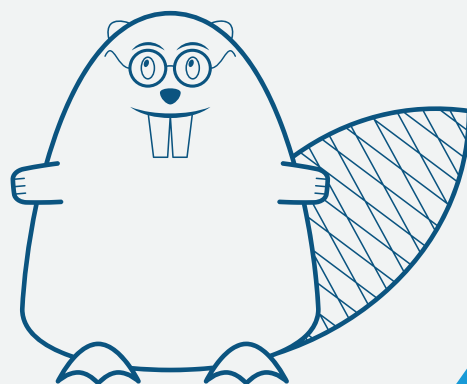


Bebr[a]s

2019. gada
1. kārtas uzdevumi ar atbildēm
9.-10. klasei



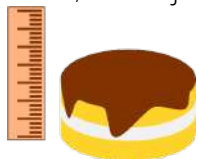
SATURS

<u>Kūkas un kaimiņi</u>	<u>2</u>
<u>Lidojumu plānošana</u>	<u>4</u>
<u>Runājošie mežgli</u>	<u>7</u>
<u>Īstais apavu izmērs</u>	<u>9</u>
<u>Izšuvums</u>	<u>12</u>
<u>Senā valoda</u>	<u>14</u>
<u>Saudzēsim vidi</u>	<u>16</u>
<u>Bišu strops</u>	<u>18</u>
<u>Sarkangalvīte</u>	<u>20</u>
<u>Noliktavas</u>	<u>22</u>
<u>Skaitītājs</u>	<u>23</u>
<u>Numurētās kārtis</u>	<u>25</u>
<u>Dzelzceļa robots</u>	<u>26</u>
<u>Kimikālijas</u>	<u>28</u>
<u>Robota zīmējums</u>	<u>30</u>

Kūkas un kaimiņi

Beļģija

Piektdienas rītā trīs kaimiņienes - Anna, Betija un Klāra, sestdienas apkaimes ballītei katra pasūtīja pa kūkai no viena un tā paša konditora. Visas trīs sākotnēji pasūtīja viena tipa kūku, kas bija 3 cm augsta.



Tomēr katra pēcāk zvanīja konditoram atkārtoti, lai mainītu pasūtījumu. Katru reizi konditors pierakstīja jauno pasūtījumu, veco izmetot ārā. Kūkas būs gatavas agrā sestdienas rītā.

Anna zvana: Uztaisi manu kūku par 1cm augstāku, nekā es sākumā pasūtīju.

Vēlāk Anna zvana vēlreiz : Tagad uztaisi manu kūku tikpat augstu, kā Betijai.

Betija zvana: Uztaisi manu kūku par 2cm augstāku, nekā es sākumā pasūtīju.

Vēlāk Betija zvana vēlreiz: Uztaisi manu kūku par 1cm zemāku, nekā es pasūtīju.

Klāra zvana: Uztaisi manu kūku par 1cm augstāku, nekā Annas kūka.

Vēlāk Klāra zvana vēlreiz: Uztaisi manu kūku par vēl 1cm augstāku, nekā iepriekš pasūtīju.

PIEZĪME: Mēs nezinām, kādos laikos konditoram tika zvanīts. Zinām tikai to, ka otrais zvans no katras kaimiņienes tika veikts uzreiz pēc pirmā zvana.

Uzdevums:

Kurš no zemāk redzamajiem apgalvojumiem ir pareizs neatkarīgi no laikiem, kad zvani tika izdarīti?

Atbilžu varianti:

- A) Annas un Betijas kūkām ir vienāds augstums
- B) Betijas kūka ir vismaz 1cm zemāka par Klāras kūku
- C) Klāras kūka ir tieši 2cm augstāka par Annas kūku
- D) Visas 3 kūkas ir vismaz 4cm augstas

Pareizā atbilde: B

Atbilde A nav pareiza: ja Betija savus zvanus veica pēc Annas, Annas kūka beigās būs 3cm augsta, bet Betijas kūka - 4cm augsta. Tas uzreiz parāda, ka D nav pareizā atbilde.

Atbilde C nav pareiza: ja Klāra veiks savus zvanus pirmā, pēc tam Betija un tikai tad Anna, Klāras kūka būs 5cm augsta, Betijas kūka - 4cm, un Annas kūka arī 4cm.

Ja paskatāmies uz Betijas norādījumiem konditoram, var redzēt, ka kādā brīdī pēcpusdienā konditors būs pierakstījis augstumus 3cm, 5cm un 4cm Betijas kūkai. Tā rezultātā arī Annas kūkai cepējs varēja pierakstīt augstumus 3cm, 4cm un 5cm.

Tas nozīmē, ka sestdien Klāras kūka būs 5cm, 6cm vai 7cm augsta. Betijas kūka vienmēr būs 4cm augsta, tātad atbilde B ir pareizā.

Kā tas ir saistīts ar informātiku?

Mūsdienu datori var vienlaikus veikt vairākus darbus, tajā skaitā pat vienas lietojumprogrammas ietvaros. Piemēram, teksta apstrādes programmas var pārbaudīt pareizrakstību teksta ievades laikā.

Kā parādīts arī šajā uzdevumā, kad vairāki uzdevumi tiek izpildīti paralēli, ne vienmēr ir iespējams noteikt kāds būs iznākums. Tas tādēļ, ka bieži ir grūti paredzēt, kāda būs reālā darbu izpildes secība un kad kurš darbs reāli tiks izpildīts.

Programmēšanas veids, kurā vairākas programmas daļas var tikt izpildītas vienlaicīgi, sauc par laiksakritīgo programmēšanu. Rakstot šādas programmas, programmētājiem jāuzmanās no šādiem efektiem un par noteiktām daļām jābūt pārliecinātiem, ka tās vienmēr tiks izpildītas iepriekš paredzētajā secībā.

Piemēram, programmētājam, kurš izstrādā programmu tīkla printerim, jāpārliecinās, ka divi dokumenti, kas nosūtīti drukāšanai vienlaicīgi, tiks pilnībā drukāti viens aiz otra, nevis pamīšus, ņemot pa lapai (vai, vēl sliktāk, pa rindai!) no katra.

Ņemot vērā to, ka mūsdienu datori īpaši tiek veidoti vairāku vienlaicīgu uzdevumu izpildei, tādi šo iespēju korektas izmantošanas līdzekļi kā laiksakritīgā programmēšana iegūst aizvien lielāku nozīmi un tiek iekļauti datorzinātņu mācību programmās.

Lidojumu plānošana

Indija

Kad lidmašīna nolaižas lidostā, tai ir iedalīts atsevišķs skrejceļš no citām lidmašīnām, lai izvairītos no negadījumiem.

Bebru Salas Lidostā divas lidmašīnas nevar nolaisties uz viena un tā paša skrejceļa, ja to nolaišanās laiki atšķiras par 15 minūtēm vai mazāk.

Piemēram, 1.lidmašīna nolaižas plkst. 6.10, 2.lidmašīna plkst. 6.25, bet 3.lidmašīna plkst. 6.26. Šādā gadījumā 1.un 2.lidmašīna nevar izmantot vienu un to pašu skrejceļu, bet 3.lidmašīna var nolaisties uz tā paša skrejceļa, kuru izmantoja 1.lidmašīna.

Tu esi lidojumu kontrolieris, un tev jāpiešķir lidmašīnām skrejceļi pēc zemāk redzamā grafika:

Lidmašīna	Laiks
9W2400	7:00
9W1321	7:21
AI561	7:20
IP397	7:40
AI620	7:18
EC254	7:34
EK427	7:03
BQ439	7:36
SG147	7:12
VA035	7:28

Uzdevums:

Kāds ir mazākais skrejceļu skaits, kas nepieciešams, lai visas lidmašīnas nolaistos savos laikos atbilstoši noteikumiem?

Pareizā atbilde: 4

Vispirms lidmašīnas ar to nosēšanās laikiem jāsakārto augošā secībā:

1. 9W2400 7:00
2. EK427 7:03
3. SG147 7:12
4. AI620 7:18
5. AI561 7:20
6. 9W1321 7:21
7. VA035 7:28
8. EC254 7:34
9. BQ439 7:36

9W2400 nolaižas 7:00, tāpēc mēs tai dodam 1.skrejceļu. Nākamās EK427 un SG147 nolaižas ar nepilnu 15min atšķirību, tāpēc mēs katrai dodam savu skrejceļu. Tātad pašlaik mums ir izmantoti 3 dažādi skrejceļi.

Katrai nākamajai lidmašīnai mēs centīsimies piešķirt tos skrejceļus, kas jau ir izmantoti, ja tas ir iespējams. Turklāt vienmēr izvēlēsimies to skrejceļu, kas pirmais ir atbrīvojies - turot hronoloģiski vissenāk izmantotu skrejceļu "rezervē" un neļaujot uz tā nosēsties, kopējo nosēšanās grafiku var tikai pasliktināt. AI620 nolaižas 7.18, kas ir vairāk nekā 15min pēc pirmās lidmašīnas, tātad mēs to varam likt uz 1. skrejceļa. AI561 nevaram likt uz tā paša skrejceļa, jo nebūs pagājušas 15 minūtes no pirmā un ceturtā lidojuma, bet varam tai piešķirt 2. skrejceļu, jo tā ir par 15 minūtēm vēlāk nekā EK427.

