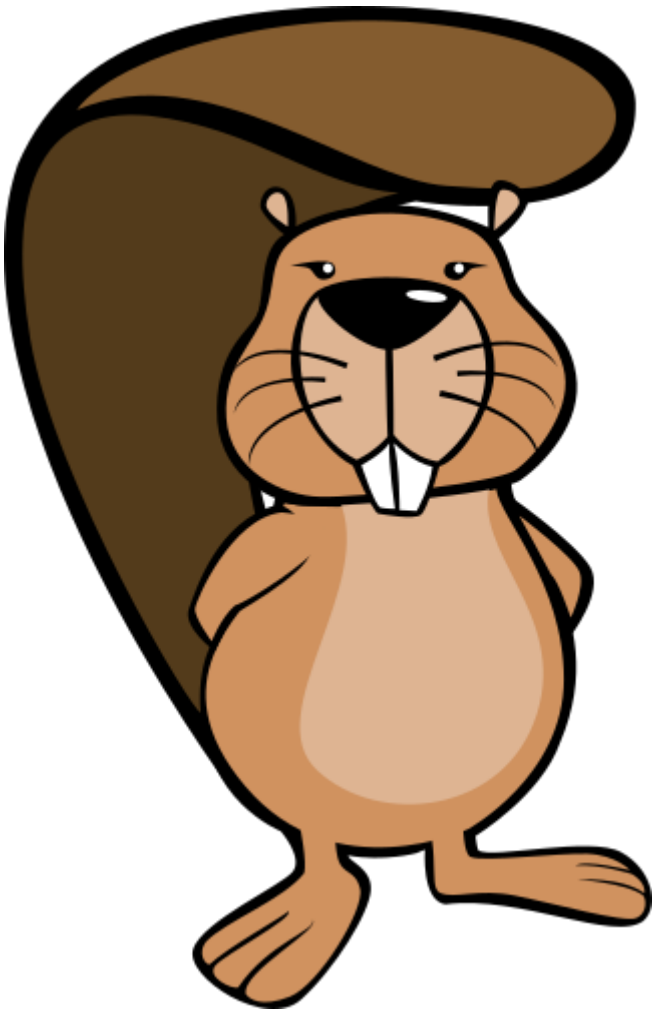
Bober 2022/23

Bebras

Naloge za tekmovanje je izbral, prevedel, priredil in oblikoval Programski svet tekmovanja:

Alenka Kavčič (UL FRI)
Anja Kneževič (OŠ Gradec)
Anja Koron (OŠ Branik)
Janez Demšar (UL FRI)
Nežka Rugelj (OŠ Gradec)
Špela Cerar (UL PEF)



Naloge z državnega tekmovanja in rešitve

ACM Slovenija

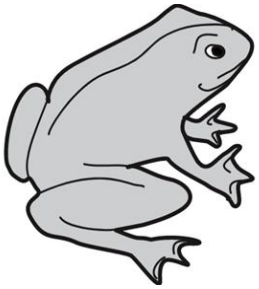
Naloge za tekmovanje je izbral, prevedel, priredil in oblikoval Programski svet tekmovanja:
Alenka Kavčič (UL FRI)
Anja Kneževič (OŠ Gradec)
Anja Koron (OŠ Branik)
Janez Demšar (UL FRI)
Nežka Rugelj (OŠ Gradec)
Špela Cerar (UL PEF)

Razvoj tekmovalnega sistema:
Gašper Fele Žorž (UL FRI)
Gregor Jerše (UL FRI)

Kazalo nalog

Pobarvaj žabo

državno, 6. in 7. razred



Na računalniškem zaslonu se prikaže risba žabe. Barva žabe ni določena. Za spreminjanje barve so na voljo štirje programi, vendar lahko zaženete le enega. Vsak program je sestavljen iz zaporedja stavkov če-potem-sicer ali če-potem; vsak stavek, ki sledi prvemu, se mora izvesti, ko se zaključi stavek pred njim.

