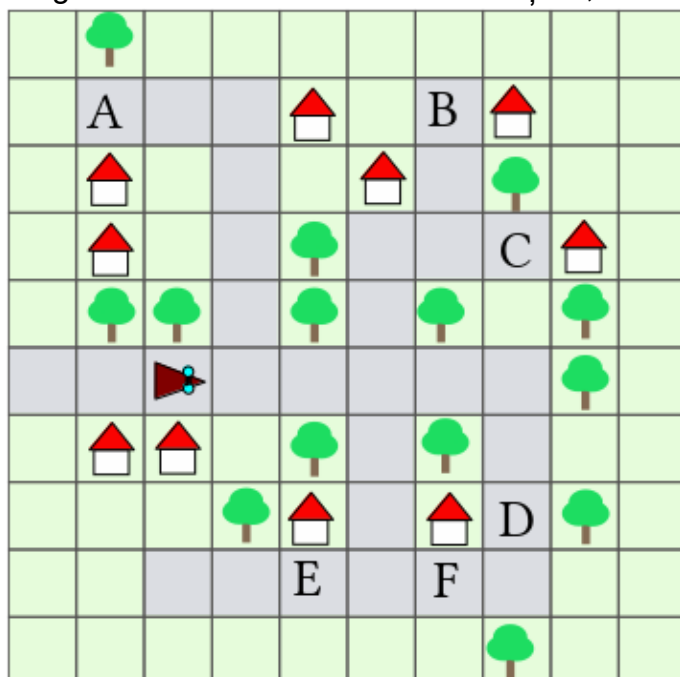
Tīna ir neredzīga, tāpēc pārvietoties pa ceļiem pilsētā var tikai izmantojot runājošās brilles. Runājošajām brillēm ir kameras un gudra objektu atpazīšanas sistēma. Tā var atpazīt četrus objektu veidus, kas attēloti arī zemāk redzamās kartas rūtiņās: māju, koku, ceļu un zālienu.

Tīnai pārvietojoties un nonākot vietā, kas kartē atbilst ieiešanai jaunā kartes rūtiņā, runājošās brilles viņai pasaka priekšā – tieši šādā secībā: kas atrodas pa kreisi, kas priekšā un kas – pa labi; piemēram: “koks ceļš māja”.

Tīna sāk (ar skatu uz kartes labo pusi) rūtiņā ar trīsstūri un klausās, ko saka viņas brilles. Lūk, ko brilles teica (sākot ar sākuma laukumu):

**koks ceļš māja,
ceļš ceļš zāliens,
koks ceļš koks,
ceļš ceļš ceļš,
koks ceļš koks,
koks māja ceļš,
ceļš ceļš koks,
māja ceļš koks**

Beigās Tīna ir nonākusi vienā no rūtiņām, kas apzīmētas ar burtu.



Uzdevums

Kurā mājā Tīna nonāks?

Atbilžu varianti

Uzklīkšķiniet uz pareizās atbildes burta un nospiediet “Saglabāt”.

Tā ir informātika!

Uzdevums parāda, kā zinot tikai ziņojumu, ko runājošās brilles teica Tīnai, virkni pietiek, lai uzzinātu maršruta pēdējo rūtiņu starp vairākām iespējām. Kopumā varētu domāt, ka ziņojumu virkne nav pietiekami informatīva, lai noteiktu, kur Tīna atrodas katrā brīdī, taču mūsu rīcībā esošās papildu zināšanas par vidi, kurā viņa darbojas, ir bagātīgas un mēs varam izsekot viņas maršrutam un noteikt galamērķi. Vienmēr ir grūti spriest, cik sensitīva var būt personas informācija, un tas attiecas arī uz telpiskajiem vai ģeogrāfiskajiem datiem.

Objektu atpazīšana ir svarīga datorzinātņu tēma. Mākslīgā intelekta sistēmu var apmācīt atpazīt tādus objektus kā mājas, koki un ceļu virsmas. Iespējams, jūsu viedtālruna kamera spēj atpazīt sejas un norāda tās ierāmējot. Autonomajam auto savā vidē jāatpazīst cilvēki, citas automašīnas, ceļa zīmes un citi objekti, kas ir svarīgi lai pieņemtu lēmumu, kurp doties tālāk. Lai palīdzētu neredzīgiem un vājredzīgiem cilvēkiem, mūsdienu tīmekļa mediji ietver automātisko alternatīvo tekstu (AAT). Šī tehnoloģija izmanto objektu atpazīšanu, lai automātiski izveidotu attēlu verbālus aprakstus (alternatīvos tekstus). “Runājošās brilles” joprojām ir zinātniskā fantastika, taču pastāv reāli projekti, kas mēģina īstenot šo ideju.