9W1321 nolaižas 7:21, un tas ir mazāk par 15 min atšķirībā par SG147 (7:12), AI620 (7:18), un AI561 (7:20), kas jau ir sadalītas pa 3 skrejceļiem. Tātad mums šai lidmašīnai ir jāizdala vēl cits skrejceļš. Un tā turpina ar visām lidmašīnām.

Tātad sanāk, ka lidmašīnas nolaidīsies uz šādiem skrejceļiem:

1.skrejceļš: 9W2400 (7:00), AI620 (7:18), EC254 (7:34)

2.skrejceļš: EK427 (7:03), AI561 (7:20), BQ439 (7:36)

3.skrejceļš: SG147 (7:12), VA035 (7:28)

4.skrejceļš: 9W1321 (7:21), IP397 (7:40)

Kā tas ir saistīts ar informātiku?

Šajā uzdevumā ir aprakstīti konflikti - divas lidmašīnas nedrīkst izmantot vienu skrejceļu, ja nolaišanās laiks atšķiras par ne vairāk kā 15 minūtēm) un resursi (piešķiramie skrejceļi), kas jāsadala tā, lai izvairītos no konfliktiem. Ir zināmi dažādi šī tipa uzdevumi. Piemēram, noskaidrot cik galdi būs nepieciešami saviesīgam pasākumam, lai pie viena galda nebūtu jāsēdina viesi, kas savā starpā nesatiek.

Pat ja aplūkojamo objektu (lidmašīnu, viesu, ...) skaits ir liels, jūs tik un tā vēlaties uzdevumu atrisināt ātri. Tādēļ informātikā šādu uzdevumu atrisināšanai ir izstrādātas īpašas metodes.

Viena no metodēm uzdevuma aprakstam izmanto grafu, kur virsotnes atbilst lidmašīnu reisiem, bet šķautnes - konfliktiem. Ja divu lidmašīnu reisu starpā ir konflikts (nolaišanās laiku mazās starpības dēļ tās nevar nolaisties uz viena skrejceļa), tad attiecīgās grafa virsotnes saista šķautne. Ja konflikta reisu starpā nav, tad arī grafā nav šķautnes starp attiecīgajām virsotnēm. Šādi sākotnējais uzdevums tiek reducēts uz grafa izkrāsošanas uzdevumu - ar kādu mazāko krāsu skaitu iespējams izkrāsot grafa virsotnes tā, lai neviena šķautne nesaistītu vienādi krāsotas virsotnes.

Viegli saprast, ka virsotnes krāsu var saistīt ar skrejceļa numuru sākotnējā uzdevumā.

Aprakstītajā uzdevumu klasē ietilpst dažādi resursu piešķiršanas uzdevumi. Piemēram, raidstaciju frekvenču piešķiršana, šablona atpazīšana, sporta spēļu plānošana, vietu plāna izveide, eksāmenu grafika sastādīšana.

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

Vienkāršākā šī uzdevuma risināšanas metode ir izmantojot t.s. "rijīgo" algoritmu - objektus (lidojumus) sakārtot augošā secībā pēc vērtības (to ielidošanas laika) un, sākot ar pirmo, pēc kārtas apstrādāt (piešķirt skrejceļus). Viegli pārliecināties, ka, lai optimāli izvietotu visus reisus, nepieciešams optimāli izvietot arī jebkuru mazāku hronoloģiski iepriekš notikušo reisu skaitu.

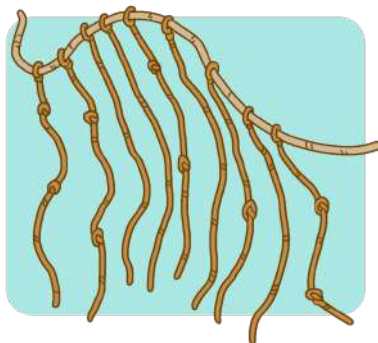
Lai rijīgais algoritms dotu optimālu risinājumu, nepieciešams pierādīt, ka katrā solī izvēloties variantu ar vislielāko vērtību un to pareizi apstrādājot, iegūtais risinājums būs labākais iespējams (vai viens no labākajiem).

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

Runājošie mezgli

Japāna

Lai paziņotu karaļvalsts jaunumus, karaliene izmanto “runājošos” mezglus, kas noteiktā veidā uzšiti uz virvēm. Piemēram, zemāk redzami mezgli nozīmē: “svinam!”.



Nozīme ir tikai virvju secībai un katrā virvē iesieto mezglu skaitam. Katrā virvē var būt iesieti 0, 1, 2 vai 3 mezgli. Ir tikai 50 dažādi iespējamie karalienes paziņojumi.

Uzdevums:

Kāds ir mazākais virvju skaits, kas karalienei ir nepieciešams jebkura paziņojuma attēlošanai?

Atbilžu varianti:

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

Pareizā atbilde: B

Ja būtu tikai viena virve, tad būtu iespējami tikai 4 dažādi paziņojumi, jo šai virvei būtu 0, 1, 2 vai 3 mezgli. Pievienojot 2. virvi, mums sanāku 4 iespējamie paziņojumi katrai, tātad $4 \times 4 = 16$ dažādi paziņojumi kopā. Šis aizvien nav pietiekami, tomēr, uzreiz kā 3. virve tiek pievienota, uzreiz jau ir $4 \times 4 \times 4 = 64$ paziņojumi. Tā kā 64 ir vairāk par 50, šis ir pietiekami, lai parādītu visus iespējamus paziņojumus.

Kā tas ir saistīts ar informātiku?

Šajā uzdevumā jums nepieciešams domāt par skaitļu pozicionālo pierakstu. Runājošo mezglu gadījumā svarīgs ir virves novietojums un tajā iesieto mezglu skaits. Tā kā katrā virvē mezglu skaits var būt viena no četrām vērtībām, tad runājošo mezglu gadījumā mums ir darīšana ar četrinieku jeb kvarternāro skaitīšanas sistēmu.

Parasti, darbojoties ar naturāliem skaitļiem, tiek lietota decimālā skaitīšanas sistēma. Decimālie cipari (no 0 līdz 9) ir kā mezglu skaits šajā uzdevumā un katra cipara pozīcija (atrašanās vieta skaitļa pierakstā pēc kārtas) līdzinās virves novietojumam pēc kārtas un atbilst noteiktai desmitnieka pakāpei. Piemēram,

$427 = 4 \times 100 + 2 \times 10 + 7$ un ar trīs decimālajiem cipariem iespējams pierakstīt $10 \times 10 \times 10 = 1000$ dažādus veselus nenegatīvus skaitļus (no 0 līdz 999).

Ļoti populāra ir arī divnieku jeb binārā skaitīšanas sistēma, kas tiek izmantota datoros datu un instrukciju glabāšanai. Šajā skaitīšanas sistēmā ir tikai divi dažādi cipari - 0 un 1, bet pozīcijas tiek sauktas par bitiem.

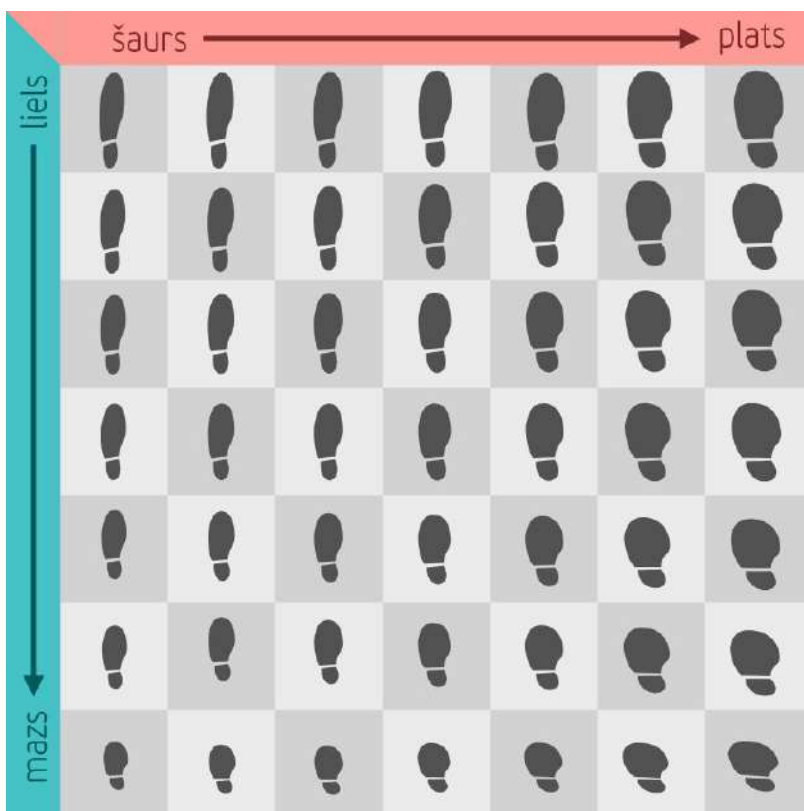
Īstais apavu izmērs

Dienvīdkoreja

Bebri devās uz apavu veikalu, lai nopirktu kurpju pāri. Viņš ieraudzīja vairākus apavus, kas bija sakārtoti tā, kā redzams attēlā zemāk.

