

Bebr[a]s

'21

1. kartas uzdevumi

11.-12. klasei

INFORMATĪVAIS
ATBALSTĪTĀJS

start(it)

www.startit.lv

SATURS



| | |
|----------------------------|----|
| Cepure un žetoni..... | 3 |
| Darbs grupās | 6 |
| Defektoskopija | 8 |
| Steidzamā satikšanās | 10 |
| Krāsu loģika | 13 |
| Reģionālās slimnīcas | 15 |
| Kompaktais ziņojums | 17 |
| Revīzijas komiteja | 19 |
| Iepirkšanās | 21 |
| Garākais fragments..... | 23 |
| Simbolu robots..... | 25 |
| Stumbru kārtošana..... | 29 |
| Cilšu apvienošanās | 31 |
| Datu pārveidošana | 33 |
| Alu business | 35 |

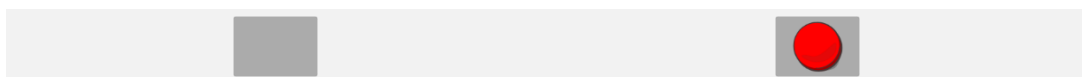
Cepure un žetoni

Čehija

Bebriem patīk spēlēt spēli, kas norisinās spēles laukumā, ko veido virknē izvietoti lauciņi, un kurā tiek izmantoti sarkani žetoni.



Bebriem ir jāpārvieto cepture no kreisā gala uz labo no viena lauciņa uz citu . Bebram ir jācepure un viņš uzvedas atšķirīgi atkarībā no tā, vai cepture atrodas rokā vai galvā. Attēlā parādītas laukuma izmaiņas kā "pirms  pēc" atkarībā no lauciņa saturu (tukšs vai tajā novietots žetons) un ceptures atrašanās vietas (rokā vai galvā).



Vienkārši pāriet uz nākamo lauciņu.



Noņem žetonu no lauciņa, cepuri uzliek galvā un pāriet uz nākamo lauciņu.



Lauciņā noliek žetonu, cepuri paņem rokā un pāriet uz nākamo lauciņu.



Žetonu atstāj lauciņā un pāriet uz nākamo lauciņu.








Uzdevums:

Spēles sākumā bebram cepture ir rokā un žetoni laukumā izvietoti tā, kā redzams zīmējumā:



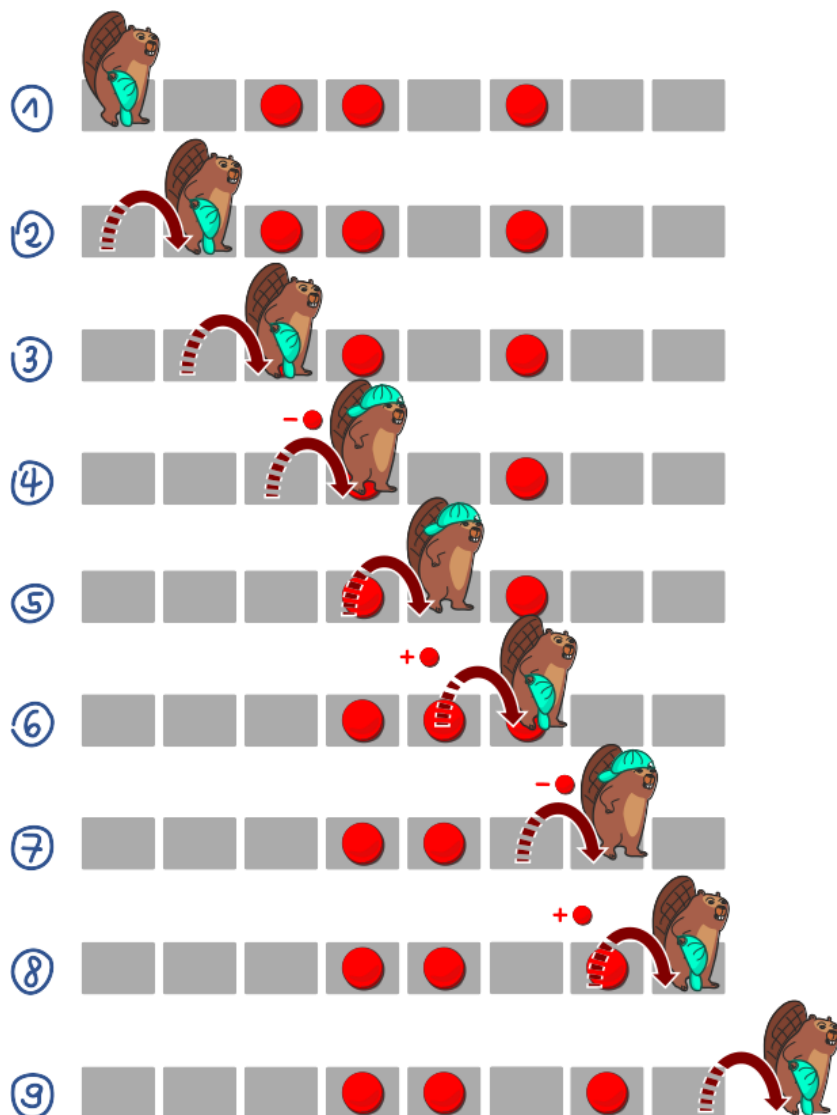
Kuros lauciņos atradīsies žetoni pēc bebra pārvietošanās no kreisā laukuma gala uz labo un pēdējā lauciņa pamešanas?

Atbilžu varianti:

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

Pareizā atbilde: C

Atbildi var atrast, veicot aprakstītā algoritma izpildi pa soļiem:



Atbildes izskaidrojums:

Bebram ir divi cepures stāvokļi: rokā vai galvā. Atkarībā no stāvokļa, tas uzvedas atšķirīgi. Bebrs līdz ar aprakstītajiem noteikumiem un lauciņiem darbojas kā Tjūringa mašīna. Tjūringa mašīna datorzinātnēs ir svarīgs un noderīgs skaitļošanas modelis. Lai gan tas ir ļoti vienkāršs, tas ir tikpat spēcīgs un efektīvs kā jebkura programmēšanas valoda. Tas nozīmē, ka mēs varam pārvērst jebkuru programmatūru Tjūringa mašīnā un, gluži pretēji, jebkuru Tjūringa mašīnu - programmā. To pirmo reizi 1936. gadā aprakstīja angļu matemātiķis un datorzinātnieks Alans Tjūrings.

Tjūringa mašīnai ir vairākas sastāvdaļas:

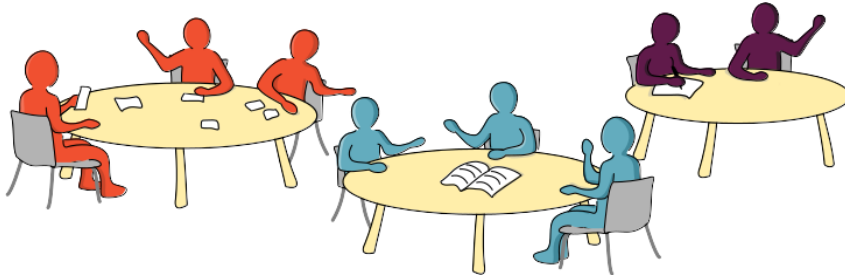
- Gara lente, kas sadalīta rūtiņās. Parasti uzskata, ka lenta ir bezgalīgi gara.
- Galīgs simbolu alfabēts, piemēram, 0, 1. Uzdevumā mēs izmantojām lauciņus ar žetonu un neaizpildītus lauciņus.

- Lasīšanas/rakstīšanas galviņa: tā varēs aplūkot rūtiņu un izlasīt tajā ierakstīto simbolu. Pēc simbola nolasišanas saskaņā ar noteikumiem galviņa pēc tam pārvietojas pa kreisi vai pa labi vienu rūtiņu. Mūsu gadījumā bebrš attēlo lasīšanas/rakstīšanas galviņu.
- Galīga stāvokļu kopa. Uzdevumā tiek izmantoti divi stāvokļi: cepure rokā un cepure galvā.
- Noteikumu kopums (pārejas noteikumi): lai aprakstītu, kā mašīna darbojas.

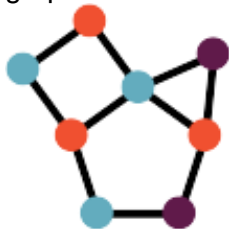
Darbs grupās

Čehija

Astoņi skolēni, lai izpildītu mājasdarbus, parasti sadalās trīs grupās, un katra grupa strādā pie sava galda.



Zemāk redzamajā attēlā katra virsotne (bumbiņa) apzīmē skolēnu, un katra krāsa apzīmē grupu.



Līnija savieno divas virsotnes tad, ja atbilstošie skolēni nevēlas strādāt vienā grupā.

Diemžēl viens galds ir salauzts, tāpēc skolēniem pieejami tikai divi galdi.

Jums jāpārlicina divi skolēni atteikties no vēlmes strādāt dažādās grupās (tādējādi mainot iepriekš redzamo attēlu), lai astoņi studenti varētu sadalīties divās grupās, nevis trīs.

Uzdevums:

Attēlā norādiet vienu līniju, kuru noņēmot, trīs grupu vietā varēs izveidot divas!

Atbilžu varianti:

Interaktīvs uzdevums. Iezīmējiet dzēšamo līniju. Drīkst iezīmēt tikai vienu līniju.

Pareizā atbilde:

Divu skolēnu pārliecināšana strādāt kopā nozīmē līnijas dzēšanu. Mums ir jāizdzēš līnija tā, lai ar divām krāsām pietiktu visu virsotņu iekrāsošanai, bet nekur divas vienas krāsas virsotnes nebūtu savienotas ar līniju.

Vienīgā iespēja ir izdzēst līniju, kas zemāk atzīmēta ar oranžu krāsu.



Pēc šīs līnijas dzēšanas mēs varam izkrāsot attēla virsotnes izmantojot tikai divas krāsas, kā parādīts zemāk.



Lai pārbaudītu, vai šīs līnijas dzēšana ir vienīgā iespējamā izvēle, vispirms aplūkosim trijstūri attēla augšējā labajā stūrī:



Ja tiek dzēsta kāda līnija ārpus šī trijstūra, mums joprojām ir vajadzīgas trīs krāsas tikai trijstūra trīs virsotnēm.

Tagad aplūkosim piecu virsotņu ciklu apakšā.



Ja tiek dzēsta kāda līnija ārpus šī piecstūra, tad šis cikls paliek neskartas, un to nav iespējams izkrāsot tikai ar divām krāsām.

Ja mēs to izmēģinām iziet šo ciklu pulksteņrādītāja virzienā, tad abām krāsām ir pārmaiņus jāmainās. Bet, kad sasniegsim pēdējo virsotni, tai būs tāda pati krāsa kā pirmajai virsotnei, jo ciklā ir nepāra skaits virsotņu.

Tāpēc mums ir jāizdzēš līnija, kas vienlaikus izjauc gan trijstūri, gan piecu virsotņu ciklu apakšā, novedot mūs pie vienīgās iespējamās atbildes.