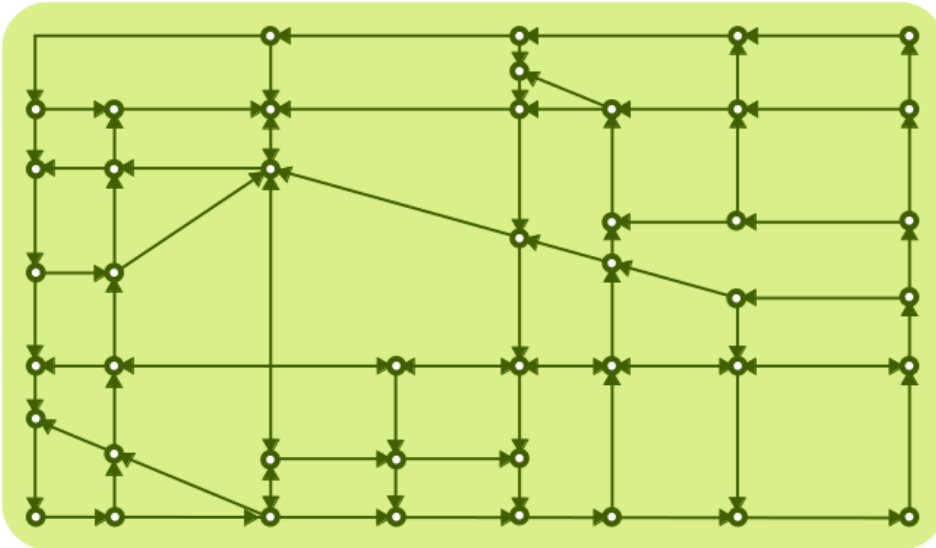
Vārīgie krustojumi

Nīderlande

Šajā pilsētas kartē var redzēt, ka daudzas ielas ir vienvirziena – tās attēlotas kā nogriežņi, kuru vienā galā ir bultiņa.

Dažās ielās ir divvirzienu satiksme, ko attēlo kā nogriežņus, kuru abos galos ir bultiņas.

Krustojumu sauc par *vārīgu*, ja to var sasniegt tikai no viena cita krustojuma.



Uzdevums

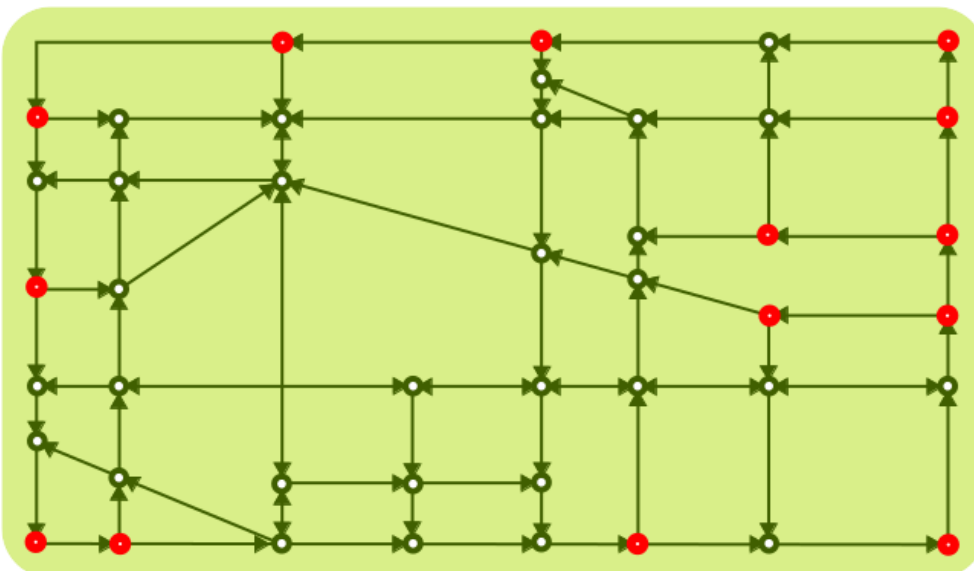
Cik daudz vārīgu krustojumu attēloti pilsētas kartē?

Atbilžu varianti

Vesels nenegatīvs skaitlis robežās [0,44]

Atbildes izskaidrojums

Kartē attēloti 14 vārīgi krustojumi (atzīmēti ar sarkaniem aplīšiem):



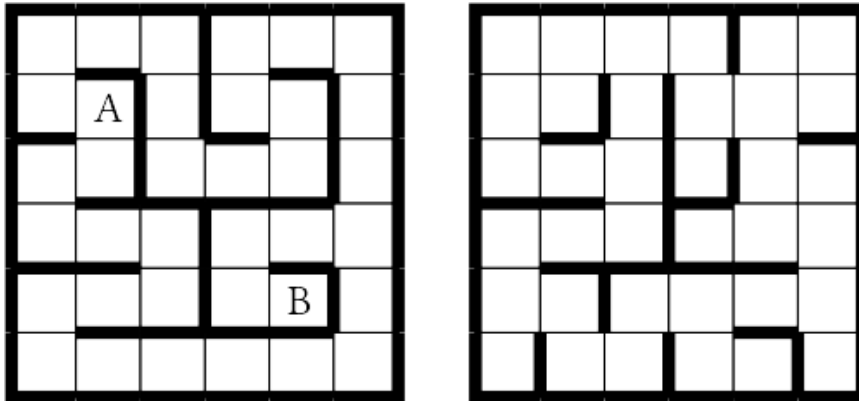
Tā ir informātika!