Apavi bija sakārtoti augošā secībā pēc izmēriem un platumiem. Mazākā un šaurākā kurpe atradās apakšējā kreisajā stūrī, bet lielākā un platākā - augšējā labajā. Visi apavi bija atšķirīgos izmēros un platumos.

Būdam aizmāršīgs, bebris ir aizmirsis savu apavu izmēru, tādēļ viņam būs jāmēra kurpes tik ilgi, kamēr atradīs tās, kas der. Pareizais pāris būs tas, kura izmērs un platums būs atbilstošs kājai. Pēc katras uzmērīšanas reizes bebris saprot, vai uzmērītā kurpe pēc izmēra un/vai platuma der, un ja nē, vai nepieciešama mazāka vai lielāka un šaurāka vai platāka.



Bebri izmanto metodi, kas garantē, ka ar n uzmērīšanas reizēm tiks atrasta īstā kurpe.

Uzdevums:

Kāds ir mazākais iespējamais n skaits?

Pareizā atbilde: 3

Bebram var paveikties atrast kurpi ar pirmo reizi, tomēr pareizo izmēru viņš var atrast divos piegājienos.

-Viņš var sākt ar pašu centrālo kurpi, kā parādīts attēlā zemāk:

Width Size	Narrow <-----> Wide						
Big ↑ ↓ Small							
							

*narrow: šaurš; wide: plats; big: liels; small: mazs; width: platums; size: izmērs

-Šī kurpe vai nu būs pareizajā izmērā, mazāka, lielāka, vai nu pareizajā platumā, šaurāka vai platāka. Balstoties uz piemērotību, bebrs sapratīs, vai kurpe viņam tieši derēs, vai būs kādā no zemāk esošajiem 9 krāsu lauciņiem;

-Ja kurpe viņam der, viņš atradis īsto pāri;

-Ja kurpe ir mazāka, bet platāka, viņam būtu jāmēra kurpes 1.zonā;

-Ja kurpe ir mazāka, bet īstajā platumā, viņam būtu jāmēra kurpes 2.zonā;

-Ja kurpe ir lielāka un šaurāka, tad jāmēra kurpes 9.zona un tā tālāk

Width Size	Narrow <-----> Wide						
Big ↑ ↓ Small	A	B					
		1		2		3	
		4		5		6	
		7		8		9	

Pieņemsim, ka bebra uzmērītās kurpes ir mazākas un platākas, nekā viņam der. Tādā gadījumā viņam ir jāmēra lielākas un šaurākas kurpes no 1.zonas.

Tagad viņam ir jāmēra centrālā kurpe zonā 1 (atzīmēta ar #1).

- Ja kurpe viņam der, viņš ir atradis īsto kurpju pāri
- Ja kurpe joprojām par mazu un platu, A kurpe būs īstā

- Ja kurpe būs par mazu, bet īstajā platumā, B kurpe būs īstā.

Kā redzam, bebram ir jāizmēra kurpes vien 3 reizes, lai atradu sev nepieciešamo izmēru. Ja viņš mērīšanu sāks no jebkuras citas pozīcijas, jāmēra būs vairākkārt.

Kā tas ir saistīts ar informātiku?

Šajā uzdevumā kurpes veikalā ir sakārtotas augoši pēc izmēriem (vienā virzienā) un pēc platumiem (otrā virzienā). Meklēšanai sakārtotos datus efektīvi ir binārās meklēšanas algoritmi.

Šajos algoritmos ar katra vaicājuma palīdzību meklēšanas telpa tiek samazināta uz pusi, tādējādi ātri (logaritmiskā laikā) atrodot vajadzīgo.

Raksturīgs šādas meklēšanas piemērs ir pretinieka iedomāta skaitļa robežās no 1 līdz 100 atrašana, ja drīkst uzdot jautājumus formā “vai iedomātais skaitlis ir lielāks par N?” (kur N kāds skaitlis no 1 līdz 99) un iespējamās tikai atbildes “jā” vai “nē”. Tad optimālā jautājumu uzdošanas stratēģija ir sākt ar $N=50$ un, atkarībā no saņemtās atbildes, katrā nākamajā gājienā izvēlēties atlikušā skaitļu segmenta viduspunktu.

Šī uzdevuma īpatnība ir tā, ka meklēšana notiek divdimensiju datos.

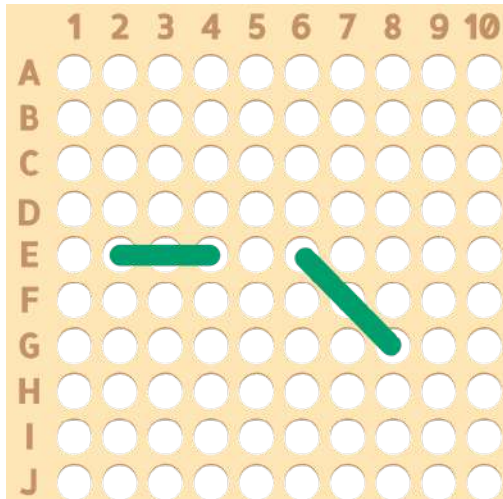
Izšuvums

Dienvidkoreja

Bebrs cenšas veidot izšuvumu, izmantojot speciālu programmu un mašīnu. Šī programma izmanto komandu ĀRĀ(cc)-IEKŠĀ(dd), kur cc un dd norāda uz adatas pozīciju režģī.

Piemēram, ĀRĀ(B2)-IEKŠĀ(A3) ir komanda, kas liek adatu novietot uz B2. Tad tā no aizmugures uz priekšu ir jāizvelk šajā lauciņā. Pēc tam adata ir jāpārvieta uz A4, kur tā ir jāiedur no priekšas uz aizmuguri.

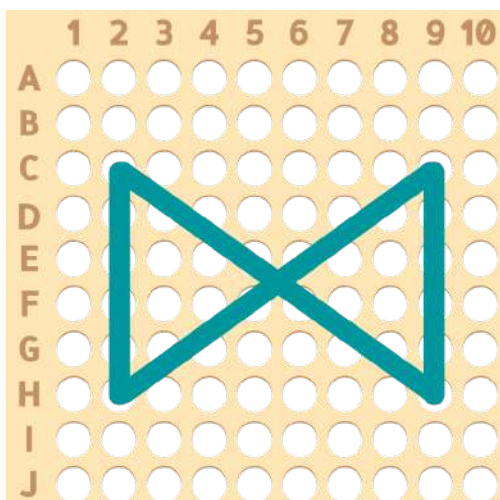
Zemāk redzamās komandas izveido šādu rakstu:



ĀRĀ(E6)-IEKŠĀ(G8);ĀRĀ(E2)-IEKŠĀ(E4)

Uzdevums:

Kādas komandas ir jāizmanto, lai izveidotu šādu rakstu?



A) ĀRĀ(H2)-IEKŠĀ(C2);ĀRĀ(H9)-IEKŠĀ(C9);ĀRĀ(C9)-IEKŠĀ(C2);ĀRĀ(H9)-IEKŠĀ(C2)

- B) $\bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(H9); \bar{A}\bar{R}\bar{A}(H2)-IEK\check{S}\bar{A}(C9); \bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(H2); \bar{A}\bar{R}\bar{A}(C9)-IEK\check{S}\bar{A}(H9)$
 C) $\bar{A}\bar{R}\bar{A}(H9)-IEK\check{S}\bar{A}(C9); \bar{A}\bar{R}\bar{A}(H9)-IEK\check{S}\bar{A}(H2); \bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(H2); \bar{A}\bar{R}\bar{A}(C9)-IEK\check{S}\bar{A}(H2)$
 D) $\bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(C9); \bar{A}\bar{R}\bar{A}(H2)-IEK\check{S}\bar{A}(H9); \bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(H2); \bar{A}\bar{R}\bar{A}(C9)-IEK\check{S}\bar{A}(H9)$

Pareizā atbilde:

B) $\bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(H9); \bar{A}\bar{R}\bar{A}(H2)-IEK\check{S}\bar{A}(C9); \bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(H2); \bar{A}\bar{R}\bar{A}(C9)-IEK\check{S}\bar{A}(H9)$

Lai izveidotu attiecīgo rakstu, mums ir nepieciešamas 4 komandas - vienu katrai no četrām līnijām nenoteiktā secībā. Komandas ir

$\bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(H9)$ vai $\bar{A}\bar{R}\bar{A}(H9)-IEK\check{S}\bar{A}(C2)$

$\bar{A}\bar{R}\bar{A}(H2)-IEK\check{S}\bar{A}(C9)$ vai $\bar{A}\bar{R}\bar{A}(C9)-IEK\check{S}\bar{A}(H2)$

$\bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(H2)$ vai $\bar{A}\bar{R}\bar{A}(H2)-IEK\check{S}\bar{A}(C2)$

$\bar{A}\bar{R}\bar{A}(C9)-IEK\check{S}\bar{A}(H9)$ vai $\bar{A}\bar{R}\bar{A}(H9)-IEK\check{S}\bar{A}(C9)$

Izvēle, kurā bija visas četras komandas, bija atbildē B.