Zagotoviti želite, da bo žaba na koncu izvajanja zelena, ne glede na to, kakšno barvo je imela pred izvajanjem programa.

Kateri program na koncu svojega izvajanja NE zagotovi zelene barve?

A)

če je žaba rdeča, **potem** jo pobarvaj rumeno, **sicer** jo pobarvaj zeleno;

če je žaba rumena, **potem** jo pobarvaj rdeče;

če žaba ni rumena, **potem** jo pobarvaj zeleno.

B)

če je žaba rdeča, **potem** jo pobarvaj rumeno;

če žaba ni rdeča, **potem** jo pobarvaj zeleno;

če je žaba rumena, **potem** jo pobarvaj rdeče, **sicer** jo pobarvaj zeleno.

C)

če je žaba rumena, **potem** jo pobarvaj zeleno;

če žaba ni rumena, **potem** jo pobarvaj rdeče;

če je žaba rdeča, **potem** jo pobarvaj zeleno, **sicer** jo pobarvaj rumeno.

D)

če je žaba rumena, **potem** jo pobarvaj zeleno, **sicer** jo pobarvaj rdeče;

če je žaba rdeča, **potem** jo pobarvaj zeleno, **sicer** jo pobarvaj rumeno.

Rešitev

Pravilni odgovor je program D: če je na začetku žaba rumena, se z izvedbo prvega navodila če-potem-sicer obarva zeleno, nato pa je pogoj "žaba je rdeča" napačen, zato se ponovno obarva rumeno.

Ob izvajanju programa A je po izvedbi navodila če-potem-sicer žaba rumena ali zelena; po izvedbi drugega navodila je rdeča ali zelena, po čemer je pogoj "žaba ni rumena" resničen in bo zato obarvana zeleno.

Če izvedete program B, bo ob izvedbi drugega navodila pogoj "žaba ni rdeča" zagotovo resničen, zato bo žaba obarvana zeleno, nakar jo tretji stavek (če-potem-sicer) spet obarva zeleno.

Če zaženete program C, bo ob izvajanju drugega navodila pogoj "žaba ni rumena" zagotovo resničen, zato bo žaba obarvana rdeče, nakar jo bo tretji stavek (če-potem-sicer) obarval zeleno.

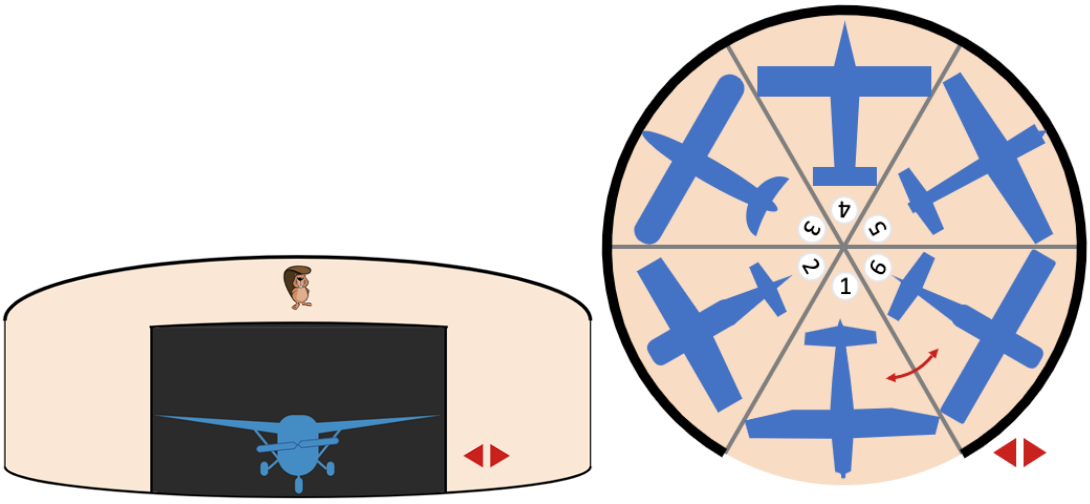
Računalniško ozadje

Pri programiranju pogosto naletimo na pogojne stavke in preverjanje pogojev. Pri sestavljanju programov morajo biti programerji zelo pozorni, da pogoje za izvedbo posameznega dela programa natančno določijo, sicer lahko dobijo nepredvidene rezultate.