Šajā pilsētas kartē redzamo struktūru sauc par tīklu. Uzbrukums neaizsargātai tīkla vietai var novest pie tā, ka noteiktas tīkla daļas kļūst nepieejamas. Tīkla struktūras analīze, meklējot šīs vārgās vietas, ir svarīga tīkla projektēšanā, lai samazinātu, piemēram, pārsūtāmās informācijas zudumus. Tīkls ir orientēts grafs, un to bultu skaitu, kas norāda uz noteiktu grafa virsotni(krustojumu), sauc par šī punkta ieejošo pakāpi. Krustojums ir vārgs, ja tā ieejošā pakāpe ir 1, un nepieejams, ja pakāpe ir 0, lai gan šajā tīklā tādu krustojumu nav.

Labirints

Uzbekistāna

Bebrēns ir nonācis labirintā, kurš izvietots divos stāvos. Katrā stāvā ir savs sienu izvietojums (skat. zīm.).



Bebrēns var pārvietoties vienā stāvā no vienas blakus rūtiņas uz otru, ja to kopīgajā malā nav sienas – tam nepieciešama viena sekunde. Vēl bebrēns var izmantot brīnumnūjiņu, lai pārvietotos uz atbilstošu rūtiņu otrā stāvā – tam nepieciešamas piecas sekundes.

Piemēram, no rūtiņas A ir iespējami trīs atšķirīgi gājieni:

1. Pāriet uz blakus rūtiņu, kas attēlā ir pa kreisi. Gājienam nepieciešama viena sekunde.
2. Pāriet uz blakus rūtiņu, kas attēlā ir uz leju. Gājienam nepieciešama viena sekunde.
3. Pāriet uz atbilstošu rūtiņu otrā stāvā. Gājienam nepieciešamas piecas sekundes.

Bebrēns atrodas rūtiņā A un pēc iespējas ātrāk vēlas sasniegt rūtiņu B.

Uzdevums

Kāds ir mazākais laiks sekundēs kādā bebrēns no rūtiņas A var nokļūt rūtiņā B?

Atbilžu varianti

Ieraksti naturālu skaitli – nepieciešamais laiks sekundēs.

Atbildes izskaidrojums

No A uz B iespējams nokļūt, ātrākais, 18 sekundēs.

Dotais uzdevums ir īsākā ceļa atrašanas uzdevums. Lai gan pastāv vairāki uzdevuma atrisināšanas veidi, parādīsim vienu no tiem, izmantojot Dijkstras algoritmu.

Attēlā zemāk katrai rūtiņai parādīts ātrākais laiks kā līdz tai nokļūt no rūtiņas A.

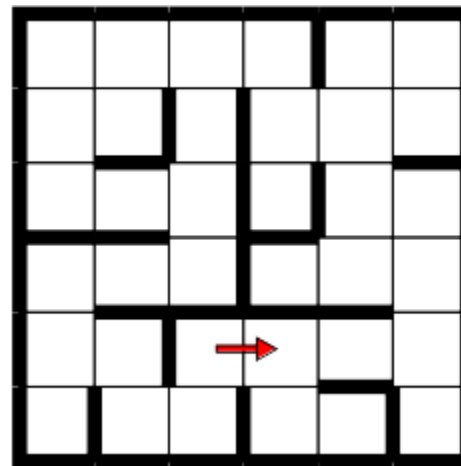
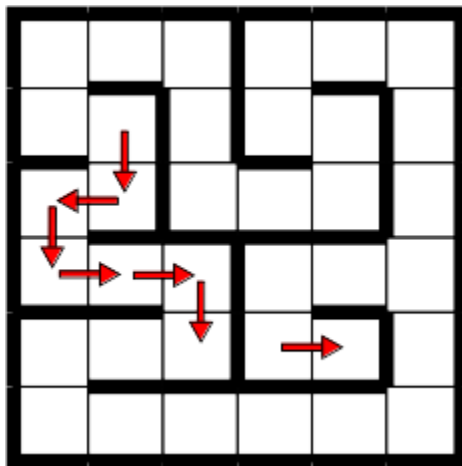
Tādējādi, redzams, ka ātrākais laiks kā sasniegt B, ir 18.

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

Optimālais maršruts ir šāds:

Laika robežu novērtējumu var iegūt šādi – 20 sekundes ir nepieciešamas, lai no A līdz B tiktu nepārejot uz otru stāvu. Savukārt, 16 sekundes atbilst situācijai, kur notiek pāreja uz otru stāvu un atpakaļ, bet otrā stāvā nebūtu sienu (t.i., Manhetenas attālums starp A un B(6) + laiks pārejai starp stāviem(10)).