A nav pareizi, jo tajā ir komanda $\bar{A}\bar{R}\bar{A}(C9)-IEK\check{S}\bar{A}(C2)$, kas rada nevajadzīgu līniju. Tāpat arī komanda $\bar{A}\bar{R}\bar{A}(H2)-IEK\check{S}\bar{A}(C9)$ vai $\bar{A}\bar{R}\bar{A}(C9)-IEK\check{S}\bar{A}(H2)$ ir pazudusi.

B ir pareizā atbilde.

C nav pareizi, jo tajā ir komanda $\bar{A}\bar{R}\bar{A}(H9)-IEK\check{S}\bar{A}(H2)$, kas rada nevajadzīgu līniju. Tāpat arī komanda $\bar{A}\bar{R}\bar{A}(C9)-IEK\check{S}\bar{A}(H9)$ vai $\bar{A}\bar{R}\bar{A}(H9)-IEK\check{S}\bar{A}(C9)$ ir pazudusi.

D nav pareizi, jo tajā ir komanda $\bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(C9)$ un $\bar{A}\bar{R}\bar{A}(H2)-IEK\check{S}\bar{A}(H9)$, kas rada nevajadzīgas līnijas. Tāpat arī komanda $\bar{A}\bar{R}\bar{A}(C2)-IEK\check{S}\bar{A}(H9)$ vai $\bar{A}\bar{R}\bar{A}(H9)-IEK\check{S}\bar{A}(C2)$ un $\bar{A}\bar{R}\bar{A}(H2)-IEK\check{S}\bar{A}(C9)$ vai $\bar{A}\bar{R}\bar{A}(C9)-IEK\check{S}\bar{A}(H2)$ ir pazudusi.

Kā tas ir saistīts ar informātiku?

Algoritms apraksta uzdevuma paveikšanai izpildāmo soļu virkni. Lai gan jēdziens "algoritms" parasti saistās ar datorzinātņi, tomēr tie ir atrodamī arī ikdienas darbos. Īpaši, ja nepieciešams šos darbus veikt efektīvi. Šajā uzdevumā algoritma pielietošana ir demonstrēta izšūšanas mašīnai veidojot vēlamo rakstu.

Senā valoda

Kanāda

Pētnieki uz alas sienas atklāja piecus senus vārdus:

paqrooob

puue

t'seqrub

meoub

lai'laigy

Lai noteiktu, pie kādas valodas pieder katrs no vārdiem, tiek izmantota punktu sistēma. Katram vārdam sākotnēji tiek piešķirti 10 punkti.

Tad punktu skaits var tikt mainīts, izmantojot šādus nosacījumus:

Sākas ar burtu "p"	-2
Beidzas ar burtu "b"	-2
Satur vairāk par 6 rakstzīmēm	+3
Satur burtu "q", aiz kura uzreiz seko burts "r" vai burts "y"	-4
Vārdā ir trīs patskaņi (burti "a","e","i","o" vai "u") pēc kārtas	+5
Satur rakstzīmi - apostrofu (')	+1

Ja rezultātā vārdam ir 10 vai vairāk punktu, sistēma nosaka, ka vārds pieder pie seno bebriešu valodas. Ja tā nav - pie koknešu valodas.

Piemēram, vārdam palliob ir deviņi punkti: $10-2-2+3=9$. Tātad vārds pieder pie koknešu valodas.

Uzdevums:

Cik daudz vārdu uz alas sienas pieder koknešu valodai? (Atbildi rakstīt tikai ar skaitļiem, bez nekādām citām rakstu zīmēm)

Pareizā atbilde: 1

Tabula zemāk apkopo iegūtos punktus katram vārdam:

Vārds	Punkti	Valoda
paqrooob	$10-2-2+3-4+5=10$	Seno bebiešu
puue	$10-2+5=13$	Seno bebiešu
t'seqrub	$10-2+3-4+1=8$	Koknešu
meoub	$10-2+5=13$	Seno bebiešu
lai'laigy	$10+3-4+1=10$	Seno bebiešu

Kā redzams - tikai viens vārds (t'seqrub) pieder koknešu valodai.

Kā tas ir saistīts ar informātiku?

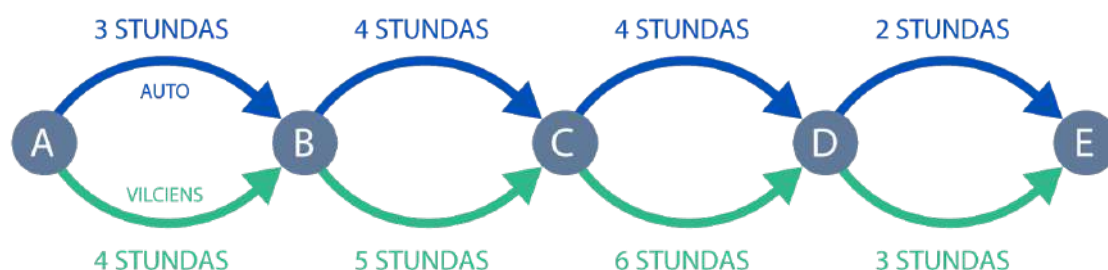
Datorzinātņu pielietošana dabīgās valodas pētniecībā - datorlingvistika ir aizraujoša zinātnes nozare, kurā joprojām noris aktīva pētniecība. Viena no pētījumu jomām ir automātiska teksta pārvēršana balss ziņojumos vai otrādi. Daudzi no mums šo tehnoloģiju jau izmanto ikdienā lietojot viedierīces vai mājas elektroniku. Otrs raksturīgs piemērs ir teksta tulkošana no vienas valodas citā. Ja valoda, no kuras nepieciešams tulkot, nav zināma, tad pirmais tulkošanas solis ir noskaidrot, kas tā par valodu. Un tieši šī uzdevuma risinājums vienkāršodā veidā ir parādīts dotajā uzdevumā.

Dabīgās valodas uzdevumu risināšana vispārīgā gadījumā ir ļoti grūtas un to perfekti atrisinājumi joprojām nav atrasti. Šajā uzdevumā tiek lietotas heuristikas, kas ar vienkāršu likumu palīdzību ļauj analizēt vārdus, cerībā, ka lielā skaitā gadījumu šī metode dos derīgas atbildes. Dažkārt pat vienkāršu heuristiku lietošana var dot apbrīnojami labus rezultātus. Psihologi strādā pie programmām, kas ļautu noteikt teksta rakstītāja vecumu un dzimumu, kā arī noteikt, ja kāds melo.

Saudzēsim vidi

Holande

Bebram Jānim ir jāceļo no pilsētas A uz pilsētu E caur B, C un D. Ceļojot no vienas pilsētas uz otru, viņš var izmantot vilcienu. Vilciena braukšanas ilgums ir redzams attēlā zemāk. Vilciena vietā viņš katrā pilsētā var īrēt mašīnu. Braukšanas ilgums redzams attēla augšpusē.



Jāni uztrauc vide, tāpēc viņš vēlas ceļot ar mašīnu pēc iespējas mazāk. Tā kā viņam galapunktā E ir jāierodas 15 stundu laikā, tad viņš ir spiests vismaz daļu attāluma ceļot ar mašīnu.

Uzdevums:

Kāds ir mazākais laiks, kuru Jānis var pavadīt, ceļojot ar mašīnu, lai galamērķī nonāktu laikā?

Pareizā atbilde: 6 stundas

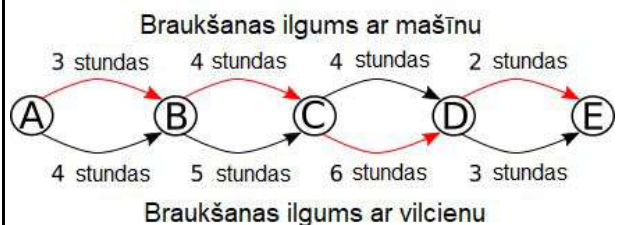
Jānis dod priekšroku ceļošanai ar vilcienu. Ja viņš visu ceļojumu veiktu ar vilcienu, tas aizņemtu 18 stundas. Tāpēc viņam ir jāaizvieto vismaz 3 stundas vilciena brauciena ar braucieni mašīnā. Mēs varam saskaitīt visas iespējas, lai to būtu iespējams izdarīt.

	Ietaupītais laiks, braucot ar mašīnu vilciena vietā	Brauciena ilgums
No A uz B	1 stunda	3 stundas
No B uz C	1 stunda	4 stundas
No C uz D	2 stundas	4 stundas
No D uz E	1 stunda	2 stundas

Ir jāsamazina kopējais braukšanas ilgums tā, lai ietaupītas sanāktu vismaz trīs stundas. To ir iespējams veikt tikai ar pārsēšanos.