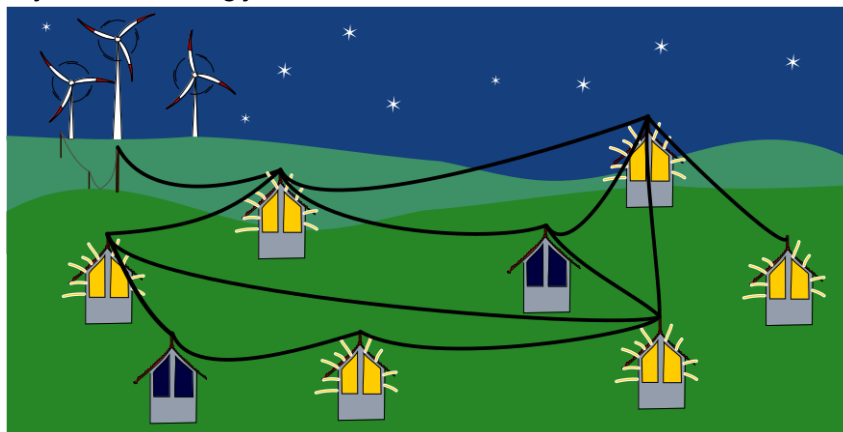
Atbildes izskaidrojums:

Daudzas reālās pasaules problēmas var formulēt kā grafa krāsošanu. Šajā uzdevumā tas ir grafs, kurā virsotnes apzīmē skolēnus un šķautne starp divām virsotnēm parāda, ka atbilstošie skolēni nevēlas strādāt kopā vienā grupā. Ja mēs krāsojam virsotnes k krāsās, tad to var uzskatīt par katra skolēna iekļaušanu kādā no k grupām. Šādas krāsojums būs *pareizs*, ja jebkurām divām virsotnēm, kas tieši savienotas ar malu, būs atšķirīgas krāsas. Bieži vien mēs vienkārši sakām *krāsojums*, kad patiesībā domājam *pareizu krāsojumu*. Šķautni sauc par *kritisku*, ja, izdzēšot to, krāsošanai pietiek ar mazāku krāsu skaitu. Šajā uzdevumā tas nozīmē, ka, ja divi atbilstošie studenti pārdomās un piekritīs strādāt kopā, tad pietiks ar mazāku grupu skaitu.

Defektoskopija

Ungārija

Bebru pilsētā elektroenerģiju ražo vēja ģeneratori kalnos un tā tiek nogādāta uz bebru mājām, izmantojot elektroenerģijas sadales tīklu:



Daži tīkla fragmenti ir bojāti un abās mājās ar izslēgtām gaismām vairs nav elektrības! Visās citās mājās elektrība ir.

Elektrību no mājas uz māju var pievadīt pa sadales tīklu jebkurā virzienā.

Uzdevums:

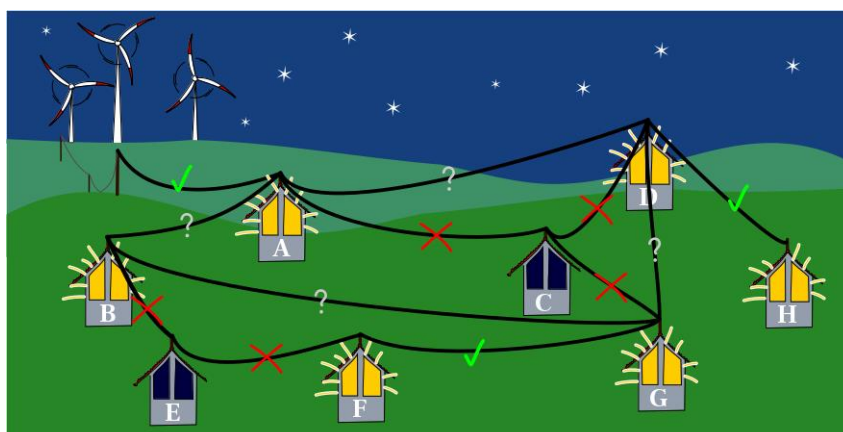
Interaktīvs uzdevums. Atbilstoši māju pašreizējam stāvoklim par katru fragmentu šajā elektroenerģijas sadales tīklā norādiet, vai (1) tas ir bojāts (X), vai (2) ir zināms, ka tas darbojas (✓) vai (3) bez papildu informācijas nevar pateikt, vai tas ir kļūdaini vai darbojas (?). Noklikšķiniet uz katra fragmenta (ja nepieciešams, vairākas reizes), lai norādītu tā stāvokli.

Atbilžu varianti:

Parādiet attēlu ar iepriekš redzamo karti versijā, kurai katram fragmentam ir pievienota maināma un noklikšķināma attēla etiķete. Sākotnēji visi fragmenti tiek parādīti kādā nenoteiktā stāvoklī. Noklikšķinot uz tiem, tie liek aplūkot trīs stāvokļus, kurus var izvēlēties atbildei: (1) parādot kā kļūdainu, (2) parādot kā strādājošu, (3) parādot kā "nav iespējams pateikt".

Pareizā atbilde:

Šeit ir karte, kas parāda to, ko mēs zinām par elektrības sadales tīkla fragmentiem:



Pirmā lieta, ko mēs zinām, ir tas, ka divi tiešie fragmenti uz māju E un trīs tiešie fragmenti uz māju C ir bojāti. Tā kā visām kaimiņu mājām ir elektrība, ja darbotos jebkurš no šiem trim fragmentiem būtu nodrošinājis elektrību arī mājām C un E.

Fragmenti, kas vienatnē nodrošina elektrību mājām, kurās ir ieslēgtas gaismas, nevar būt bojāti, pretējā gadījumā tur nevarētu nonākt elektrība. Tas attiecas uz fragmentu, kas ved uz māju H, un fragmentu no mājas G uz māju F. Arī fragmentam no vēja ģeneratoriem uz māju A ir jādarbojas, pretējā gadījumā vispār nevienam nebūtu elektrības.

Pārējās mājas B, G un D ir vairākkārt savienotas ar māju A. Piemēram, B var saņemt elektroenerģiju tieši no A, bet var saņemt arī no G, ja savienojums ar A ir bojāts. To pašu var teikt par D.

Atbildes izskaidrojums:

Datortīklos, tāpat kā elektrības sadales tīklos, dažas saites var būt kļūdainas-lēnas, pārslogotas vai pilnīgi pārautas. Izmantojot dublējošas saites tīklā tiek nodrošināta tā nepārtraukta pieejamība defektu gadījumā (ar nosacījumu, ka nav pārāk daudz kļūdu vienlaikus).

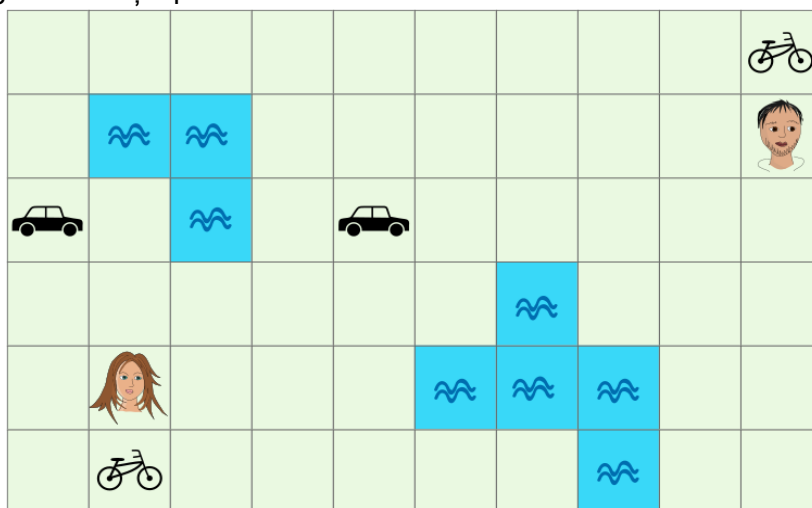
Lai attēlotu tīkla struktūru, datorzinātnieki parasti izmanto *grafus*. Pastāv daudz algoritmu darbam ar grafiem, lai, piemēram, pēc iespējas efektīvāk noteiktu kļūdainu saiti, ņemot vērā tīkla struktūru.

Kļūdu labošana sistēmā ir uzdevums, kas datorzinātniekiem ļoti bieži ir jā dara ne tikai datortīklos, bet arī programmatūras izstrādē. Lai labotu kļūdu, ir jānosaka precīzs tās avots, un šis process parasti tiek veikts pakāpeniski vairākos posmos. Daži programmētāji uzskata, ka jūs nekad nevarat atrast visas programmas kļūdas un kļūdas.

Steidzamā satikšanās

Lietuva

Diviem draugiem steidzami jāsatiekas (skat. zemāk esošo karti). Viņi var pārvietoties no viena blakus esoša kvadrāta uz otru (horizontāli vai vertikāli) tieši vienas minūtes laikā. Ja viņi sasniedz kvadrātu, kurā atrodas velosipēds vai automašīna, viņi to var izmantot, lai pārvietotos ātrāk - ar velosipēdu (divus kvadrātus vienā minūtē) vai ar automašīnu (piecus kvadrātus vienā minūtē). Draugi nevar ceļot pa ūdeni.



Uzdevums:

Kāds ir mazākais iespējamais minūšu skaits, kas nepieciešams, lai satiktos vienā (jebkurā) kvadrātā?

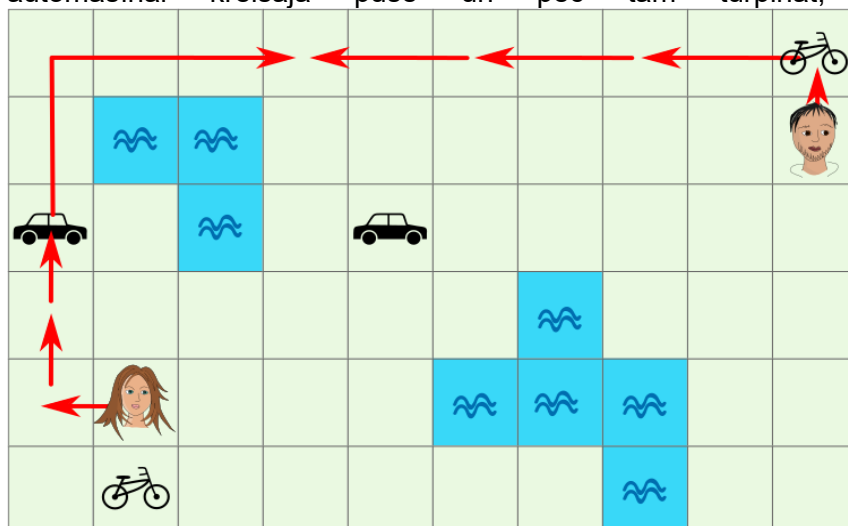
Atbilde:

Ievadīt mazāko nepieciešamo minūšu skaitu kā veselu skaitli.

Pareizā atbilde: 4

Pareizā atbilde ir 4 minūtes. To var panākt, izmantojot zemāk redzamo maršrutu:

(Vēl viena iespēja ir doties uz velosipēdu kreisajā pusē un braukt ar velosipēdu līdz automašīnai kreisajā pusē un pēc tam turpināt, kā norādīts iepriekš.)



Lai pamatotu, kāpēc nepietiek ar trim minūtēm, var rīkoties šādi:

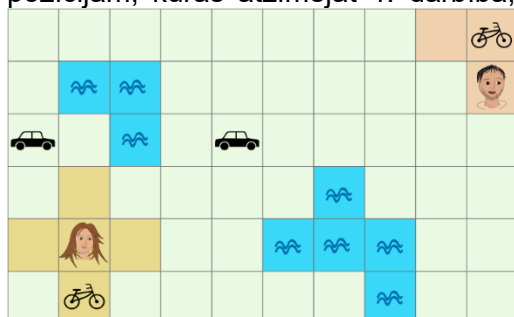
- Lai gan trīs minūšu laikā jūs varat sasniegt automašīnu kreisajā pusē, neatliek laika, lai ar to jebkur aizbrauktu. Un nevienu automašīnu otra persona (kas atrodas labajā pusē) nevar sasniegt trīs minūšu laikā. Tātad automašīnas nav noderīgas, un mēs varam tās noņemt no kartes.
- Abi draugi ir vairāk nekā piecu minūšu attālumā viens no otra, ja pārvietotos tikai ar kājām, tāpēc kādam no viņiem ir nepieciešams velosipēds. Patiesībā velosipēds ir vajadzīgs *abiem*, jo tos šķir vairāk nekā 9 pozīcijas. Bet katra velosipēda sasniegšanai nepieciešama vienu minūti, un, kad ir atlikušas tikai divas minūtes, viņi nevar sasniegt viens otru pat izmantojot velosipēdus.

Atbildes izskaidrojums:

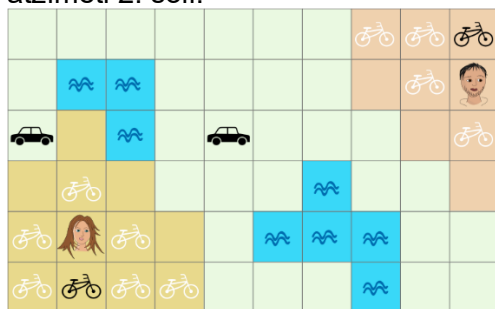
Kā jūs risinājat šo uzdevumu? Vai nejausi atradāt īsu maršrutu un cerējāt, ka īsāku maršrutu nevarēs atrast, vai arī izmēģinājat desmitiem dažādu iespēju un atcerējāties īsāko laiku?