Tā ir informātika!

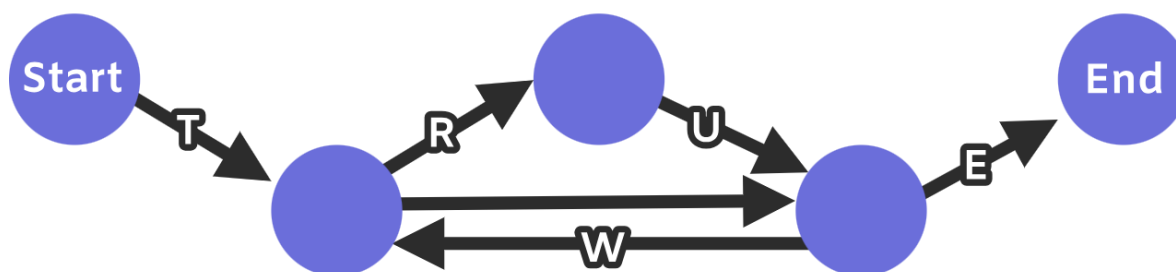
Īsākā ceļa problēma ir labi zināma grafu teorijas problēma. Tās lietojumi ietver optimāla ceļa atrašanu datortīklā, pilsētas kartē utt.

Īsākā ceļa uzdevumam režģa diagrammā ir dažādi lietojumi. Piemēram, vadu maršrutēšana daudzelementu mikroshēmu izstrādē (tā sauktais "labirinta skrējējs"). Konkrētajā uzdevumā pārvietošanās starp stāviem aizņem vairāk laika, salīdzinot ar pārvietošanos vienā stāvā (5 pret 1). Tas atbilst faktam, ka daudzslāņu mikroshēmās šķērslīmeņu savienojumi ir dārgāki nekā savienojumi viena slāņa ietvaros. Parastās daudzelementu mikroshēmas tiek veidotas izmantojot 10 slāņu silīcija kristālu, bet šajā uzdevumā bija tikai divi slāņi. Turklāt aprakstīto mikroshēmu projektēšanā ir jāmaršrutē miljoniem un miljardiem vadu, kas savieno tranzistorus, ieejas, izejas un atmiņas šūnas.

Vārdi

Amerikas Savienotās Valstis

Birgita veido gadījuma “vārdus”, izmantojot attēlā redzamo diagrammu. Viņa vienmēr sāk no lauciņa ‘Start’ un, sekojot bultām, pieraksta burtu virknes galā kārtējo sastapto burtu. Birgita vārda veidošanu beidz tad, kad nonāk lauciņā ‘End’.



Piemēram, šādi viņa var izveidot “vārdus” ‘TWWWE’ un ‘TRUE’.

Uzdevums

Cik daudz atšķirīgu astoņburtīgu “vārdu” Birgita šādi var izveidot?

Atbilžu varianti

Ievadīt vārdu skaitu kā naturālu skaitli.

Atbildes izskaidrojums

Pareizā atbilde: ir 9

Ievērosim, ka vārds vienmēr sākas ar ‘T’ un beidzas ar ‘E’. Tādējādi, mūs interesē vārdu fragmenti garumā 6, kas sastāv no burtiem ‘R’, ‘U’ un ‘W’, turklāt diagrammā šis fragments sākas otrajā aplīti (kur ienāk ‘T’), bet beidzas priekšpēdējā (no kura iziet ‘E’).

Aplūkosim, cik un kā iespējams izveidot noteikta garuma fragmentus ar šīm īpašībām:

- Garumā 1: 1 (W; tukšā bulta, kam seko W, un tukšā bulta)
- Garumā 2: 2 (WW un RU)
- Garumā 3: 3 (WWW; WRU; RUW)
- Garumā 4: 4 (WWWW; WWRU; RUWW; WRUW)
- Garumā 5: 6 (WWWWW; WWRUW; WWRUW; WRUWW; RUWWW; RUWRU)
- Garumā 6: 9 (WWWWW, WWWWRU, WWWRUW, WWRUWW, WRUWWW, RUWWWW, WRUWRU, RUWRUW, RUWWRU)

Tā ir informātika!

Visu iespēju izskatīšana un pārbaude ir tas, ko dators patiešām spēj.

Šajā gadījumā jūs apzināti cenšaties pierakstīt visus fragmentus, taču jāatceras, ka jāuzmanās tos neskaitīt divreiz.


Šī ir algoritmiska "mašīna" vai rutīna, kas ģenerē burtu virkni ar noteiktām īpašībām. Šāda veida sistēmas dažkārt tiek izmantotas, lai ģenerētu noteikta formāta gadījuma paroles (piemēram, satur noteiktu zīmju skaitu, starp kurām ir arī speciālie simboli).