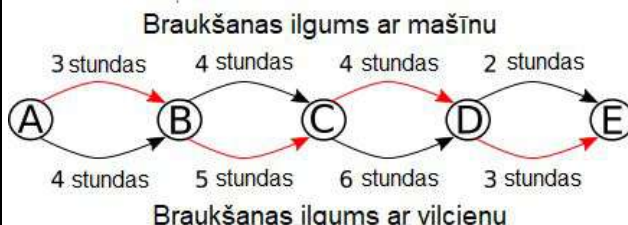
Ir iespējami četri risinājumi. Visos gadījumos ceļā jāpavada 15 stundas, lai nokļūtu no punkta A līdz punktam E. Ir jāizvēlas viens no braukšanas veidiem, kurā vismazāk laika tiek pavadīts braucot ar mašīnu.

Pirmais: 9 stundas ar mašīnu



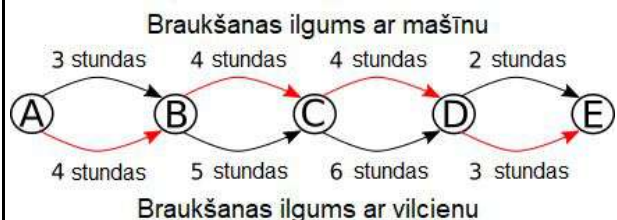
No/uz	Mašīna	Vilciens	Stundas
A uz B	Mašīna 3 stundas		3
B uz C	Mašīna 4 stundas		4
C uz D		Vilciens 6 stundas	6
D uz E	Mašīna 2 stundas		2
Pavadītais laiks	9 Stundas	6 Stundas	15

Otrais: 7 stundas ar mašīnu



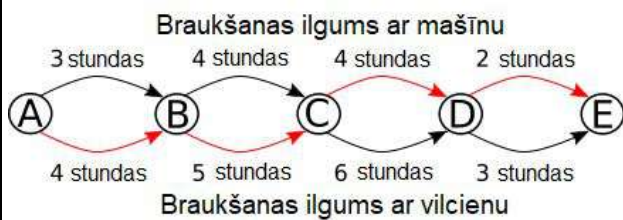
No/uz	Mašīna	Vilciens	Stundas
A uz B	Mašīna 3 stundas		3
B uz C		Vilciens 5 stundas	5
C uz D	Mašīna 4 stundas		4
D uz E		Vilciens 3 stundas	3
Pavadītais laiks	7 Stundas	8 Stundas	15

Trešais: 8 stundas ar mašīnu



No/uz	Mašīna	Vilciens	Stundas
A uz B		Vilciens 4 stundas	4
B uz C	Mašīna 4 stundas		4
C uz D	Mašīna 4 stundas		4
D uz E		Vilciens 3 stundas	3
Pavadītais laiks	8 Stundas	7 Stundas	15

Ceturtais: 6 stundas ar mašīnu



No/uz	Mašīna	Vilciens	Stundas
A uz B		Vilciens 4 stundas	4
B uz C		Vilciens 5 stundas	5
C uz D	Mašīna 4 stundas		4
D uz E	Mašīna 2 stundas		2
Pavadītais laiks	6 Stundas	9 Stundas	15

No A uz E ir iespējams nokļūt arī ātrāk, bet tādā gadījumā Jānim ir jāpavada vairāk laika braucot ar mašīnu.

Kā tas ir saistīts ar informātiku?

Šāda veida uzdevumi tiek saukti par optimizācijas uzdevumiem. Ir dots noteikts ierobežojumu kopums (šajā gadījumā - ceļojumam kopējais atvēlētais laiks) un ir kāds lielums, kuru pie šiem ierobežojumiem nepieciešams maksimizēt vai minimizēt (šoreiz nepieciešams minimizēt mašīnas izmantošanas laiku).

Lai atrisinātu šāda veida uzdevumus, bieži ir izdevīgi to pārfrāzēt vieglāk atrisināmā uzdevumā - kā tas ir darīts arī šoreiz.

Dotais uzdevuma risinājums ir saistīts arī ar datordomāšanu. Uzdevumā dotie dati tiek strukturēti tabulu veidā, kur pirmajā tiek aprēķināts katrā posmā ietaupītais laiks, ceļojot ar mašīnu vilciena vietā. Pēc tam šī informācija tiek izmantota, lai iegūtu četrus derīgus risinājumus un atrastu optimālo (ar mazāko mašīnas izmantošanas laiku).

Bišu strops

Pakistāna

Biškopim ir strops ar bitēm. Viņš vēlas stropu novietot tā, lai attāluma summa starp stropu un līdz katrai no puķēm būtu pēc iespējas mazāka. Lauks ar ziediem ir attēlots zemāk redzamajā rūtiņu tabulā, kuras rindas ir sanumurētas no 1 līdz 9 un kolonnas apzīmētas ar burtiem no A līdz I.

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

Bites šajā laukā lido tikai horizontāli un vertikāli, tāpēc attālums starp divām rūtīm ir horizontālā un vertikālā attāluma summa. Piemēram, attālums starp C4 un D7 ir 4 (3 rūtis vertikāli un 1 rūtis horizontāli).

Uzdevums:

Kur biškopim būtu jānovieto strops, lai attālumu summa no stropa līdz katrai puķei ir pēc iespējas mazāka? (iespējamās vietas kartē ir iezīmētas ar )

Atbilžu varianti:

- A) D5
- B) C7
- C) E5
- D) A9

Pareizā atbilde: A) D5

Problēma var tik atrisināta, atrodot puķu skaita virkņu mediānas horizontāli un vertikāli. Tā kā kopā šajā laukā ir 17 puķes, vidējā puķe sanāk devītā pēc kārtas (atstājot 8 puķes

vienā un 8 puķes otrā pusē). Sakārtojot vertikāli, mediāna ir puķe rūtī A5, tātad strops būtu jāievieto 5.rindā. Sakārtojot horizontāli, mediāna sanāk vai nu puķe rūtī D3, vai D9, jo tās abas ir vienā kolonnā. Tātad strops jāieliek D kolonnā. Tas mūs noved līdz optimālajai stropa novietošanas vietai, kas ir D5.

Tā kā par attālumu ir uzskatīta vertikālā un horizontālā attāluma summa (saukta arī par Manhetenas attālumu, nevis taisnā līnijā (Eiklīda attālums)), problēma var tikt atrisināta, atsevišķi atrodot stropa novietojuma kolonnu un atsevišķi - rindu, jo novietojums vienā virzienā neietekmē kopējo attālumu otrāi. Turklāt algoritms, kas tiek izmantots optimālās kolonnas atrašanai, var tikt izmantot arī rindas atrašanai, un tādā veidā optimālā stropa atrašanās vieta būs tā, kur abas šīs koordinātas tiks sakombinētas.

Kā tas ir saistīts ar informātiku?

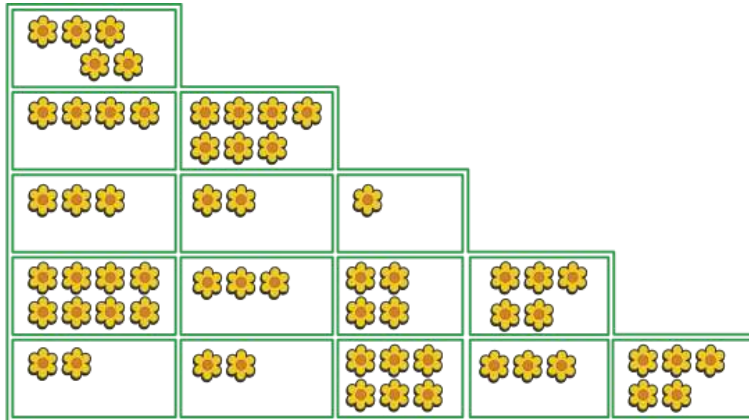
Šī uzdevuma paredzētais risināšanas veids (nerēķinot visus iespējamo ceļu garumus) izmanto "skaldi un valdi" pieeju, jo sākotnējais uzdevums tiek sadalīts divos apakšuzdevumos: atrast optimālo rindu un atrast optimālo kolonu. Pēc tam, kad abi apakšuzdevumi ir atrisināti, iegūtie rezultāti tiek apvienoti vienā kopējā rezultātā.

https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm

Sarkangalvīte

Rumānija

Sarkangalvīte vēlas salasīt puķes no vecmāmiņas dārza. Dārzs ir sadalīts vairākās daļās, kur katra daļa satur konkrētu puķu skaitu. Sarkangalvīte sāk savu ceļu no augšējā kreisā stūra un tad dodas lejup uz apakšējo labo stūri, ejot tikai uz leju vai pa labi.