Datorprogramma, kas paredzēta šāda veida uzdevumiem, izmantotu sistemātisku pieeju, visticamāk, izmantojot algoritmu, ko sauc par **“meklēšanu plašumā”** jeb **platkursiju**. Šim uzdevumam tas izskatītos šādi:

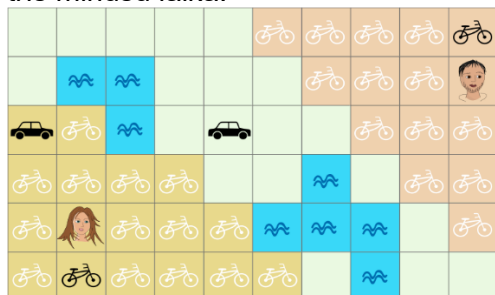
1. Kartē atzīmējiet visus kvadrātus, kurus katrs draugs var sasniegt vienas minūtes laikā.
2. Atzīmējiet visus kvadrātus, kurus var sasniegt ne vairāk kā vienas minūtes laikā no pozīcijām, kuras atzīmējāt 1. darbībā, un sekojiet līdzi, kāda veida transportu izmantojāt.



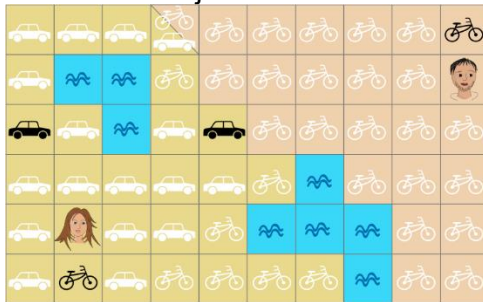
Atzīmējiet visus kvadrātus, kurus var sasniegt vienas minūtes laikā no kvadrātiem, kas tika atzīmēti 2. solī.



Tā kā abas mūsu atzīmētās zonas nepārklājas, jūs redzat, ka draugi nevar sasniegt viens otru trīs minūšu laikā.



Veiciet vēl vienu soli: atzīmējiet visus kvadrātus, kurus var sasniegt vienas minūtes laikā no 3. solī atzīmētajiem kvadrātiem.



Tagad divi mūsu atzīmētie reģioni pārklājas (vienā kvadrātā), norādot, ka **pēc četrām minūtēm** abi draugi šajā kvadrātā var satīkties.

Jūs droši vien esat iepazinies ar programmatūru, kas atrod ātrāko maršrutu starp divām vietām kartē, izmantojot tikai ceļus, nevis braucot pa kalniem un upēm. Šis uzdevums ir ļoti līdzīgs, bet tagad divas personas virzās viena pretī otrai, nevis viena persona uz fiksētu vietu.

Tā kā risinājuma meklēšana tiek veikta sistemātiski, dators bieži atrod risinājumus, kas sākumā nav acīmredzami - dažreiz apvedceļš ar mazāk luksoforiem var būt labāks risinājums nekā tiešs ceļš vai sabiedriskā transporta maršruts ar vairākkārtēju ātru pārsēšanos var izrādīties ātrāks par tiešo autobusu.

Datorzinātnē ir zināmas dažādas metodes, lai atrastu šādu uzdevumu labāko risinājumu. Papildus iepriekš aprakstītajai meklēšanai plašumā ir arī **zaru un robežu** algoritmi, kas darbojas līdzīgi, bet meklēšanu paātrina pamatoti īsceļi: ja mēs jau esam atraduši diezgan labu risinājumu, mēs varam atmett iespējas kas, kā mēs jau zinām, nevar dot labāku risinājumu par līdz šim labāko atrasto.

Ja problēma kļūst pārāk sarežģīta, pat ātram datoram būs nepieciešams ilgs laiks, lai atrastu vislabāko atrisinājumu. Un praksē bieži vien pietiek atrast ļoti labu atbildi, pat ja tā nav labākā iespējamā. (Ja jūs varat sasniegt galamērķi 78 minūtēs, jūs, iespējams, īpaši neuztrauc tas, ka varētu būt cits maršruts, kam būtu nepieciešamas 77 minūtes.)

Viens paņēmieni, kas tiek izmantots šajā gadījumā, ir intuitīvs **rijīgs algoritms**, kas katrā solī izvēlas to turpinājumu, kas tajā brīdī šķiet optimāls, un neskatās uz priekšu, kas varētu notikt turpmākajos soļos. Mūsu uzdevumā tas nozīmētu, ka abi draugi vienmēr dodas uz kvadrātu, kas viņus satuvina. Šajā gadījumā tā nav laba stratēģija, jo tad viņi lielāko ceļa daļu veiktu kājām, patērējot daudz laika. Tomēr ir arī cita veida problēmas, kurās šī mantkārīgā stratēģija rada samērā labus risinājumus. Un šādus risinājumus var atrast daudz ātrāk nekā izmantojot citas metodes.

Krāsu loģika

Urugvaja

Džordžs savā datorā spēlē spēli "Krāsu loģika" (pazīstama arī kā "Mastermind"): dators veido paroli no četriem atšķirīgiem cipariem. Spēlētājs cenšas uzminēt paroli, veicot vairākus minējumus - ievadot četru ciparu kombinācijas. Uz katru minējumu dators atbild, paziņojot pareizo ciparu skaitu - tādu, kas parādās gan minējumā, gan parolē. Datora atbildē tiek norādīts arī tas, vai minējumā pareizie cipari atrodas pareizajās vietās. Džordžs veica vairākus minējumus (?).

| ? | ? | ? | ? | Atbilde |
|---|---|---|---|--|
| 5 | 7 | 2 | 0 | Viens no cipariem ir pareizs un tas atrodas pareizajā vietā. |
| 6 | 0 | 3 | 1 | Viens no cipariem ir pareizs, bet tas neatrodas pareizajā vietā. |
| 1 | 4 | 8 | 5 | Divi cipari ir pareizi, bet tie nav pareizajās vietās. |
| 1 | 5 | 9 | 6 | Neviens no cipariem nav pareizs. |
| 8 | 1 | 2 | 5 | Viens no cipariem ir pareizs, bet tas neatrodas pareizajā vietā |

No iegūtajām atbildēm viņam izdevās atklāt paroli.

Uzdevums:

Kāda ir parole?

Atbilžu variant:

Kā risinājums jāiesniedz četru ciparu virkne - uzminētā parole.

Pareizā atbilde: 3748

Pareizā atbilde ir 3748. Lai atrastu šo kombināciju, mums sistemātiski jāanalizē pieejamā informācija un iteratīvi jācenšas vai nu atrast pareizos ciparus un to atrašanās vietas, vai arī jācenšas izslēgt nederīgos. Viens no risināšanas veidiem ir mēģināt vispirms atrast ciparus un pēc tam pareizu to secību.

Turklāt var ņemt vērā, ka minējumi ir savstarpēji neatkarīgi - to secība nav svarīga.

No ceturtā minējuma varam izsecināt, ka parolē nav ciparu 1, 5, 6 un 9.

Tad no trešā minējuma ir skaidrs, ka parolē noteikti ir cipari 4 un 8. Tad no piektā minējuma ir skaidrs, ka parolē nav cipara 2.

Varam paskatīties uz atbildēm ņemot vērā izslēgtos ciparus (aizstāti ar #):

| ? | ? | ? | ? | Atbilde |
|---|---|---|---|--|
| # | 7 | # | 0 | Viens no cipariem ir pareizs un tas atrodas pareizajā vietā. |
| # | 0 | 3 | # | Viens no cipariem ir pareizs, bet tas neatrodas pareizajā vietā. |
| # | 4 | 8 | # | Divi cipari ir pareizi, bet tie nav pareizajās vietās. |
| # | # | # | # | Neviens no cipariem nav pareizs. |
| 8 | # | # | # | Viens no cipariem ir pareizs, bet tas neatrodas pareizajā vietā |

Zinām, ka paroli veido cipari 4 un 8, kā arī divi cipari no 0, 3 un 7. Tā kā vienlaikus parolē nevar atrasties 0 un 7 (pirmais minējums), kā arī 0 un 3 (otrais minējums), tad derīgs ir tikai ciparu komplekts 3, 4, 7, 8.

No pirmā minējuma seko, ka 7 atrodas otrajā pozīcijā. Tātad 8 atrodas ceturtajā pozīcijā (jo 1. un 3. nevar saskaņā ar trešo un piekto minējumu).

Tālāk viegli izsecināt, ka 3 atrodas 1., bet 4 - 3. pozīcijā.

Gala rezultāts - parole ir 3748.

Atbildes izskaidrojums:

Šis uzdevums ir balstīts uz spējas loģiski domāt, kā arī varētu būt saistīts ar atpakaļejoša algoritma piemērošanu.

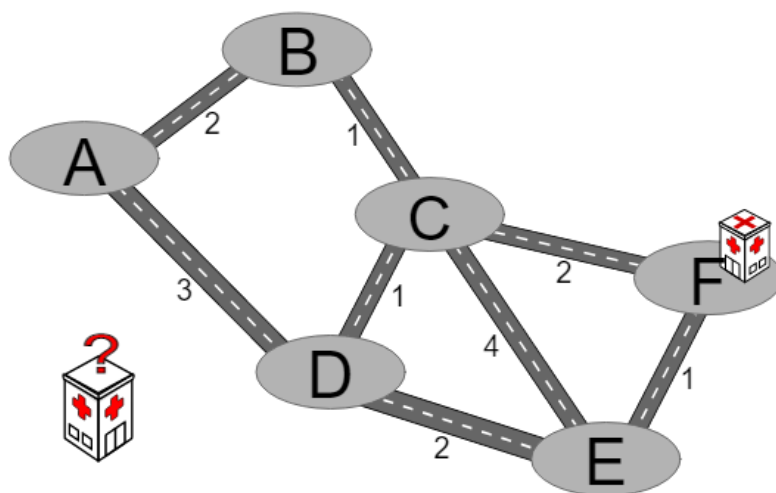
Loģikai ir nepārvērtējama loma praktiski jebkurā datorzinātņu jomā (datu bāzēs, skaitļošanas sarežģītības noteikšanā, programmēšanas valodās, mākslīgajā intelektā, aparatūras un programmatūras projektēšanā un verificēšanā utt.), Un tas neapšaubāmi ir viens no pamatiem, kas nodrošina briedumu un veiklību, lai asimilētu nākotnes datorzinātnes jēdzienus, valodas, paņēmienus utt.

Ir daži sarežģīti algoritmi, kas izmanto līdzīgu loģisku spriedumu virkni, lai atklātu iebrucējus tīklos vai kļūdas loģiskajās shēmās. Tā kā var būt grūti pārbaudīt katru daļu atsevišķi, bieži tiek izmantota testa jautājumu kopa, un, pamatojoties uz atbildēm, iespējams noteikt iebrucēja/kļūdas klātbūtni.

Reģionālās slimnīcas

Beļģija

Reģionā ir sešas pilsētas. Tikai divās pilsētās var būt slimnīca. Kad cilvēkiem nepieciešama klīniska aprūpe, viņi pēc iespējas ātri jānogādā slimnīcā. Zemāk redzamajā zīmējumā attēlotas reģiona pilsētas, kā arī laiks stundās, kas vajadzīgs, lai no vienas pilsētas nokļūtu otrā. Dažreiz, lai sasniegtu kādu pilsētu, nepieciešams braukt cauri vienu vai vairākām citām pilsētām.



Abu slimnīcu atrašanās vieta jāizvēlas tā, lai cilvēki pēc iespējas īsākā laikā varētu nokļūt kādā no tām. Tas nozīmē, ka ilgākajam laikam, lai no jebkuras pilsētas nokļūtu slimnīcā, jābūt pēc iespējas mazākam.

Uzdevums:

Viena slimnīca atrodas pilsētā F. Kurā pilsētā jāatrodas otrai slimnīcai?