Bebru spēles

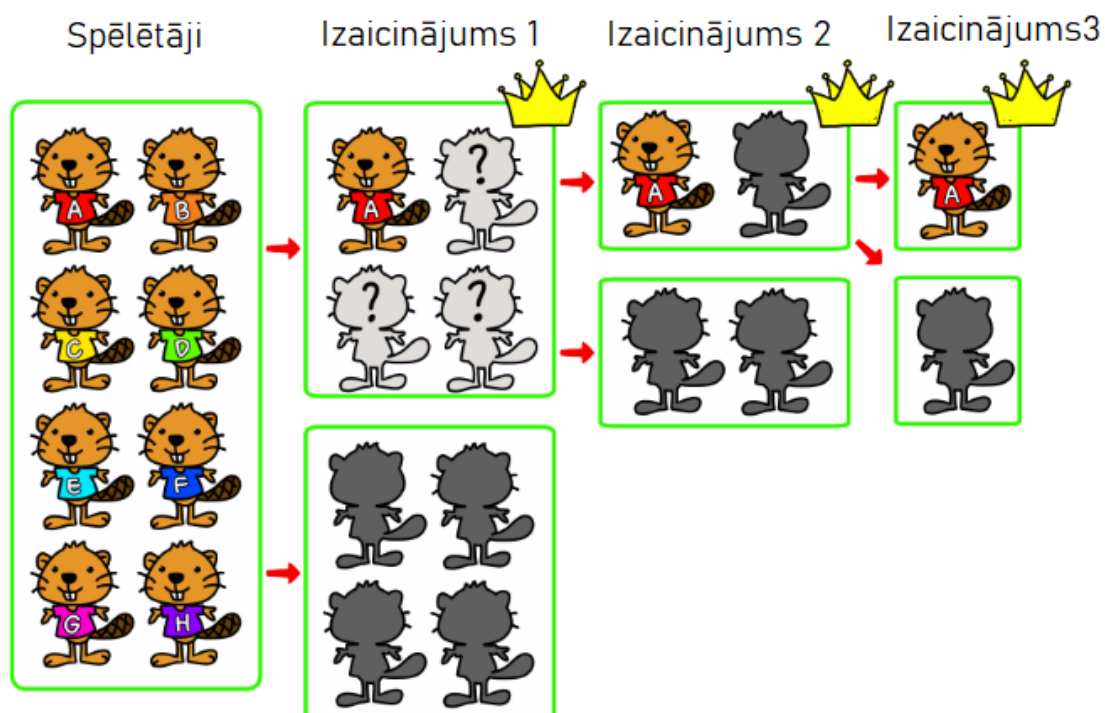
Korejas Republika

Bebru spēles ir ikgadējas sacensības, kas sastāv no trim izaicinājumiem. Pirmais un otrais izaicinājums tiek izspēlēti komandās, kur komandas dalībnieku sastāvus nosaka izloze. Komanda ar lielāku dalībnieku punktu kopsummu uzvar, bet visas komandas pāriet uz nākamo izaicinājumu.

Katrā nākamajā izaicinājumā komandas dalībnieku skaits ir divreiz mazāks nekā iepriekšējā izaicinājumā. Pēdējais izaicinājums ir individuāla spēle (komandā palicis viens dalībnieks). Ja jūs esat uzvarējis visos trīs izaicinājumos, jūs esat uzvarētājs.

Bebris Ada  ir šīgada spēļu uzvarētāja. Katra dalībnieka katrā izaicinājumā iegūto punktu skaits parādīts tabulā:

Vārds	Ada 	Bruno 	Cilda 	Deizija 	Ede 	Fanija 	Georgs 	Hauks 
1. 1. izaicinājums	15	16	19	18	17	20	19	19
2. 2. izaicinājums	20	27	30	24	28	24	30	30
3. 3. izaicinājums	10	14	11	15	16	13	9	12



Uzdevums

Kuri trīs dalībnieki bija Adas komandā 1. izaicinājumā?

Atbildes izskaidrojums

Adas komandas biedri pirmajā izaicinājumā ir Deizija 🌻, Fanija 🧑🏻 un Georgs 🧑🏻.

Trešais izaicinājums ir individuāla spēle. Tā kā Georgs ir vienīgais spēlētājs, kuram ir mazāk punktu nekā Adai, otrajā izaicinājumā viņiem bija jābūt vienā komandā.

Viņu punktu summa otrajā izaicinājumā ir 50. Tai ir jābūt lielākai par otras divu bebru komandas punktu summu. Divas spēlētājas Deizija un Fanija ir vienīgās, kuru punktu summa ir mazāka par 50. Tāpēc viņām ir jābūt vienā komandā ar Adu un Georgu pirmajā izaicinājumā. Tā kā mēs tagad zinām šos dalībniekus, mums vairs nav jāpārbauda otras komandas dalībnieki.