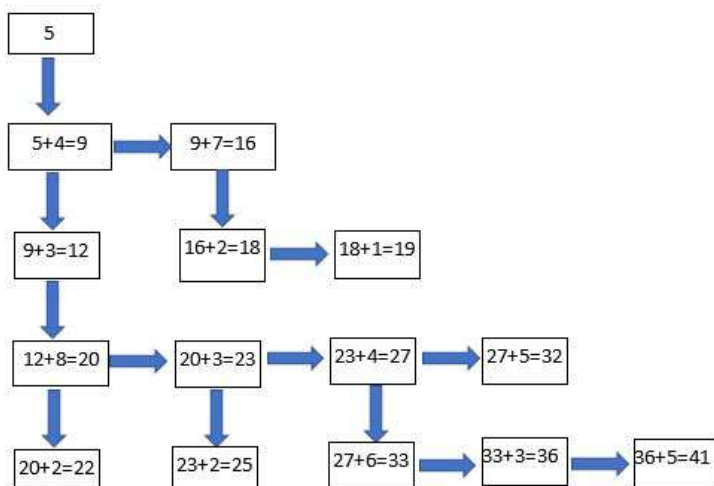
Uzdevums:

Kāds ir lielākais ziedu skaits, kurus viņa var salasīt sava gājiena laikā?

Pareizā atbilde: 41

Katrā daļā mēs varam izrēķināt maksimālo puķu skaitu, kuru Sarkangalvīte var iegūt no sākotnējās ailes līdz tai daļai (sauksim to par maksimumu). Maksimumu var izrēķināt šādi:

- Ja daļu var sasniegt tikai no vienas iepriekšējās daļas, tad tās maksimums ir ziedu skaita summa šajā daļā un iepriekšējās daļas maksimums.
- Ja daļu var sasniegt no divām iepriekšējām daļām, tad tās maksimums ir puķu skaits tajā daļā, kam pieskaitīts lielākais maksimums no divām iepriekšējām daļām.



Kā tas ir saistīts ar informātiku?

Šis ir dinamiskās programmēšanas uzdevums. Šādos uzdevumos labākais atrisinājums tiek iegūts, "pa ceļam" atrisinot šī uzdevuma apakšuzdevumus un šos risinājumus izmantojot gala rezultāta iegūšanai.

Šajā uzdevumā starprezultāti ir lielākais puķu skaits, ko Sarkangalvīte var savākt līdz katrai dārza daļai virzoties no sākuma līdz beigu daļai. Gudrība slēpjas tajā, ka nav nepieciešams šo skaitli rēķināt no jauna, bet var izmantot iepriekš iegūto rezultātu un lielāko līdz konkrētajai daļai savākto puķu skaitu var iegūt kā lielāko no divām summām, katrā solī veicot vienu salīdzināšanu un vienu saskaitīšanu. Ar šī algoritma palīdzību brīdī, kad nonāksim pēdējā daļā, mums jau būs aprēķināta vajadzīgā atbilde.

Daudzi praktiski optimizācijas uzdevumi tiek risināti ar šādu, laika ziņā efektīvu, risināšanas algoritmu.

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

Noliktavas

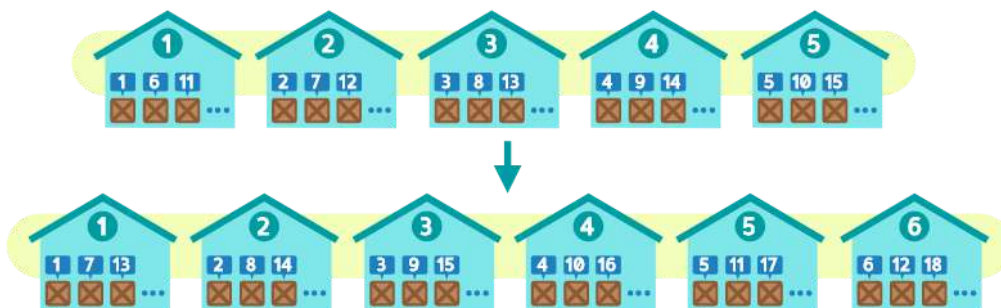
Krievija

42 eži novieto savas mantas 5 noliktavās. Pirmais ezis novieto savas mantas pirmajā noliktavā, otrais otrajā, ..., sestais atkal pirmajā noliktavā un tā tālāk.

Vienu dienu eži uzbūvēja jaunu noliktavu, un tā tika atzīmēta ar ciparu 6. Viņi nolēma pārvietot savas mantas noliktavās tā, lai sadalījums joprojām būtu vienkāršs. Pirmais ezis novieto savas mantas pirmajā noliktavā, otrais otrajā, ..., septītais atkal pirmajā noliktavā un tā tālāk.

Uzdevums:

Cik ežiem nebija jāpārvieto mantas jaunajā veidā?



Pareizā atbilde: 10

Eži numur 1, 2, 3, 4, 5 un 31, 32, 33, 34, 35 savas mantas nepārvietoja. Lai to noskaidrotu mums, pirmkārt, ir nepieciešams vienkāršot numerāciju: jānumurē eži un noliktavas no 0. Tādā veidā mēs iegūstam ežus no 0 līdz 41. Un ezim numur x nevajadzētu pārvest savas mantas, ja $x \bmod 5 = x \bmod 6$. Tas ir iespējams tad, ja $30 \mid x - r$, kur $r = x \bmod 5 = x \bmod 6$. Tātad, tas ir iespējams tikai ar $x = 0 + 0, 1, 2, 3, 4$, un $x = 30 + 0, 1, 2, 3, 4$.

Kā tas ir saistīts ar informātiku?

Šajā uzdevumā notiek sadalīta resursu glabāšana. Lai noteiktu, kur katrs resurss jānovieto vai jau atrodas, iespējams lietot jaučējadresēšanu jeb jaukšanu. Katram resursam tiek aprēķināta jaučējatslēga, kas norāda vietu, kur attiecīgais resurss jānovieto vai ir atrodams. Piemēram, ja resursi ir cilvēki, kuru vārdi mums ir zināmi, tad kā jaučējatslēgu var izmantot vārda pirmo burtu. Ja rodas vajadzība atrast Matīsu, tad vispirms atrodam, kādi cilvēki ar "M" mums ir zināmi (piemēram, vēl arī Matlīde, Mareks un Mikus) un tad informāciju par Matīsu meklējam jau šajā ierobežotajā (visiem viena jaučējatslēga) kopā.

Uzdevumā jaučējatslēga tiek aprēķināta kā atlikums, dalot ar veselu skaitli, un atslēgas vērtība tiek mainīta līdz ar jaunās noliktavas uzcelšanu. Labas jaučējfunkcijas, kuras datu tabulas lieluma (jauna noliktava šajā uzdevumā) izmaiņu dēļ izsauc nepieciešamību pārvietot tikai ierobežotu datu apjomu, sauc par konsistentām jaučējfunkcijām.

Skaitītājs

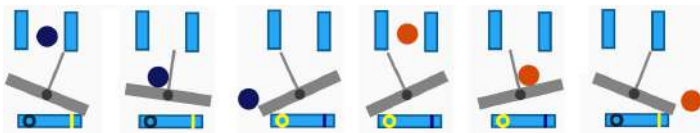
Austrija

Iekārtā ir 4 dēļiši, kas, līdzīgi kā šūpolēs, var būt noliekušies pa labi vai pa kreisi

Dēlītis, kas noliecas pa kreisi = 0

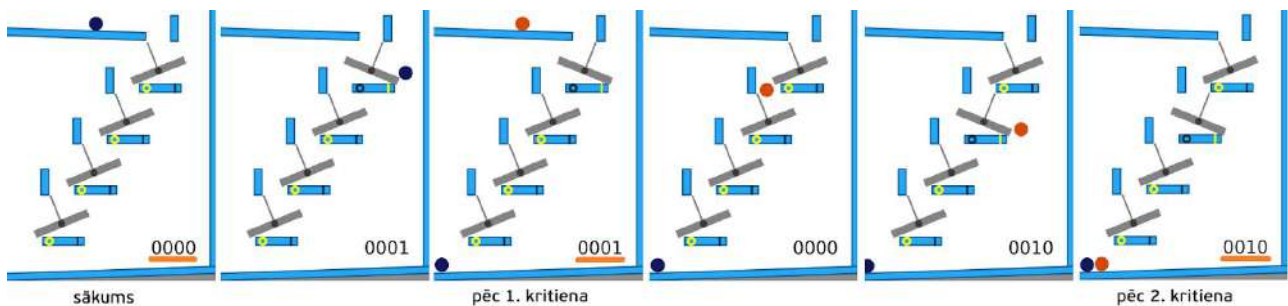
Dēlītis, kas noliecas pa labi = 1

Kad krītoša bumbiņa nonāk uz kāda no dēļiņiem, tas noliecas un bumbiņa noripo.



Zemāk redzama iekārtas darbība, kad tiek ripinātas 2 bumbiņas.

Pirmajā attēlā visi dēļiši ir pa kreisi, tāpēc skaitītājs rāda 0000:



Uzdevums:

Kāds būs skaitītāja rādījums pēc 5 nomestām bumbiņām?

Pareizā atbilde: 0101

Skaitītāja rādījumi mainās šādi: 0001, 0010, 0011, 0100, 0101

Šī ierīce ir binārais skaitītājs.

Sākuma pozīcija ir 0000.

Pirmā bumbiņa noliec 1.dēlīti pa labi. Skaitītājs rāda 0001.