Atbilžu varianti:

- A. Pilsēta A.
- B. Pilsēta B.
- C. Pilsēta C.
- D. Pilsēta D.
- E. Pilsēta E.

Pareizā atbilde: B

Tā kā viena slimnīca atrodas pilsētā F, ir piecas iespējas, kur atrasties otrai. Katrai iespējai mums ir jāpārbauda minimālais laiks, kas nepieciešams, lai no katras pilsētas tiktu uz kādu slimnīcu. Nākamajā tabulā kolonnas parāda visus iespējamus divu slimnīcu novietojumus, bet rindas - stundu skaitu, kas nepieciešams, lai sasniegtu tuvāko slimnīcu.

| Minimālais laiks no pilsētas ... | Īpašo slimnīcu atrašanās vietas | | | | |
|----------------------------------|---------------------------------|----------|----------|----------|----------|
| | A un F | B un F | C un F | D un F | E un F |
| A | 0 | 2 | 3 | 3 | 5 |
| B | 2 | 0 | 1 | 2 | 3 |
| C | 2 | 1 | 0 | 1 | 2 |
| D | 3 | 2 | 1 | 0 | 2 |
| E | 1 | 1 | 1 | 1 | 0 |
| F | 0 | 0 | 0 | 0 | 0 |
| Maksimālais laiks | 3 | 2 | 3 | 3 | 5 |

Kā parādīts pēdējā rindā, lielākais stundu skaits, kas nepieciešams slimnīcas sasniegšanai, otras slimnīcas izvietojuma pilsētā B gadījumā ir labākais risinājums. Tad no katras pilsētas var nokļūt slimnīcā ne vairāk kā 2 stundu laikā.

Atbildes izskaidrojums:

Šis uzdevums ir saistīts ar grafa virsotņu k -centra atrašanās uzdevumu. Tas nozīmē, ka svarotā grafā (šķautnēm ir svārs) jāizvēlas k virsotnes tā, lai lielākais attālums no jebkuras virsotnes līdz kādai no izvēlētajām k virsotnēm būtu mazākais iespējams. Šajā uzdevumā k vērtība ir 2 un grafa virsotnēm atbilst pilsētas, bet svarotajām šķautnēm - ceļi starp pilsētām ar noteiktu ceļā pavadāmo laiku.

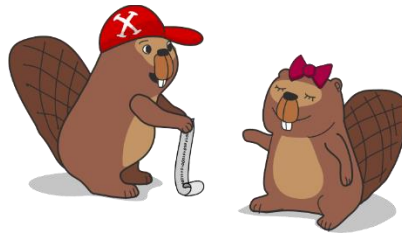
Šāds uzdevums ir ļoti izplatīts, mēģinot izvēlēties, kur izvietot tādas būves kā ugunsdzēsības depo, skolas, policijas iecirkņi utt. Vai slimnīcas, kā šajā uzdevumā. Lielums, kas šajos uzdevumos jāsamazina, var būt laiks, kas nepieciešams, lai sasniegtu šos objektus, vai attālums, vai vēl kāds cits kritērijs.

Kompaktais ziņojums

Čehija

Bebrs Harijs vēlas aizšifrēt ziņojumu, aizstājot tajā esošos burtus ar binārajiem cipariem 1 un 0. Viņš zina, ka parasti tekstos burti T un E parādās biežāk, tādēļ viņš šos burtus ir nolēmis aizkodēt ar īsāku virkni. Harijs burtus T, E, A, K, C un R kodēs šādi:

| Burts | T | E | A | K | C | R |
|-------|---|----|------|------|------|------|
| Kods | 1 | 00 | 0010 | 0110 | 1010 | 1110 |



Harijs nosūtīja Ivetai kodētu ziņojumu :

1 0 0 1 0 0 1 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 0

Iveta jau ir noskaidrojusi, ka šis ziņojums beidzas ar burtu E.

Uzdevums:

Kāds ir pilnais Harija ziņojuma teksts?

Atbilde:

Ieraksti burtu virkni - pilno ziņojuma tekstu.

Pareizā atbilde: TAKECARE

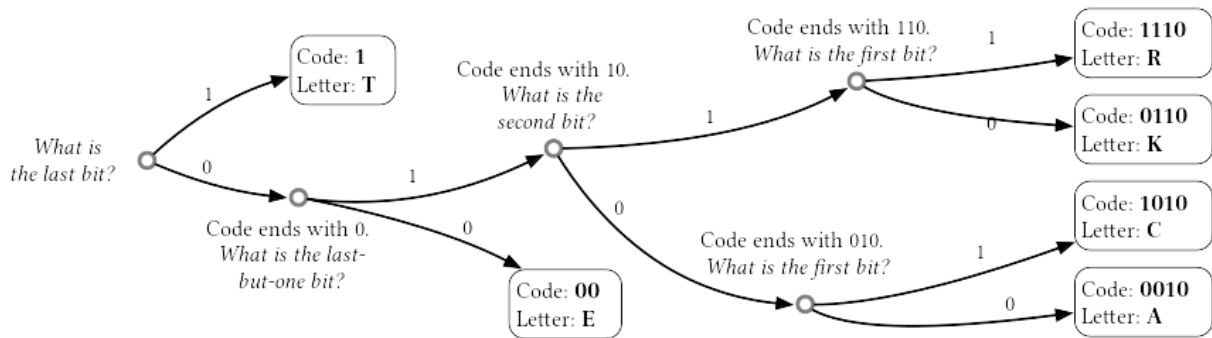
Šeit ir burtu un to binārā attēlojuma atbilstība Harija ziņojumā:

| Burts | T | A | K | E | C | A | R | E |
|-------|---|------|------|----|------|------|------|----|
| Kods | 1 | 0010 | 0110 | 00 | 1010 | 0010 | 1110 | 00 |

Lai atjaunotu ziņojumu, jāatrod veids, kā visu ziņojumu sadalīt pa burtiem. Ja mēs sākam no kreisās puses, to nav tik vienkārši izdarīt: mēs saskarsimies ar iespējamiem turpinājuma variantiem. Mēs varam diezgan viegli noteikt, ka pirmais burts ir T un atbilst "1", bet otrais burts var būt vai nu E, ko apzīmē ar "00", vai A, ko apzīmē "0010". Šobrīd mēs nevaram droši zināt, kura versija ir pareiza. Tomēr vēstījums ir nepārprotams: ja mēs izdarīsim nepareizu izvēli otrajā pozīcijā, tad vēlāk iestrēgsim un sapratīsim, ka pareiza bija otrā iespēja.

Mūsu gadījumā mēs varam saprast, ka, ja mēs sākam no beigām, mums nekad nav "jāizdomā", kā dekodēt burtu. Tas ir tāpēc, ka kodam nav *sufiksu*: nav neviena koda vārda, kas beidzas ar to pašu 1 un 0 secību, kas pats ir cits koda vārds. Tādējādi tekstu var viegli nepārprotami rekonstruēt, lasot bināro kodu no labās uz kreiso pusi. Kad burta kods ir atrasts, kodu var apmainīt pret burtu.

Zemāk redzamā diagramma parāda, kā bināro ziņojumu var nolasīt no labās uz kreiso pusi, viennozīmīgi iegūstot burtus:



Ja mēs būtu gribējuši šo ziņojumu nepārprotami atšifrēt ejot no kreisās puses uz labo, mums būtu vajadzīgs *prefiksbrīvs kods*, t.i., kods kur neviens kods nesākas ar tādu 1 un 0 secību, kas pats par sevi ir kāda burta kods. Harija kods nav prefiksbrīvs, jo burta A kods “0010” sākas ar “00”, kas pats par sevi ir E kods .

Atbildes izskaidrojums:

Visi objekti, ar kuriem dators strādā, ir jāapraksta kā bitu secība. Tas attiecas arī uz tekstiem. Vienmēr tiek sagaidīts, ka sākotnējo objektu var rekonstruēt no tā binārā attēlojuma, taču tas ir iespējams tikai tad, ja nekad nenotiek tā, ka diviem vai vairākiem dažādiem objektiem ir vienāds binārais attēlojums. Datorzinātniekiem nepieciešams izstrādāt tādas kodu sistēmas, lai būtu iespējams efektīvi rekonstruēt sākotnējo objektu (piemēram, tekstu) no tā binārā attēlojuma.

Ja kāds vēlas saspiest tekstu (iegūt pēc iespējas īsāku teksta bināro attēlojumu), tad laba stratēģija ir ņemt īsākus bināros kodus biežāk sastopamajiem burtiem un izmantot garāku kodu tiem burtiem, kas parādās reti. Šajā gadījumā ir jā rūpējas, lai pareizi izvēlētos kodus, kas garantē nepārprotamu dekodēšanu (oriģinālā teksta rekonstrukcija). Šim nolūkam ļoti laba izvēle ir kodi bez prefiksiem un bez sufiksiem, kuru principi ir aprakstīti iepriekš sniegtajā atbildes skaidrojumā.

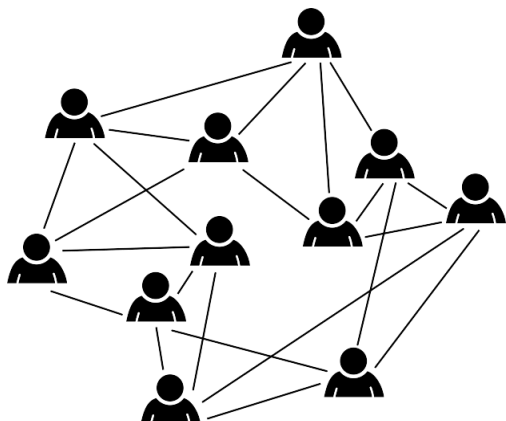
Revīzijas komiteja

Čehija

Bebras pilsētas domē ir 11 deputāti. Deputāti savā starpā var būt *saistīti* kā darba kolēģi, vienas ģimenes locekļi, biznesa partneri vai vienas politiskās partijas biedri.

Katrus divus deputātus, kas ir savā starpā saistīti, diagrammā savieno līnija.

Domes revīzijas komitejā drīkst darboties tikai tādi deputāti, kas savā starpā nav saistīti.



Uzdevums:

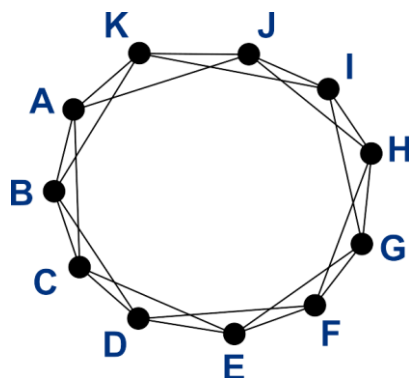
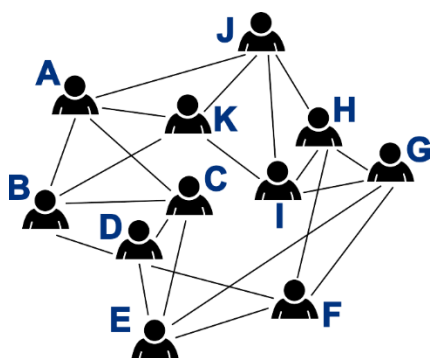
Kāds ir lielākais deputātu skaits, kas var darboties revīzijas komitejā?

Atbilžu varianti:

- A. 2
- B. 3
- C. 4
- D. 5

Pareizā atbilde: B

Lielākas ērtības labad deputātiem atbilstošos punktus apzīmēsim ar burtiem (attēls zemāk kreisajā pusē). Mēs varam saskaitīt, ka no katra punkta iziet četras līnijas. Labāk tas ir redzams, pārkārtojot grafu (attēlā pa labi).



Piemēram, deputāts C ir saistīts tikai ar deputātiem A, B, D, E.

Revīzijas komiteju mēs varam sākt izvēlēties jebkurā vietā, jo grafs ir simetrisks - tam nav īpašu punktu.

Ja izvēlamies deputātu A, nākamais tuvākais deputāts, kas nav saistīts ar A pretēji pulksteņrādītāja virzienam ir D. Nākamais, kas nav saistīts ne ar A, ne D, ir G. Nākamais kandidāts būtu J, bet J ir attiecības ar A, tāpēc J nevar būt revīzijas komitejā. Tikai 3 deputāti (šajā gadījumā A, D, G) nav savā starpā saistīti.