Tātad tas nozīmē, ka pirmajā izaicinājumā komandai (Ada, Deizija, Fanija, Georgs) bija 72 punkti, savukārt (Bruno, Cilda, Ede, Hauks) bija 71 punkts. Tātad Adas komanda uzvar. Otrajā izaicinājumā (Ada, Georgs) bija 50 punkti, bet (Deizija, Fanija) bija 48 punkti. Trešajā izaicinājumā Ada uzvar ar 10 punktiem pret Georgu ar 9 punktiem. Tātad Ada ir bebru spēļu uzvarētāja.

Tā ir informātika!

Ja kāds mēģinātu šo uzdevumu atrisināt ar visi variantu pārslasi, tad būtu jāizmēģina visas iespējamās kombinācijas, kas ir kombināciju skaits no 7 pa 3, jeb 35, un katrai no šīm kombinācijām jānovērtē izaicinājumu rezultāti. Tas aizņemtu daudz laika.

Šādos gadījumos datorzinātnieki meklē efektīvākas uzdevuma risināšanas metodes. Šajā gadījumā tā vietā, lai atrisinātu problēmu ar pārslasi, ļoti ātri risinot no beigām, tiek iegūts pareizais risinājums, kā parādīts atbildes skaidrojumā.

Šo metodi sauc par atpakaļejošo meklēšanu, un to izmanto situācijās, kad tiek meklēts risinājums, kam jāatbilst noteiktiem ierobežojumiem. Dažos gadījumos abus meklēšanas virzienus var kombinēt.

Robota maršruts

Somija

Drons savu ceļojumu sāk kādā no laukuma baltajām rūtiņām. Tas ir vērsts vienā no četriem iespējamiem virzieniem - pret kādu no rūtiņas malām.

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

Tad tas apstaigā tieši astoņas citas baltās rūtiņas pēc šāda algoritma:

1. Pāiet divas rūtiņas uz priekšu.
2. Esošajā rūtiņā pagriezies 90 grādus pa kreisi.
3. Pāiet četras rūtiņas uz priekšu.
4. Esošajā rūtiņā pagriezies 90 grādus pa labi.
5. Pāiet divas rūtiņas uz priekšu.

Uzdevums

Cik daudz ir laukuma rūtiņu, kurās varēja sākties aprakstītais robota maršruts?

Atbilžu varianti

Jāievada vesels nenegatīvs skaitlis.

Atbildes izskaidrojums

Atbilde ir 8. Vispirms ievērosim, ka ir četri drona apmeklēto rūtiņu varianti:

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

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

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

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

Lai atrastu šos variantus, iespējams, visvieglāk ir sākt ar piecu secīgu baltu rūtiņu meklēšanu un pēc tam mēģināt paplašināt galapunktus pretējos perpendikulāros virzienos (ievērojot, ka pirmajam pagriezienam jābūt pa kreisi!). Lai pārliecinātos, ka atrasti visi

varianti, pārlase jāveic sistemātiski (piemēram, no laukuma augšas kreisās puses uz apakšējo labo pusi).

Drona ceļš ir simetrisks, tāpēc drons katrā no četrām iespējam maršrutu var sākt no diviem galapunktiem. Tāpēc ir 8 dažādas iespējamās sākuma rūtiņas.

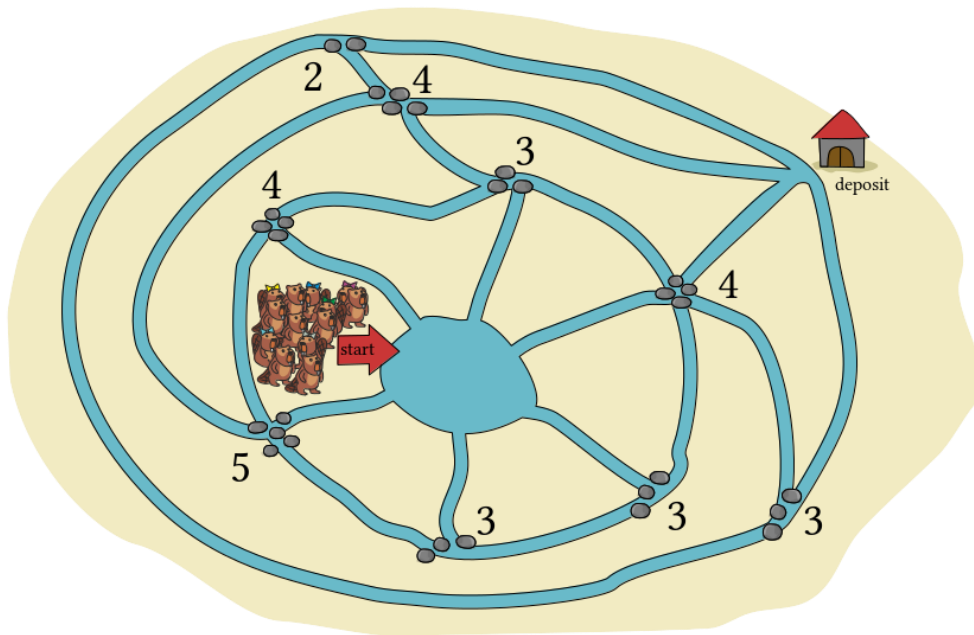
Tā ir informātika!