Otrā bumbiņa noliec 1.dēlīti pa kreisi, bet 2. - pa labi. Skaitītājs rāda 0010.

Trešā bumbiņa noliec 1.dēlīti pa labi un noripo. 2.dēlītis paliek noliecas pa labi. Skaitītājs rāda 0011.

Ceturtā bumbiņa 1.dēlīti noliec pa kreisi, 2.dēlīti arī pa kreisi, bet 3.dēlīti pa labi. Skaitītājs rāda 0100.

Pēdējā bumba 1.dēlīti noliec pa labi un noripo nost. 2.dēlītis paliek noliecas pa kreisi, bet 3. - noliecas pa labi. Skaitītājs rāda 0101.

Atbilde jāsniedz kā četru ciparu (katrs 0 vai 1) virkne.

Kā tas ir saistīts ar informātiku?

Ja skaitļu pierakstam tiek izmantoti tikai divi cipari (parasti - 0 un 1), tas nozīmē, ka tiek izmantota divnieku jeb binārā skaitīšanas sistēma, kas jau no pirmsākumiem ir ļoti populāra datoros, jo viegli iedomāties un tehniski realizēt situāciju, kad katrs elements ir tikai vienā no diviem stāvokļiem - ieslēgts (1) vai izslēgts (0).

Šajā uzdevumā aprakstītā mašīna apraksta mehānisku bināru četrциparu skaitītāju, kas var skaitīt līdz 15. Visi četrциparu binārie skaitļi ir uzskaitīti tabulā:

Binary number	Number
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

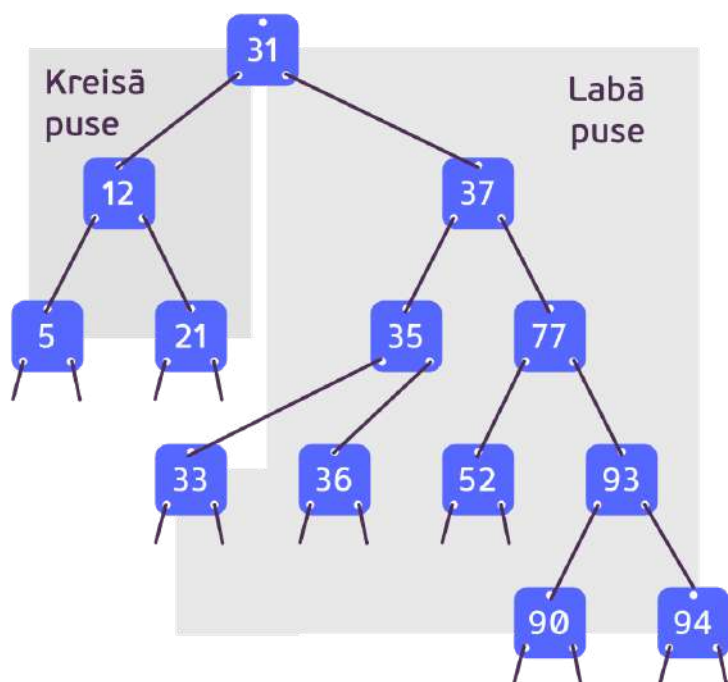
**Binary number: binārais skaitlis; Number: skaitlis.*

Varat noskatīties video, kā šim līdzīgs binārs skaitītājs darbojas:

https://www.youtube.com/watch?v=zELAfmp3fXY&fbclid=IwAR1iVWNrmrj-bNzD26i5xNivAaN2Ue7o_AWkRtlpRGamiHFEsOfigpRWizQ

Numurētās kārtis

Holande



No numurētām kārtīm ir izveidota struktūra. Katra kārts ir veidota šādi:

Tai augšpusē ir caurums, bet apakšā divas virves, pie kurām var piestiprināt citas kārtis, dēvētas par pēcteča kārtīm.

Uz katras kārts ir uzrakstīts vesels skaitlis N .

Ja numurētā kārts ir piestiprināta pie kreisās virves, tās un tās pēcteču kāršu numuriem ir jābūt mazākiem par skaitli N .

Ja numurētā kārts ir piestiprināta pie labās virves, tās un tās pēcteču kāršu numuriem ir jābūt lielākiem par skaitli N .

Uzdevums:

Šobrīd struktūrā kārtīm kopā ir 14 brīvas virves. Pie cik no šīm virvēm lielākais vēl var pievienot kārtis, lai pagarinātu doto struktūru?

Pareizā atbilde: 11

Ir 14 tukšas virves. Kārtij ar numuru 36 vairs nevar pievienot nevienu citu kārti. Kreisās pēcteča kārts numuram būtu jābūt lielākam par 35 un mazākam par 36, bet šāds vesels skaitlis neeksistē. Labās puses pēcteča kārtij būtu jābūt lielākam par 36 un mazākam par 37, bet arī šāds vesels skaitlis neeksistē. Kārtij ar numuru 94 var pievienot tikai labo pēcteča kārti.

Kā tas ir saistīts ar informātiku?

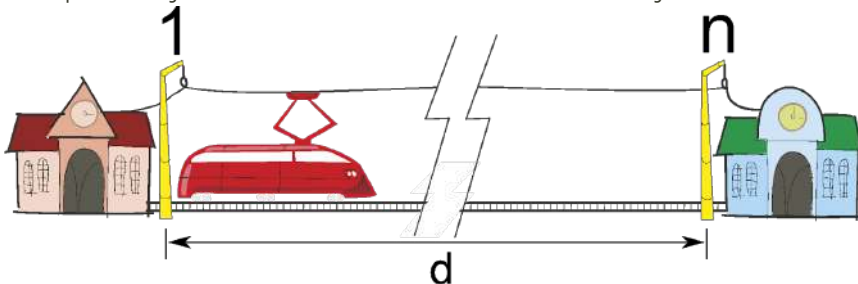
Uzdevumā aprakstītā struktūra ir pazīstama kā binārais meklēšanas koks. Tas izskatās kā otrādi apgriezts koks. Par bināru to sauc tāpēc, ka katrai kārtij ir piestiprinātas tieši divas virves (priedēklis "bi-" nāk no latīņu valodas ar nozīmi "divi") un, tātad, katrai kārtij var piesaistīt ne vairāk kā divas citas kārtis. Šādās datu struktūrās var glabāt visdažādākā tipa datus un to priekšrocība ir iespēja ātri (logaritmiskā laikā) atrast tajā esošos datus.

Dzelzceļa robots

Čehija

Dzelzceļa līnija starp divām stacijām ir apgādāta ar elektrību. Mastiem, kas piegādā elektrību, ir jābūt novietotiem regulāros intervālos - pirmajam mastam jāatrodas pirmajā stacijā, bet pēdējam - otrajā.

Dzelzceļa robots darbojas pēc programmas, kas redzama zemāk. Attālums starp stacijām ir d metri, un robotam ir jānovieto n masti.



Uzdevums:

Pievieno derīgos parametrus programmai tā, lai robots varētu strādāt pareizi!

Aizvelc pareizo izteiksmi uz pareizo kastīti programmā pa labi!

Atbildes:



n	$n+d$
d	$n-d$
1	$d-n$
2	$d*n$
$n+1$	n/d
$d+1$	d/n
$n-1$	$n/(d+1)$
$d-1$	$d/(n+1)$
$n/2$	$n/(d-1)$
$d/2$	$d/(n-1)$



Pareizā atbilde: n , $n-1$, $d/(n-1)$

Ja robotam ir jānovieto n masti, viņam sākumā ir arī jāuzlādē n masti. Pēc tam, kad viens masts ir novietots, $n-1$ masti ir palicis pāri. Tātad robotam ir jāatkārto $n-1$ reizes pārvietošanās un viena masta novietošana. Tad mums ir jāsadala attālums d ar $(n-1)$, lai

saprastu, cik daudz robotam ir nepieciešams pārvietoties, lai novietotu nākamo mastu, tādējādi $(n-1) * [d/(n-1)] = d$.

Kā tas ir saistīts ar informātiku?

Uzdevumā dots koda fragments ir procedūra, kurā tiek izmantoti mainīgie d un n . Mainīgā vārds apzīmē kādu skaitlisku vērtību, kas laika gaitā var mainīties. Mainīgie d un n ir definēti iepriekš galvenajā programmā un tiek izmantoti procedūrā. Šādos gadījumos mainīgos sauc par globāliem mainīgajiem un tiem var piekļūt no atšķirīgām procedūrām šajā programmā.

Uzdevumā tiek izmantoti arī cikli - instrukciju virknes (cikla ķermenis), kas tiek atkārtotas vairākkārt. Šajā uzdevumā ir svarīgi noskaidrot, cik reizes cikla ķermenis jāizpilda. Šoreiz, lai uzstādītu visus mastus, cikla ķermenis jāizpilda $n-1$ reizi.

Procedūras un funkcijas ir būtisks programmēšanas valodu koncepts, tās tiek izmantotas specifisku apakšuzdevumu veikšanai un to darbības raksturu var vadīt ar tām nododamo parametru vērtībām.