Mēs iegūsim vienādu revīzijas komitejas locekļu skaitu, sākot ar jebkuru deputātu (punktu grafā) un virzoties jebkurā virzienā, tāpēc revīzijas komitejā nevar būt vairāk par trim locekļiem.

Atbildes izskaidrojums:

Attiecības starp pilsētas domes deputātiem tiek modelētas ar *grafu*. Tas sastāv no *virsoņiem* (kas atbilst personām/objektiem un parasti tiek attēlotas kā punkti), kas pa pāriem savienotas ar *šķautņiem* (parasti tiek attēlotas kā līnijas, kas savieno punktus). Gan uzdevuma diagramma, gan skaidrojuma diagramma attēlo vienu un to pašu grafu.



Grafs ir abstrakta struktūra un ir noderīgs, ja vēlaties koncentrēties uz sakarībām: grafā ir uzsvērtas svarīgās iezīmes (kurš ir ar ko saistīts), bet tiek izlaistas nebūtiskas detaļas (kas ir iesaistītās personas/objekti, kādas tieši attiecības tos saista).

Datori var ļoti efektīvi strādāt ar grafiem, tāpēc datorzinātniekiem ļoti labi jāizprot grafi, to veidi un īpašības.

Izmantojot grafu terminoloģiju, šajā uzdevumā *grafā* ir jāatrod *kodols* - lielākā iespējamā *neatkarīgu virsoņu kopa*, t.i., tāda diagrammas virsoņu apakškopa, kurā nekādas divas virsoņnes nav savienotas ar šķautni. Precīzāk, šeit mēs meklējam grafa *virsoņu neatkarības skaitli*, kas ir lielākās neatkarīgu virsoņu kopas lielums.

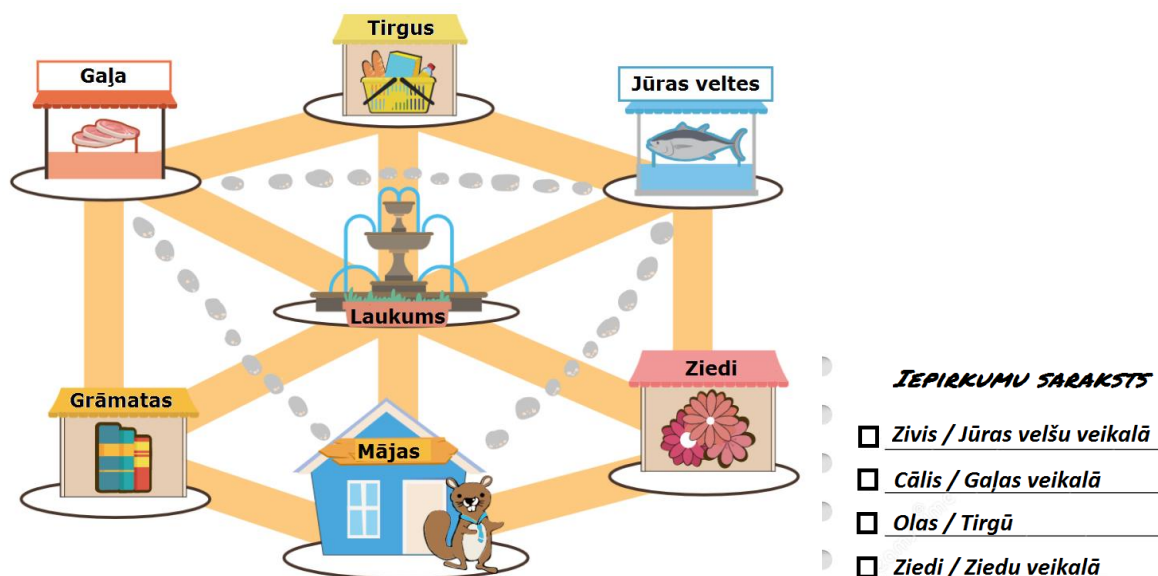
Iepirkšanās

Taivāna

Attēlā zemāk ir dota bebru ciemata karte, kurā dzīvo Bebru ģimene. Lai pārietu no vienas ēkas uz otru, Bebri iet pa dubļainu  vai akmeņainu taku .

Pastaiga starp divām ēkām Mazajam Bebram, ejot pa dubļainu taku aizņem 5 minūtes, bet ejot pa akmeņainu taku - 8 minūtes.

Piemēram, Mazajam Bebram ir nepieciešamas 5 minūtes, lai dotos no laukuma uz ziedu veikalu vai grāmatnīcu, un 8 minūtes, lai atgrieztos mājās no jūras velšu vai gaļas veikala.



Bebru māte lūdz Mazo Bebru nopirkt pārtiku. Iepirkumu saraksts ir parādīts iepriekšējā zīmējumā. Mazajam Bebram jāsāk ceļojums no mājām, jānopērk visas iepirkumu sarakstā minētās preces un jāatgriežas mājās. Papildus, lai zivis būtu svaigas, Mazajam Bebram jūras velšu veikals jāapmeklē tieši pirms došanās mājās.

Uzdevums:

Kāds ir mazākais iespējamais laiks, kāds Mazajam Bebram nepieciešams pārtikas preču iegādei?

Atbilde:

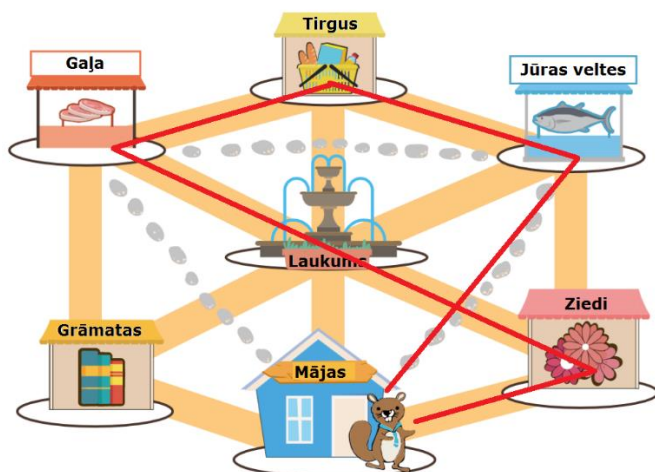
Ievadiet veselu skaitli robežās no 25 līdz 43 - mazāko nepieciešamo iepirkšanās laiku minūtēs.

Pareizā atbilde: 33

Pieņemsim, ka jūs esat Mazais Bebrs, kuram jāiet iepirkties un atgriezties mājās. Lai zivis būtu svaigas, jūras velšu veikalam vajadzētu būt pēdējai pieturai pirms došanās mājās. Tāpēc sākotnējo uzdevumu var sadalīt divos secīgos apakšuzdevumos: (1) nopirkt trīs, no zivīm atšķirīgās, preces, (2) nopirkt zivis un doties mājās. Vismazākais laiks, kas nepieciešams katram apakšuzdevumam, ir parādīts zemāk:

| (1) iepirkties T irgū, G aļas un Z iedu veikalā | | (2) iepirkties J ūras velšu veikalā un doties M ājās | | Kopējais laiks (min) |
|--|------------------------|--|------------------------|----------------------|
| Maršruts | minimālais laiks (min) | Maršruts | minimālais laiks (min) | |
| M-G-Z-T | 28 | T-J-M | 13 | 41 |
| M-G-T-Z | 23 | Z-J-M | 13 | 36 |
| M-Z-T-G | 20 | G-J-M | 16 | 36 |
| M-Z-G-T | 20 | T-J-M | 13 | 33 |
| M-T-G-Z | 25 | Z-J-M | 13 | 38 |
| M-T-Z-G | 30 | G-J-M | 16 | 46 |

Sadalot uzdevumu, mēs varam izpētīt visus iespējamus maršrutus, lai izpildītu katru apakšuzdevumu. Un, aprēķinot minimālo laiku, kas nepieciešams katra apakšuzdevuma izpildei, mēs savukārt varam iegūt minimālo laiku, kas nepieciešams visa uzdevuma pabeigšanai. Balstoties uz iepriekš redzamo tabulu, Mazajam Bebram ir nepieciešams izvēlēties maršrutu, kas parādīts attēlā:



Atbildes izskaidrojums:

Šajā uzdevumā mēs varam atrast īsāko maršrutu, to uzzīmējot. Tomēr reālajā dzīvē, kad 15 iespējamo taku vietā ir tūkstošiem taku, mums īsākā ceļa atrašanai būtu nepieciešama tādu tehnoloģiju palīdzība, kā GPS un maršrutēšanas programmatūra.

Datorzinātnēs grafi ir izplatīts veids, kā parādīt attiecības datu starpā. Grafus var izmantot, lai attēlotu saites starp objektiem. Tā ir metode, kā attēlot attiecības starp lietām izmantojot virsotnes un šķautnes. Diagrammas arī atvieglo attiecību (parasti apzīmē kā šķautnes) aprakstīšanu starp sarežģītu konceptu būtiskajiem jēdzieniem (parasti apzīmē kā virsotnes).

Grafu teorijā īsākā ceļa uzdevuma mērķis ir atrast maršrutu ar īsāko kopējo attālumu starp divām virsotnēm grafā. Dažreiz dažādiem ceļiem ir atšķirīgs svars, tāpēc tas jāņem vērā aprēķinot kopējo attālumu. Šajā gadījumā īsākā ceļa problēma tiek atrisināta, atrodot minimālo svērto attālumu.

Garākais fragments

Urugvaja

16 simbolu virkne izveidota izmantojot četrus veidu simbolus:



Šajā virknē tieši trīs simbolus drīkst nomainīt pret citiem.

Uzdevums

Kāds pēc simbolu nomaiņas ir garākais iespējamais vienādu simbolu fragments (vienādu secīgu simbolu apakšvirkne)?

Atbilžu varianti

- A) 4
- B) 5
- C) 6
- D) 7

Pareizā atbilde: 6

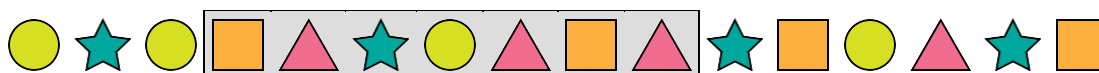
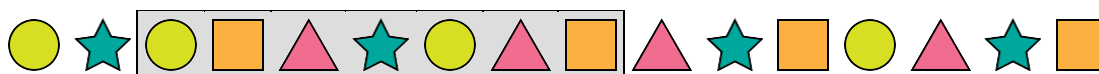
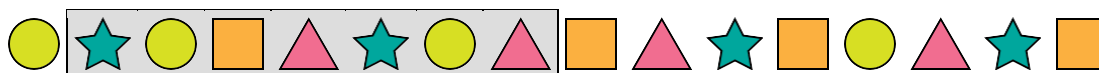
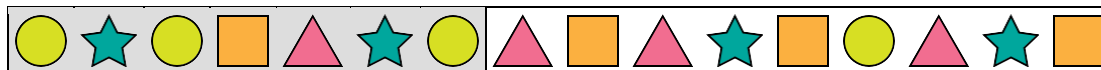
Pareizā atbilde ir 6. Lai to parādītu, mums ir jāpierāda divas lietas: (1) ka ir iespējams fragments garumā 6, un (2), ka nav iespējams fragments, kura garums pārsniedz 6.

Pirmo daļu ir viegli pierādīt. Lūk, kā var izveidot 6 trijstūru fragmentu:



Lai pierādītu, ka nav iespējams fragments, kura garums ir lielāks par 6, aplūkosim jebkuru fragmentu garumā 7. Tā kā drīkst mainīt tikai trīs simbolus, kādā fragmentā garumā 7 jau sākumā jābūt četriem vienādiem objektiem.

Sākotnējā virknē ir desmit fragmenti ar garumu 7, no kuriem daži ir parādīti zemāk. Neviens no šiem fragmentiem nav četru vienādu objektu (pārbaudiet to arī attiecībā uz šeit neparādītajiem fragmentiem!).