Uzdevums ir vienkāršs rotācijas attēla (vai parauga) invariantu atbilstības piemērs, kas ir viens no uzdevumiem plašākā datorzinātņu attēlu apstrādes jomā. Šeit uzdevumā tika definēts meklējamais paraugs(1.–5. darbība), un testa režģis bija attēls, kurā paraugu meklēt. Paraugu atrašanai ir vairākas lietojumprogrammas, un šī pieeja tiek izmantota seju atpazīšanā un medicīnisko attēlu apstrādē. Dažās paraugu atbilstības metodēs meklējamais paraugs bieži tiek aprakstīts kā maršruts.

Akmeņu savākšana

Šveice

Bebri ģimenes māju veido 21 kanāls, kuru attīrīšanai nepieciešams savākt 31 traucējošu akmeni. Bebri nolēmuši savākt akmeņus, vienlaikus sākot darbu no mājas centra (kartē "start") un nogādāt tos krātuvē (kartē "deposit").



Akmeņi ir smagi, tāpēc vienlaikus katrs bebrs var pārvietot tikai vienu vai divus no tiem, bet ne vairāk. Lai nokļūtu no viena kanālu krustojuma līdz nākamajam, bebri vienmēr peld tieši vienu stundu. Pēc vienlaicīgas darbu sākšanas bebbiem četrus stundu laikā jāsavāc visi akmeņi.

Uzdevums

Kāds mazākais bebru skaits nepieciešams aprakstītā darba paveikšanai?

Atbilžu varianti

Ievadi naturālu skaitli robežās no 1 līdz 31

Atbildes izskaidrojums

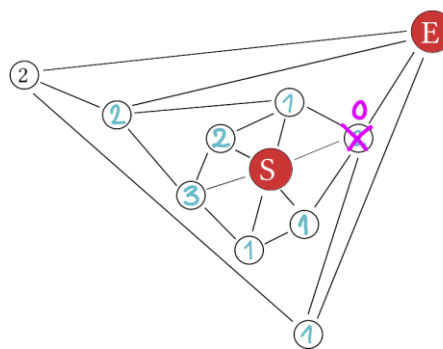
Pareizā atbilde: 14

Īsā versija

Ir tikai viens ceļš, pa kuru bebri var sasniegt krātuvē divu stundu laikā. Šajā ceļā ir četri akmeņi. Divi bebri var nogādāt šos četrus akmeņus krātuvē un var turpināt darbu un paspēt nogādāt krātuvē vēl četrus akmeņus. Tad atliek $23 = 31 - 8$ akmeņi un nepieciešami vēl vismaz 12 bebri, jo atlikušie akmeņi atrodas uz ceļiem, kuru garums ir 3 vai 4, tāpēc četrus stundu laikā pēc akmeņu nogādes vairs neatliek laika citu akmeņu nogādāšanai. Apskatot akmeņu izvietojumu, var izdomāt, ka atlikušos 23 akmeņus krātuvē var nogādāt 12 bebri, no kuriem 11 bebri pārvietos divus un viens – tikai vienu akmeni.

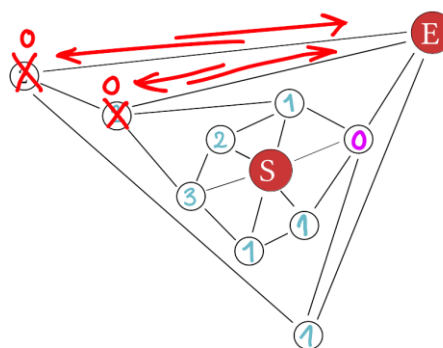
Otrā (detalizētā) versija

Vēl viens bebrs savāc divus akmeņus divu stundu laikā. Mēs varam “atkārtoti izmantot” arī šo bebru. $15-2=13$ akmeņi vēl palikuši. Līdz šim izmantojam deviņus bebrus. Divus no tiem var “izmantot atkārtoti”.

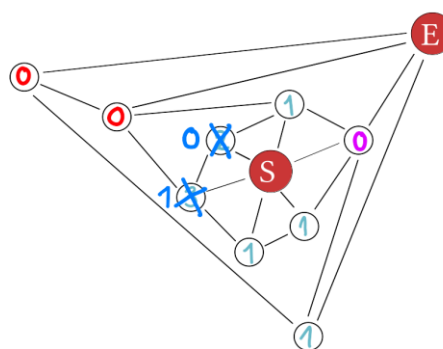


Tagad abus bebrus var “izmantot atkārtoti”. Katrs no tiem savāc divus akmeņus.