Ķīmikālijas

Ungārija

Trīs zinātnieki     darbojas ar ķīmikālijām un viņu mērķis ir no četrām ķīmikālijām izveidot jaunu ķīmikāliju .

Katrs zinātnieks strādā divas dienas. Ik dienas strādā tikai viens zinātnieks, un viņš vai viņa var izpildīt tikai vienu darbību dienā.

Viena darbība nozīmē savienot divas ķīmikālijas, izveidojot jaunu, kuru pēc tam var izmantot jebkuram skaitam darbību nākotnē. Katrs zinātnieks var veikt tikai divu veidu darbības:

Ķīmiķe Karlija (C)	 +  → 	 +  → 
Fiziķe Paula (P)	 +  → 	 +  → 
Biologs Berijs (B)	 +  → 	 +  +  → 

Uzdevums:

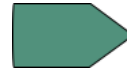
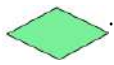
Sekojošā burtu secība norāda uz secību, kādā zinātnieki strādā. Kura no darbību secībām neļautu zinātniekiem izveidot  ķīmikāliju?

Atbilžu varianti:

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

Pareizā atbilde: C

Tā kā B (biologa Berija) otrajā darbībā izveidojas gala produkts, viņam jābūt pēdējam sarakstā. Lai izveidotu gala produktu, izmantojot savu otro darbību, viņam būtu jābūt



Lai izveidotu melno apli, C (ķīmiķei Karlijai) ir jābūt pieejai un ir jādodas uz laboratoriju pēc fiziķes Paulas. Tāpat arī C ir jāveic sava pirmā darbība pirms otrās darbības. Lai veiktu savu pirmo darbību, B ir nepieciešams, lai pirms viņa P izpilda savu pirmo darbību.

Atbildē "C" pirmais P secībā var izveidot tikai vienu no nepieciešamajām ķīmikālijām, kas nepieciešamas nākamā darbību veikšanai. Vienai no tām ir jāizgāžas. Secība C P B C P B veiksmīgi nevarēs izveidot nepieciešamo produktu.

Kā tas ir saistīts ar informātiku?

Datorzinātnē un programmēšanā tiek veidoti un izmantoti procesi, kuriem ir noteikta veida ievaddati un pēc kuru izpildes tiek iegūti noteikta veida izvaddati. Ja viens process ir atkarīgs no cita procesa rezultāta, tad šo procesu nedrīkst sākt, ja tam nepieciešamie dati vēl nav sagatavoti. Programmētājam ir jānodrošina tāda procesa vadība, kuras laikā visi procesi tiek uzsākti tikai tad, kad iepriekšējie procesi ir beigušies un visi nepieciešamie dati ir sagatavoti.

Robota zīmējums

Slovākija

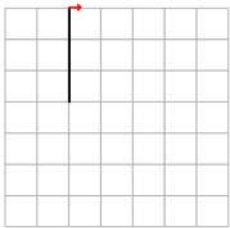
Robots, kustoties pa kvadrātveida režģi, ar līnijām veido attēlu. Katru attēlu apraksta trīs ciparu virkne.

Piemēram, "3,1,5" apraksta 4.zīmējumā redzamo figūru, jo nozīmē:

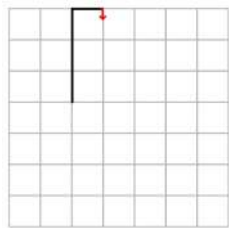
Pārvietoties 3 vienības (rūtiņas malas garums) uz priekšu, tad pagriezties pa labi (1.zīm.)

Pārvietoties 1 vienību uz priekšu, tad pagriezties pa labi (2.zīm.)

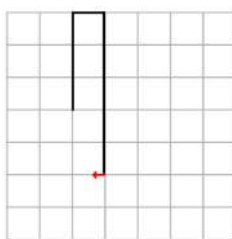
Pārvietoties 5 vienības uz priekšu, tad pagriezties pa labi (3.zīm.).



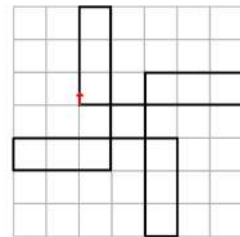
1.zīmējums



2.zīmējums



3.zīmējums



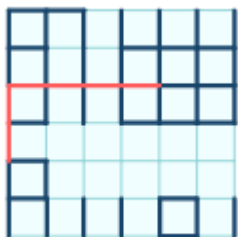
4.zīmējums

Dotā darbību virkne tiek atkārtota bezgalīgi.

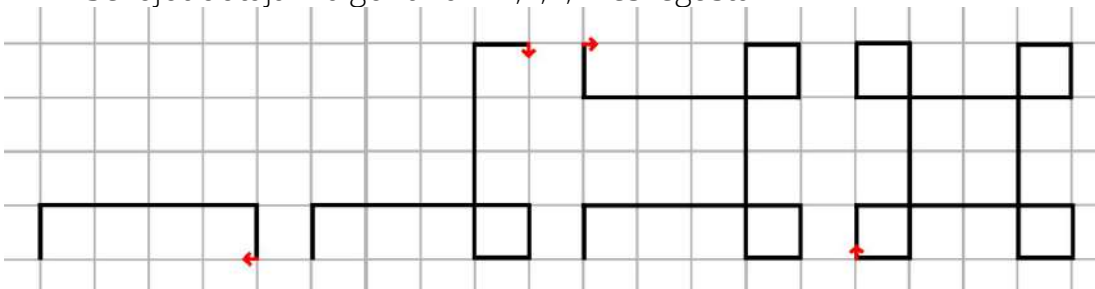
Uzdevums:

Robots ir sācis zīmēt figūru, kuru apraksta "2,4,3". Klikšķini uz līnijām, lai uzzīmētu šo figūru pilnībā!

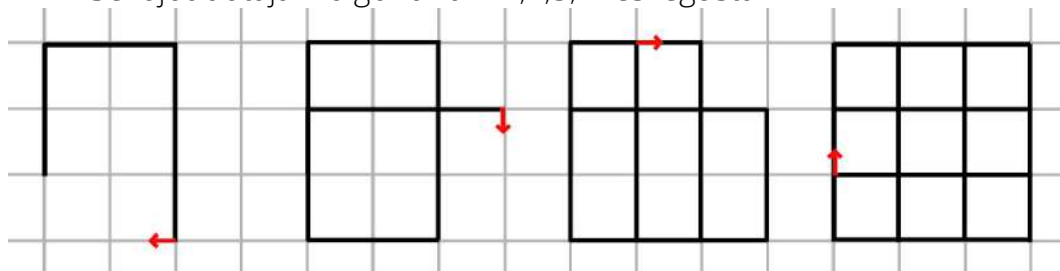
Pareizā atbilde:



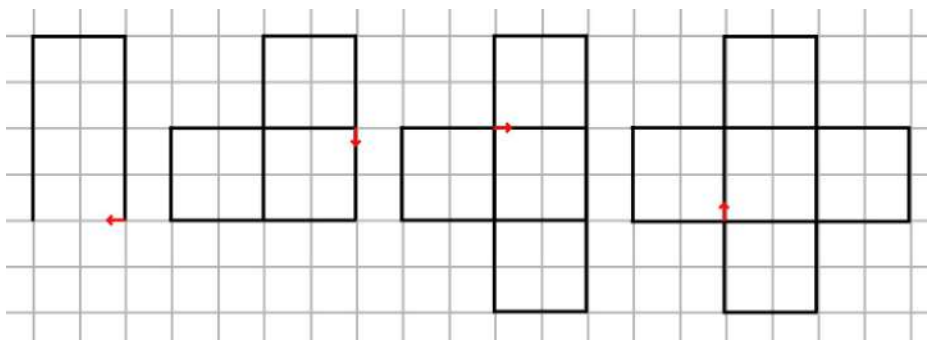
Sekojošot dotajam algoritmam 1,4,1, mēs iegūstam



Sekojošajam algoritmam 2,2,3, mēs iegūstam



Sekojošajam algoritmam 4,2,4, mēs iegūstam



Sekojošajam algoritmam 3,3,3, mēs iegūstam 3x3 izmēra kvadrātu.

Kā tas ir saistīts ar informātiku?

Datorprogrammu izpildes gaitā datorā tiek izpildītas datorprogrammas instrukcijas. Katra instrukcija satur noteiktu veicamās darbības aprakstu. Darbības izpilde izraisa noteiktu efektu, kas saistīts ar izpildītās darbības semantiku.

([https://en.wikipedia.org/wiki/Execution_\(computing\)](https://en.wikipedia.org/wiki/Execution_(computing)))

Katra uzdevuma atrisināšana ietver instrukciju semantikas un aprakstītā algoritma izpildes gaitas izpratni, kas ir neatņemama kvalitatīvas programmēšanas sastāvdaļa.

Jums jāspēj izlasīt un saprast instrukciju aprakstu, kā arī izpildīt tās soli pa solim, lai pārliecinātos par izraisītajiem efektiem. Šī ir ļoti būtiska laba programmētāja spēja, kas diendienā tiek izmantota datorprogrammu atklāšanas procesā.



Copyright Bebras