Tā kā nav iespējams izveidot fragmentui, kura garums ir 7, tad noteikti nav iespējams, ka eksistē fragments, kura garums pārsniedz 7.

Tādējādi mēs esam parādījuši, ka visgarākā vienādu simbola fragmenta garums ir 6.

Atbildes izskaidrojums

Šis uzdevums ir saistīts ar *garākā fragmenta*, kas atbilst noteiktiem kritērijiem, atrašanu.





Informātikā ir daudz gadījumu, kad ir lietderīgi atrast garāko fragmentu, jo īpaši atrast garāko *kopīgo* divu virkņu fragmentu..

Garākā kopīgā fragmenta atrašana palīdz atklāt plaģiātus un saspiest datus, noņemot atkārtotošās datu daļas.

Garāko fragmentu atrašanai izmanto divu rādītāju un bīdāmā loga metodes.

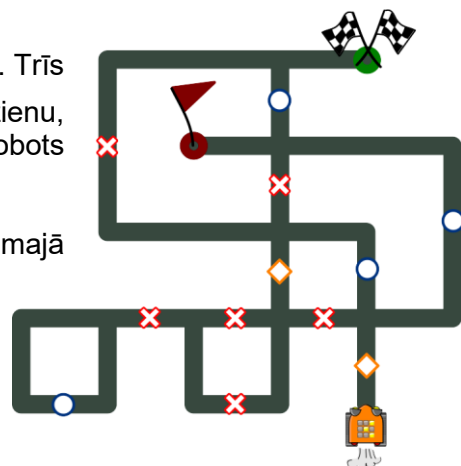
Simbolu robots

Čehija

Robots sāk no parādītās pozīcijas un pārvietojas pa līnijām. Trīs simboli ,  un  izvietoti uz līnijām, un nosaka virzienu, kādā robotam vajadzētu virzīties nākamajā krustojumā. Robots nedrīkst sasniegt .

Katram simbolam ir atšķirīga nozīme, un tas nākamajā krustojumā var likt:

- **pagriezties pa kreisi**, vai
- **pagriezties pa labi**, vai
- **doties taisni**.



Diemžēl mēs nezinām, kurš simbols ko nozīmē.

Simbola nozīme paliek nemainīga neatkarīgi no robota kustības virziena.

Attēlā redzamā bultiņa parāda, kā robots pagrieztos divās dažādās situācijās, ja trīsstūra






simbols nozīmētu "pagriezties pa kreisi".

Uzdevums:

Palīdziet robotam sasniegt finišu , piešķirot simboliem pareizo nozīmi.

Atbilde:

Interaktivitāte: velciet un nometiet simbolus ,  un  - tie ir pārvietojami, un tos var piesaistīt tekstiem "pagriezties pa kreisi", "pagriezties pa labi" un "doties taisni", piemēram, velkot savienojošas līnijas vai pārvietojiet uz tukšajiem lauciņiem.

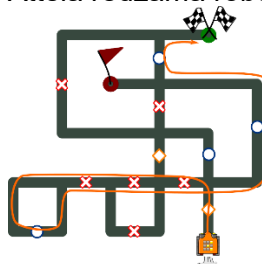
Pareizā atbilde:


 = doties taisni

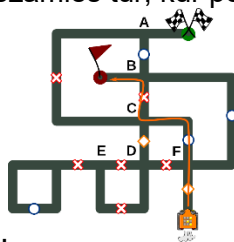
 = pagriezties pa labi

 = pagriezties pa kreisi





Attēlā redzama robota pārvietošanās.

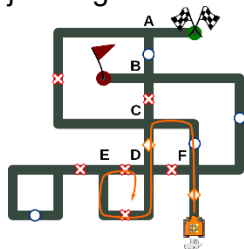











pa kreisi". Tad robots nogriezīsies pa kreisi krustojumā B un sasniegs . Tas nav tas, ko mēs vēlamies, tāpēc mēs atgriezāties tur, kur pēdējoreiz piešķīrām

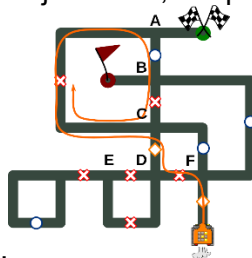


nozīmi kādam simbolam - uz krustojumu C.



- [1.2. minējums] Pieņemsim, ka robots krustojumā C nogriezīsies pa kreisi. Tad simbols  nozīmē "pagriezies pa kreisi" un simbols  nozīmē "pagriezies pa labi". Krustojuma D robots dotos taisni (saskaņā ar ) , un pēc tam sasniegtu krustojumu E, kurā nogrieztos pa labi (saskaņā ar ) un sasniegtu krustojumu D. Krustojumā D robots atkal nogrieztos pa labi, tāpēc ieietu (mūžīgajā) ciklā. Tas nav risinājums, kuru mēs vēlamies atrast, tāpēc mēs atgriezāties krustojumā C, kur pēdējoreiz veicām simbolu nozīmes izvēli. Bet šajā krustojumā abas iespējas jau esam izskatījuši, tādēļ jāatgriežas vēl tālāk - līdz krustojumam F un jāizmēģina cita simbolu nozīme un, kā sekas, cits maršruta turpinājums.







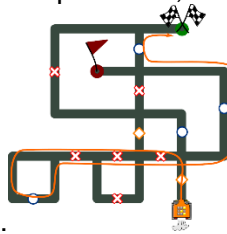
- [2. minējums] Pieņemsim, ka robots krustojumā F pagriezīsies pa kreisi, kas nozīmē, ka simbols  nozīmē "pagriezies pa kreisi".
- [2.1. minējums] Kad robots sasniedz krustojumu D, atkal pastāv vairākas iespējas. Robots nevar pagriezies pa kreisi, jo tas nozīmētu, ka simboliem  un  ir vienāda nozīme. Robots varētu krustojumā D pagriezies pa labi, kas nozīmētu, ka simbols  nozīmē "pagriezies pa labi", un tāpēc simbols  nozīmē "iet taisni". No D robots turpinās ceļu uz C. Pie C tas pagriezīsies pa kreisi, saskaņā ar . Krustojumā A tas būtu pagriezies pa labi, saskaņā ar . Krustojumā B tas dotos taisni līdz krustojumam C, saskaņā ar . Krustojumā C tas atkal nogrieztos pa labi, saskaņā ar , nonākot ciklā. Tas nav tas, ko mēs vēlamies, tāpēc mēs atgriezāties līdz krustojumam D, kur pēdējoreiz nolēmām



pagriezies pa labi un izpētīt citu maršrutu.

- [2.2. minējums] Tagad pieņemsim, ka robots ceļu turpina taisni (atcerieties, ka tas nevar pagriezies pa kreisi), kas simbolam  piešķir nozīmi "doties taisni" un simbolam  - "pagriezies pa labi". Tagad var izsekot maršrutam un saprast, ka

robots sasniegs . Tādējādi pareizā simbolu nozīmes interpretācija ir šāda:
 nozīmē "pagriezties pa kreisi",  nozīmē "doties taisni" un  nozīmē



"pagriezties pa labi".

Atbildes izskaidrojums:

Pirmā izskaidrojumā piedāvātā stratēģija tiek saukta par *brutālā spēka* metodi. Tas nozīmē izskatīt un pārbaudīt visas iespējas, līdz tiek atrasts vēlamais rezultāts. Dažreiz var būt daudz iespēju, un visu izskatīšana var aizņemt daudz laika, tāpēc ir jāatrod citas stratēģijas.

Otru stratēģiju sauc par *atgriezmeklēšanu*. Izmantojot šo paņēmienu, pakāpeniski tiek veidoti risinājumu varianti un varianti tiek atmesti, tiklīdz tiek konstatēts, ka variants nenoved pie vēlamā rezultāta. Atgriezmeklēšanas priekšrocība attiecībā pret brutālā spēka metodi ir tāda, ka jums nav jāpārskata jaunie varianti no paša sākuma, jo jūs atgriežaties pie tā soļa, kurā izdarījāt pēdējo izvēli, un turpināt no turienes.

Konkrētajā uzdevumā brutālā spēka metode var darboties labāk, jo robotam ir maz mainīgo (simbolu) un maz izpētāmo ceļu. Tomēr kopumā lielākiem un sarežģītākiem uzdevumiem, kad mainīgo skaits palielinās un ir vairāk izpētāmo ceļu, atgriezmeklēšana nodrošinās labāku un elegantāku risinājumu.

Mīklas, piemēram, Sudoku, var eleganti atrisināt, izmantojot atgriezmeklēšanu.

Stumbru kārtošana

Vācija

Upē, viens otram blakus, peld dažādu izmēru koku stumbri. Bebra Hamida uzdevums ir sašķirot stumbrus pēc to lieluma.

Hamids pārvietojas gar upes krastu, vienmēr apstājoties starp diviem blakus stumbriem. Viņš salīdzina šos divus stumbrus pēc lieluma un vajadzības gadījumā tos apmaina. Hamids zina, ka viņš var šķirot stumbrus šādā veidā neatkarīgi no tā, kāds ir stumbru sākuma izkārtojums:

- *Sāciet no sākuma stāvokļa - kreisajā malā esošā stumbra labajā pusē.*
- *Atkārtojiet darbības, līdz esat nonācis labajā malā esošā stumbra labajā pusē:*
 - *Ja stubrs pa kreisi ir mazāks par stubru pa labi, tad paejiet vienu stubru pa labi.*
 - *Ja stubrs pa labi ir mazāks par stubri pa kreisi, tad:*
 - *samainiet šos stumbrus vietām;*
 - *ja vien neesat sākuma stāvoklī, tad paejiet vienu stubru pa kreisi.*

Darbību skaits, kas Hamidam jāveic, lai sakārtotu stumbrus, ir atkarīgs no tā, kādā secībā stumbri atrodas pirms šķirošanas. Sliktākajā gadījumā Hamidam sešu stumbru sašķirošanai būtu jāveic 30 darbības.

Uzdevums:

Ja Hamidam būtu jāsakārto 60 stumbri, tad kādā diapazonā būtu veicamo darbību skaits?

Atbilžu varianti:

- A. [0 ... 30]
- B. [6 ... 70]
- C. [59 ... 300]
- D. [59 ... 3600]

Pareizā atbilde: D

- A. [0 ... 30] ir nepareizs. Pat labākajā gadījumā, ja stumbri jau no sākuma ir sakārtoti, Hamidam ir jāveic 59 gājieni, kas jau ir vairāk nekā 30.
- B. [6 ... 70] un C [59..300] arī ir nepareizi. Lai to pierādītu, mums ir jāizpēta sliktākais gadījums, kad stumbri sākumā ir sakārtoti pretējā secībā. Šajā gadījumā Hamids sasniedz stumbru k -tajā pozīcijā un pārvieto to uz pirmo (kreisāko) pozīciju, pēc tam dodas uz stumbru $(k+1)$ -ajā pozīcijā. Tātad stumbram k -tajā pozīcijā mums jāpārvietojas $k-2$ reizes pa labi, lai to sasniegtu, un $k-2$ reizes pa kreisi, lai to novietotu sākumā. Tādējādi mēs iegūstam summu $2(1+2+\dots+58)$, un algoritma beigās mums vēl jāpievieno 59 gājieni no kreisākās līdz labējai pozīcijai. Summa ir tieši $59 \cdot 59 = 3481$. Šis skaitlis nepieder nevienam no aplūkotajiem diviem diapazoniem.
- D. [59 ... 3600] ir pareizs. Pierādīsim, ka iepriekš analizētais sliktākais gadījums patiešām ir vissliktākais. Kad Hamids sasniedz stumbru k -tajā pozīcijā, visi iepriekšējie stumbri jau ir sakārtoti pareizi, tāpēc viņam atliek tikai ievietot šo jauno stubru pareizajā vietā starp iepriekšējiem. Tad viņš dodas uz $(k+1)$ pozīciju. Tātad, jo mazāks ir stubrs k -tajā pozīcijā, jo lielāks gājienu skaits tā novietošanai būs nepieciešams.