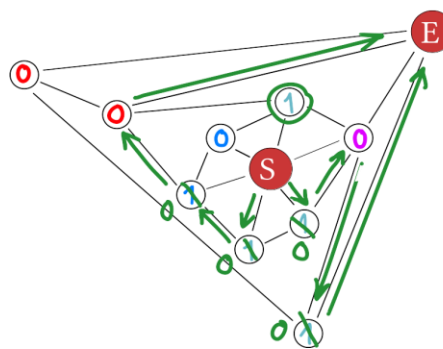
$13-4=9$ akmeņi vēl palikuši. Līdz šim izmantojam deviņus bebrus. Nevienu no tiem nevar “atkārtoti izmantot”.



Mēs izmantojam to pašu stratēģiju un tālāk aplūkojam visus krustojumus ar vismaz diviem akmeņiem. Mums vajag divus bebrus. Viņi savāks četrus akmeņus. $9-4=5$ akmeņi vēl palikuši. Līdz šim izmantojam $9+2=11$ bebrus.



Pēdējais solis ir atrast risinājumu (iespējams, izmēģinājumu un kļūdu ceļā), kas izmanto pēc iespējas mazāk bebru. Divi bebri savāc pa diviem akmeņiem. Trešais un pēdējais bebrs savāc pēdējo akmeni. Labākais risinājums ir A): $11+3= 14$ bebrī.



Ir 31 akmens, tikai divi bebrī var atnest pa diviem akmeņiem un tad atgriežoties pēc vēl citiem, citi bebrī var nogādāt ne vairāk kā divus akmeņus. Tātad nepieciešami vismaz 13,5 bebrī. Labākajam risinājumam nepieciešami 14 bebrī.

Tā ir informātika!

Lasītāja acīs stāsts ir pievilcīgs un detaļām bagāts. Tomēr pirmā datorzinātnieka darbība ir informācijas, kas patiešām ir nepieciešama problēmas risināšanai, noteikšana un visa pārējā atmešana.

Nozīmīgā informācija ietver:

- kanālus un to krustojumus (akmeņi, starts un krātuve).
- skaitliskos ierobežojumus: bebrs no viena krustojuma līdz nākamajam peld tieši vienu stundu; bebrim jānonāk krātuvē četrus stundu laikā; viņiem jāsavāc 31 akmens; tie katrs var nest vienu vai divus akmeņus.
- ir jāsavāc 31 akmens, izmantojot pēc iespējas mazāk bebru.

No otras puses, mums nav jāņem vērā precīza kanālu atrašanās vieta kartē, bebru izskats utt. Šo informāciju var izlaist.

Datorzinātnieki šādu formālu attēlojumu sauc par grafu. Graf sastāv no virsotnēm un šķautnēm starp tām. Attālumu starp divām virsotnēm var izteikt vairākos veidos, piemēram, saskaitot šķautņu skaitu, kas mums jāšķērso, lai nokļūtu no vienas virsotnes uz otru. Grafi ir viens no galvenajiem datorzinātnieka uzdevumu risināšanas instrumentiem. Navigācijas sistēmas ir labi zināms lietojumprogrammu piemērs, kas lielā mērā izmanto grafus.

Mūsu uzdevumā bebrim ir paredzēts savākt 31 akmeni. Iespējams, ir vairāk nekā viens veids, kā sasniegt šo mērķi. Varētu iedomāties izmantot 14, 18, 20 vai pat 31 bebru. Tomēr uzdevums nepārprotami pieprasa risinājumu, kas izmantotu pēc iespējas mazāk bebru. No šī viedokļa mums rūpīgi jāapsver, vai ir kāds veids, kā bebru "atkārtoti izmantot".

Uzdevumi, kuros meklē labāko iespējamo risinājumu, ir ļoti izplatīti. Informātikas terminoloģijā tās ir optimizācijas problēmas. Navigācijas sistēmai ir jāizvēlas labākais veids, kā nokļūt no A līdz B. Protams, a priori ir jādefinē precīza "labākā" risinājuma nozīme. Navigācijas sistēmas gadījumā kritērijs varētu būt attāluma samazināšana kilometros vai laiks, kas automašīnai nepieciešams mērķa sasniegšanai.

Skaitļošanas domāšanas ziņā skolēni, iespējams, nedaudz vienkāršos dotā uzdevuma attēlojumu – akmeņu zīmēšanas vietā rakstīs skaitļus un attēlos kanālus ar līniju palīdzību. Tomēr attēls jau ir neformāls, bet precīzs grafa veids. Skolēniem ir jāsadala uzdevums mazākās un pārvaldāmās daļās, jāatrisina (iespējams, ar izmēģinājumu un kļūdu palīdzību) katra no tām un jāizsecina, vai viņu risinājums atbilst visiem ierobežojumiem un izņēmumiem, kas varētu rasties.