Atbildes izskaidrojums:

Datorzinātnē kārtošanas algoritmus izmanto, lai sakārtotu objektus noteiktā secībā. Visbiežāk izmantotā ir skaitliskā (skaitļiem) un leksikogrāfiskā secība (visu veidu datiem),

kuru pamatā ir sakārtots alfabēts). Efektīva šķirošana ir svarīga, lai optimizētu citu algoritmu, piemēram, meklēšanas algoritmu, kuriem ir nepieciešami sakārtoti ievaddati, efektivitāti. Kārtošana var būt noderīga, lai normalizētu datus un radītu cilvēkam lasāmu izvadi.

Viens no zināmajiem kārtošanas algoritmiem ir *gnome sort*. Tas ir konceptuāli vienkāršs, un tas vienlaikus darbojas ar vienu objektu, taču, veicot mijmaiņas darījumus, objekts nokļūst pareizajā vietā. Ja saraksts sākotnēji ir gandrīz sakārtots, n objektu sakārtošanai pietiek ar n mijmaiņas darbībām. *Gnome sort* (saukts par "stulbu kārtošanu") sākotnēji 2000. gadā ierosināja irāņu datorzinātnieks Hamids Sarbazi-Azads (Šarifas Tehnoloģiju universitātes Datorzinātņu un inženierzinātņu profesors).

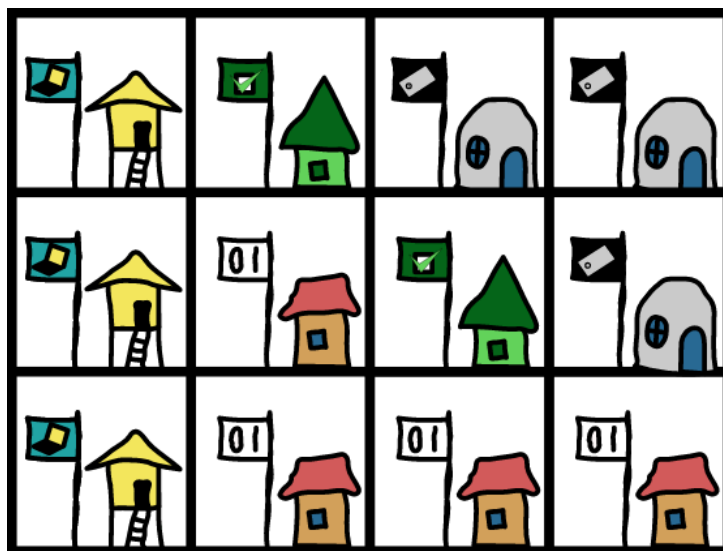
Runājot par kādu algoritmu, mums vienmēr ir jādomā, cik ātrs tas ir - cik darbību noteiktam objektu skaitam tas sliktākajā gadījumā prasa. Šim algoritmam, ja mums ir n stumbri, mums nepieciešamas aptuveni n^2 darbības. Mēs to saucam par *kvadrātisku atkarību*. Citas atbildes šajā uzdevumā atspoguļo citas iespējamās sakarības: konstants darbību skaits (nav atkarības no objektu skaita) [0..30], lineāra atkarība [6..70] un logaritmiska atkarība [59..300].

Šo atkarību sauc par *algoritma sarežģītību*, un to pēta skaitļošanas sarežģītības teorija.

Cilšu apvienošanās

Filipīnas

Tālajā Bebrzemē dzīvoja četras ciltis, kas sastāvēja no vairākiem ciemiem, kā parādīts attēlā: binārieši, viedtālrunieši, izvēlrūtiņieši un klēpj datorieši. Kādu dienu viņi visi nolēma apvienoties. Tomēr, lai Bebrzemē neradītu haosu, tika nolemts, ka vienlaikus apvienosies tikai divas ciltis. Laiks, kas vajadzīgs, lai divas ciltis apvienotos, izteikts mēnešos, ir vienāds ar šo divu cilšu kopējais ciemu skaits. Pēc tam abas ciltis kļūst par vienu cilti, un apvienošanās process tiek atkārtots, līdz paliek tikai viena vienota cilts.



Uzdevums:

Kāds ir mazākais mēnešu skaits, kas vajadzīgs, lai visas ciltis apvienotos?

Atbilde:

Ievadiet mazāko mēnešu skaitu kā veselu skaitli.

Pareizā atbilde: 24

Kopējais apvienošanas reižu skaits ir 3 - par vienu mazāks nekā cilšu skaits, jo pēc katras apvienošanas kopējais cilšu skaits samazinās par 1.

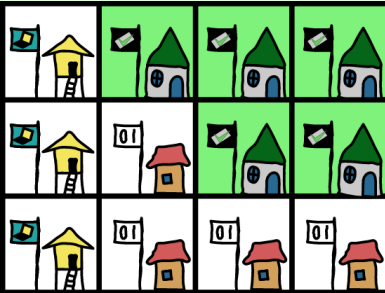
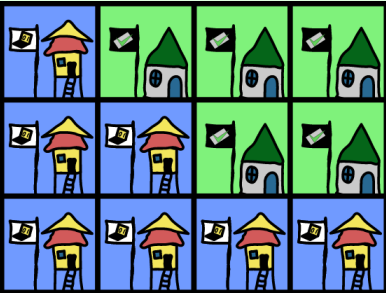
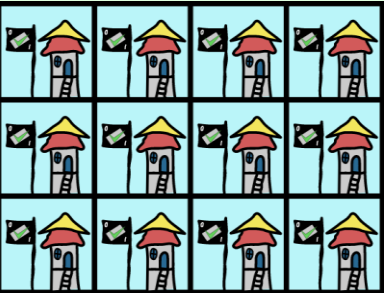
Pēdējā apvienošanās jebkurā gadījumā aizņems 12 mēnešus - visu sākotnējo cilšu ciemu kopskaits.

Ja apvienošanu veic pa ciemu pāriem, tad pirmās divas apvienošanas kopā arī aizņems 12 mēnešus, turklāt tas nav atkarīgs no tā, kā ciltis tiks sadalītas pa pāriem.

Ja izvēlas citu stratēģiju, ka pirmajam apvienoto cilšu pārim pievienojas trešā cilts, tad otrās apvienošanas ilgums būs vienāds ar pirmās apvienošanas ilgumu plus trešās cilts ciemu skaits. Dotajam ciemu skaitam pirmās divas apvienošanās aizņems, mazākais, 13 mēnešus $((2+3)+(2+3+3))$, kas ir vairāk nekā 12.

Šajā uzdevumā darbojas šāda vispārīga pieeja: katru apvienošanu nepieciešams veikt divām ciltīm ar mazāko ciemu skaitu.

Viens no šādiem apvienošanās variantiem parādīts zemāk esošajā tabulā:

| | | |
|--|--|--|
| <p>1. Apvienojas izvēlrūtiņieši (2 ciemi) un viedtālrūņieši (3 ciemi). Pēc apvienošanās viņus var saukt par IV-iešiem.</p> | <p>2. Tagad apvienojas divas mazākās - bināriešu (4 ciemi) un klēpjdatoriešu (3 ciemi) ciltis. Pēc apvienošanās tos varam saukt par BK-iešiem.</p> | <p>3. Visbeidzot, 5 IV-iešu un 7 BK-iešu ciemi apvienojas vienā lielā IVBK-iešu ciltī.</p> |
|  |  |  |
| <p>Tas aizņems <u>5 mēnešus</u>, un rezultātā tiek iegūti 3 klēpjdatoriešu, 4 bināriešu un 5 IV-iešu ciemi.</p> | <p>Tas aizņems <u>7 mēnešus</u>, un rezultāts ir 5 IV-iešu un 7 BK-iešu ciemi.</p> | <p>Tas aizņem <u>12 mēnešus</u>.</p> |

Tāpēc minimālais mēnešu skaits, lai apvienotos visas četras Bebrzemes ciltis, ir 24 (tabulā attēlotajā apvienošanas variantā 5+7+12).

Atbildes izskaidrojums:

Šis ir *optimizācijas uzdevuma* piemērs, kura mērķis ir izstrādāt stratēģiju, kas maksimāli palielina vai samazina noteiktu skaitlisku lielumu, ievērojot noteiktus ierobežojumus. Optimizācijas uzdevumi ir sastopami ikdienā: atrast īsāko ceļu uz galamērķi, izveidojot grafu, kas atbilst vislielākajam nepārklājošos darbību skaitam utt. Ir vairāki veidi, kā risināt optimizācijas uzdevumus, un viens no tiem ir *rijīgie algoritmi*.

Rijīgie algoritmi balstās uz pieņēmumu, ka, veicot labāko izvēli katrā posmā (*lokālais optimums*), tiks sasniegts labākais galarezultāts (*globālais optimums*). Šajā uzdevumā šis pieņēmums tiek apmierināts: ciltīm ir jāsamazina katras apvienošanās mēnešu skaits, lai samazinātu kopējo mēnešu skaitu visā apvienošanās procesā. Tomēr jāuzsver, ka, lai gan rijīgā paradigma nav universāls risinājums visiem optimizācijas problēmu veidiem, tomēr tā parasti nodrošina pieņemama tuvinājuma atrašanu saprātīgā laikā.

Datu pārveidošana

Turcija

Datus var kodēt un attēlot dažādos veidos. Vienu veidu var transformēt par citu, pakļaujot datus procesiem, ko sauc par pārveidošanu. Bebrs Elifa ir izgudrojusi simbolu virkņu pārveidošanas iekārtu (\rightarrow) un izveidojusi piecus procesus, kas katrs kādu simbolu grupu pārvērš citā (rezultāts), kā parādīts zemāk:

| process | sākotnējie simboli | rezultāts |
|---------|--------------------|-------------------|
| ! | * | \rightarrow @@ |
| ? | # | \rightarrow ** |
| & | *@ | \rightarrow # |
| ^ | ## | \rightarrow *@@ |
| \$ | @@@ | \rightarrow |

Katrs process simbolu virknē ietekmē tikai pirmo simbolu grupu, kas atbilst konkrētajam procesam, piemēram, ## \rightarrow ? = **#, bet ne ****. Ja simbolam process nav piemērojams, iekārta neko nedara un pāriet uz nākamo procesu.

Tādējādi Elifa var sasniegt vēlamu rezultātu, pārveidojot simbolu virknes kamēr vien iespējams.

Uzdevums:

Kāda būs procesu secība, lai Elifas iekārta divu simbolu virkni "##" pārveidotu par trīs simbolu virkni "@@@"?

Atbilžu varianti:

- A. ## \rightarrow & \rightarrow ^ \rightarrow & \rightarrow ? \rightarrow ! \rightarrow ? \rightarrow \$
- B. ## \rightarrow & \rightarrow ? \rightarrow & \rightarrow ^ \rightarrow ? \rightarrow ! \rightarrow \$
- C. ## \rightarrow ^ \rightarrow & \rightarrow ? \rightarrow & \rightarrow ? \rightarrow ! \rightarrow ? \rightarrow \$
- D. ## \rightarrow ^ \rightarrow & \rightarrow ? \rightarrow & \rightarrow ? \rightarrow ! \rightarrow ! \rightarrow ! \rightarrow \$

Pareizā atbilde: D

Visi simbolu virknes pārveidošanas posmi procesu ietekmē ir parādīti zemāk.

D: ## \rightarrow *@@ \rightarrow #@ \rightarrow **@ \rightarrow *# \rightarrow *** \rightarrow @@** \rightarrow @@@@* \rightarrow @@@@@@@ \rightarrow @@@

Citas atbildes ir nepareizas.

A: ## \rightarrow ## \rightarrow *@@ \rightarrow #@ \rightarrow **@ \rightarrow @@*@@ \rightarrow @@*@@ \rightarrow @@*@@

B: ## \rightarrow ## \rightarrow **# \rightarrow **# \rightarrow **# \rightarrow **** \rightarrow @@*** \rightarrow @@***

C: ## \rightarrow *@@ \rightarrow #@ \rightarrow **@ \rightarrow *# \rightarrow *** \rightarrow @@** \rightarrow @@** \rightarrow @@**

Atbildes izskaidrojums:

Šajā uzdevumā tiek izmantota šablonu atpazīšana un pārveidošana. Datu konvertēšana pastāvīgi notiek datoros dažādos veidos pārveidojot datus no viena formāta uz citu. No otras

pusēs, arī šablonu un paraugu atpazīšana tiek izmantota, lai atklātu likumsakarības lielos datu masīvos. Šis uzdevums apvieno šīs divas datorzinātņu tēmas. Uzdevums ne tikai pieprasa veikt izmaiņas pārveidošanas procesā, bet arī prasa atšķirt zīmes, realizējot dažādus pārveidošanas modeļus.

Aprakstītā iekārta faktiski attēlo Markova algoritmu. Šo algoritmu veido dažādi likumi, saskaņā ar kuriem simbolu grupas tiek aizstātas ar citām. Markova algoritmam ir arī stingri darba pārtraukšanas nosacījumi, kas nav realizēti aprakstītajā iekārtā. Pirmkārt, algoritms apstājas, ja neviens likums nav piemērojams, un, otrkārt, daži likumi tiek atzīmēti kā beigu likumi.

Alu bizness

Lielbritānija

Lauks sadalīts 8 x 8 rūtiņu režģī. Āpši un truši dzīvo laukā izvietotās alās, kur Ā = āpsis un T = trusis (skat. attēlu zemāk).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| T | | Ā | | Ā | T | | |
| | Ā | T | | | | T | |
| Ā | | | | T | | | Ā |
| | T | T | | Ā | | T | |
| Ā | | | Ā | | T | Ā | |
| | Ā | T | | | | | |
| | | | T | | Ā | | T |
| Ā | | T | Ā | | T | | Ā |

Āpši un truši labi satiek viens ar otru. Tomēr, ja āpsis vai trusis noskaidro, ka dzīvo tādā alā, kurai kaimiņos ir mazāks skaits sava veida īpatņu, viņš šo alu pamet un dodas uz citu lauka rūtiņu. Pārvietojoties, visi ievēro *Alu virsnieka pārvietošanās noteikumus*.

Alu virsnieka pārvietošanās noteikumi:

1. Visi āpši tiek aptaujāti, vai viņi vēlas pārvietoties, un, kad visi āpši ir aptaujāti, tiek aptaujāti visi truši.
2. Dzīvnieki tiek aptaujāti pēc kārtas noteiktā secībā: rindu pēc rindas (no augšas uz leju), un katru rindu ejot no kreisās puses uz labo.
3. Kad āpsis vai trusis tiek aptaujāts un vēlas pārvietoties, tas nekavējoties pāriet uz citu režģa daļu, izvēloties pirmo kvadrātu, kas atbilst Alu virsnieka pārvietošanās noteikumiem tādā pašā secībā kā iepriekš: rindu pēc rindas no augšas uz leju un katrā rindā no kreisās uz labo pusi.
4. Katrs āpsis un trusis tiek aptaujāts tikai vienu reizi.

Piemēri:

- Āpsis, kuram neviena no astoņām apkārt esošajām rūtiņām nav aizņemta, paliks tur, kur atrodas.
- Āpsis, kuram blakus rūtiņās atrodas viens āpsis, viens trusis un sešas rūtiņas nav aizņemtas, paliks tur, kur ir.
- Āpsis, kuram blakus rūtiņās atrodas viens āpsis un divi truši, izvēlēsies pārvietoties.

Uzdevums:

Kad visi āpši un visi truši būs pārcēlušies, kurš no šiem attēliem parāda pareizo dzīvnieku izvietojumu?

Atbilžu varianti:

| A | B | C | D | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|---|---|---|--|--|---|--|--|--|--|--|--|--|--|---|--|---|--|--|--|--|---|--|--|--|---|---|--|--|--|--|--|--|--|--|---|--|--|--|---|---|--|---|--|--|--|--|--|--|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|---|--|--|--|--|--|--|--|---|--|--|--|--|--|--|---|---|---|--|--|---|--|--|---|---|---|--|--|--|--|--|--|---|---|--|---|--|--|--|--|---|---|--|--|--|--|--|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|--|--|---|--|---|--|---|--|--|---|--|--|--|--|--|---|--|--|--|--|--|--|--|--|--|---|--|--|--|--|---|---|--|--|---|--|--|---|---|---|---|--|---|--|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|--|--|--|--|---|--|--|--|---|--|---|--|--|--|---|--|--|---|--|--|--|--|--|--|--|--|--|---|--|---|--|--|---|---|--|--|--|--|--|---|
| <table border="1"> <tr><td></td><td>Ā</td><td>Ā</td><td>Ā</td><td>Ā</td><td>T</td><td>T</td><td>T</td></tr> <tr><td>Ā</td><td>Ā</td><td></td><td>Ā</td><td>Ā</td><td>T</td><td>T</td><td>T</td></tr> <tr><td>Ā</td><td>Ā</td><td>Ā</td><td></td><td></td><td>T</td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td>T</td><td></td></tr> <tr><td>Ā</td><td></td><td></td><td></td><td></td><td>T</td><td></td><td></td></tr> <tr><td></td><td>Ā</td><td>T</td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td>T</td><td></td><td></td><td></td><td>T</td></tr> <tr><td>Ā</td><td></td><td>T</td><td></td><td></td><td></td><td></td><td></td></tr> </table> | | Ā | Ā | Ā | Ā | T | T | T | Ā | Ā | | Ā | Ā | T | T | T | Ā | Ā | Ā | | | T | | | | | | | | | T | | Ā | | | | | T | | | | Ā | T | | | | | | | | | T | | | | T | Ā | | T | | | | | | <table border="1"> <tr><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td></tr> <tr><td>Ā</td><td>T</td><td>T</td><td>T</td><td>T</td><td></td><td></td><td></td></tr> <tr><td>Ā</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>Ā</td><td></td><td></td><td></td><td></td><td></td><td></td><td>Ā</td></tr> <tr><td>Ā</td><td>Ā</td><td></td><td></td><td>Ā</td><td></td><td></td><td>Ā</td></tr> <tr><td>Ā</td><td>Ā</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>Ā</td><td>Ā</td><td></td><td>T</td><td></td><td></td><td></td><td></td></tr> <tr><td>Ā</td><td>Ā</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table> | T | T | T | T | T | T | T | T | Ā | T | T | T | T | | | | Ā | | | | | | | | Ā | | | | | | | Ā | Ā | Ā | | | Ā | | | Ā | Ā | Ā | | | | | | | Ā | Ā | | T | | | | | Ā | Ā | | | | | | | <table border="1"> <tr><td>Ā</td><td>Ā</td><td>Ā</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td></tr> <tr><td>Ā</td><td>Ā</td><td>Ā</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td></tr> <tr><td>Ā</td><td>Ā</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td>Ā</td><td></td><td>T</td><td></td></tr> <tr><td>Ā</td><td></td><td></td><td>Ā</td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>Ā</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td>T</td><td></td><td></td><td></td><td></td></tr> <tr><td>Ā</td><td>T</td><td></td><td></td><td>T</td><td></td><td></td><td>Ā</td></tr> </table> | Ā | Ā | Ā | T | T | T | T | T | Ā | Ā | Ā | T | T | T | T | T | Ā | Ā | | | | | | | | | | | Ā | | T | | Ā | | | Ā | | | | | | Ā | | | | | | | | | | T | | | | | Ā | T | | | T | | | Ā | <table border="1"> <tr><td>Ā</td><td>Ā</td><td></td><td>T</td><td></td><td>T</td><td>T</td><td>T</td></tr> <tr><td>Ā</td><td>Ā</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td></tr> <tr><td>Ā</td><td>Ā</td><td>T</td><td>T</td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td>Ā</td><td></td><td></td><td></td></tr> <tr><td>Ā</td><td></td><td>Ā</td><td></td><td></td><td></td><td>Ā</td><td></td></tr> <tr><td></td><td>Ā</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td>T</td><td></td><td>Ā</td><td></td><td></td></tr> <tr><td>Ā</td><td>T</td><td></td><td></td><td></td><td></td><td></td><td>Ā</td></tr> </table> | Ā | Ā | | T | | T | T | T | Ā | Ā | T | T | T | T | T | T | Ā | Ā | T | T | | | | | | | | | Ā | | | | Ā | | Ā | | | | Ā | | | Ā | | | | | | | | | | T | | Ā | | | Ā | T | | | | | | Ā |
| | Ā | Ā | Ā | Ā | T | T | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | | Ā | Ā | T | T | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | Ā | | | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | | | | | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Ā | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | T | | | | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| T | T | T | T | T | T | T | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | T | T | T | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | | | | | | | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | | | Ā | | | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | Ā | T | T | T | T | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | Ā | T | T | T | T | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | Ā | | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | | | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | T | | | T | | | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | | T | | T | T | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | T | T | T | T | T | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | Ā | T | T | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | | Ā | | | | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | T | | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Ā | T | | | | | | Ā | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Pareizā atbilde: A

Atbilde A ir pareiza. Tas ir dotā algoritma izpildes rezultāts.

Ir nepieciešama uzmanīga algoritma pielietošana, jo dzīvnieku pārvietošanās dēļ situācija dinamiski mainās.

Lai atrastu atbildi, aplūkosimi lauka augšējā kreisā stūra kvadrātu, kurā sākumā atrodas trusis.

Tā kā truši pārvietojas pēc āpšiem, āpsis nekad nevar nokļūt šajā laukumā. Tas izslēdz variantus C un D. Tad mums ir jānosaka, vai trusis paliks vai pārvietosies.

Apstrādājot augšējo rindu, pirmais āpsis paliek vietā. Otrais izvēlas pārvietoties, jo tam ir tikai viens kaimiņš, un tas ir trusis. Pirmā pieejamā un āpsim piemērotā vieta ir otrais kvadrāts augšējā rindā blakus pirmajam trusim. Tas nozīmē, ka tad, kad visi āpši būs pārcēlušies, trusis laukuma augšējā kreisajā stūrī izvēlēsies pārvietoties, atstājot rūtiņu tukšu. Neviens cits trusis uz to nepārcelsies, jo šai rūtiņai kaimiņos ir tikai āpši, bet nav trušu. Vienīgā atbilde, kurā augšējā kreisā stūra rūtiņa ir tukša, ir atbilde A.

Atbilde B tiek iegūta, ja dzīvnieks par pieņemamu atzīst tikai tādu situāciju, kurā viņa veida īpatņi ir vairāk, nekā otra veida. Arī šajā gadījumā vispirms tiek aptaujāti āpši un tad truši. Bet šis ir cits algoritms, tāpēc šī atbilde nav pareiza.

Atbilde C attēlo situāciju, ka dzīvnieki tiek aptaujāti un, ja nepieciešams, pārvietojas pēc kārtas, nešķirojot āpšus un trušus. Pārējie algoritma noteikumi ir ievēroti.

Atbilde D attēlo situāciju, ka vispirms tiek aptaujāti, un, ja nepieciešams, pārvietojas truši un tikai tad - āpši. Pārējie algoritma noteikumi ir ievēroti.

Atbildes izskaidrojums:

Uzdevuma izaicinājums ir precīzi pielietot doto algoritmu, kura pamatā ir Šellinga segregācijas modelis. Tas modelē sociālo izvēļu rezultātus, proti to, kur cilvēki dzīvos, ņemot vērā saprātīgu izvēli. Reālās dzīves modelis ir sarežģītāks nekā šajā uzdevumā izmantotais, un parāda, ka divas cilvēku grupas dabiski sadalīsies atsevišķās kopienās, atklāti neizrādot aizspriedumus, bet vienkārši vēloties, lai to tuvumā atrastos viņiem līdzīgi cilvēki. Tas palīdz izskaidrot, kāpēc daudzās pasaules pilsētās veidojas geto - cilvēku grupas, kurām ir viena un tā pati reliģija, kultūra vai ienākumu līmenis, bieži vien kompakti dzīvo noteiktos pilsētas apgabalos. Ja nekustamā īpašuma aģents izvēlas ietaupīt laiku un, piemēram, piedāvā mājokļus bagātiem cilvēkiem tikai apgabalos, kur vairums bagāto cilvēku jau dzīvo, tas padara segregāciju vēl izteiktāku.

Lai gan jautājumā izmantotais modelis ir ļoti vienkāršots, var redzēt, ka pat atšķirīgu algoritmu vai tā variāciju izmantošana rada segregāciju.

Bebr[a]s' 21



Copyright Bebras