Vrtiljak letal

državno, 6. in 7. razred

Na letališču Bobrovo je v okroglem hangarju na vrtljivi ploščadi parkiranih šest letal. Vrtljivo ploščad nadzornik Damjan vrti v levo ali desno s pomočjo nadzorne plošče z dvema puščicama ◄►. Z enim pritiskom na gumb se ploščad zavrti za natanko en parkirni položaj levo ali desno. Vrata hangarja so dovolj široka, da iz njega lahko odpelje eno letalo. Vrtljiva ploščad se vrti zelo počasi, zato se z manjšim številom pritiskov na gumb izognemo zamudam letal.



Zjutraj, ko piloti pridejo po svoja letala, je parkirni položaj 1 vedno pri vratih. V najboljšem primeru je treba puščične tipke pritisniti petkrat, da vsa letala zapustijo hangar. To bi se zgodilo, če bi piloti do parkirnih mest dostopali po vrstnem redu: 1, 2, 3, 4, 5, 6 s petkratnim pritiskom ► ali po vrstnem redu: 1, 6, 5, 4, 3, 2 s petkratnim pritiskom ◄.

Kolikokrat bo moral nadzornik Damjan pritisniti puščične tipke v najslabšem primeru?

Rešitev

V najslabšem primeru bo moral nadzornik Damjan 16-krat pritisniti na puščične tipke. To bi se zgodilo v primerih, ko bi piloti do parkirnih mest dostopali po vrstnem redu 4, 1, 3, 6, 2, 5 ali 4, 1, 5, 2, 6, 3.

Najslabši primer najdemo tako, da izberemo naslednje parkirno mesto, ki je vedno najbolj oddaljeno od vrat. Ker lahko vrtljivo ploščad premikamo v obe smeri, obstaja več pravilnih rešitev.

Prva možnost je 4, 1, 3, 6, 2, 5 :

Za prvi dostop do položaja 4 so potrebni trije pritiski (levo ali desno).

Naslednji dostop do položaja 1 zahteva tri pritiske (levo ali desno).

Naslednji dostop do položaja 3 zahteva dva pritiska na desno.

Naslednji dostop do položaja 6 zahteva tri pritiske (levo ali desno).

Naslednji dostop do položaja 2 zahteva dva pritiska v desno.

Za dostop do položaja 5 so potrebni trije pritiski (levi ali desni).

Skupaj torej 16 pritiskov.

Druga možnost je 4, 1, 5, 2, 6, 3:

Prvi dostop do položaja 4 zahteva tri pritiske (levo ali desno).

Naslednji dostop do položaja 1 zahteva tri pritiske (levo ali desno).

Naslednji dostop do položaja 5 zahteva dva pritiska v levo.

Naslednji dostop do položaja 2 zahteva tri pritiske (levo ali desno).

Naslednji dostop do položaja 6 zahteva dva pritiska v levo.

Za dostop do položaja 3 so potrebni trije pritiski (levo ali desno).

Skupaj torej zopet 16 pritiskov.

Računalniško ozadje

Pogosto v računalništvu govorimo o iskanju najkrajše, najhitrejše oziroma najbolj optimalne rešitve, a, kot smo videli v tej nalogi, moramo za oceno učinkovitosti posameznega algoritma preveriti tudi, kako se izkaže v najslabšem primeru. Le če upoštevamo vse primere rešitev lahko ocenimo, kateri algoritem je učinkovitejši za rešitev posameznega problema.

Čudograd

državno, 6. do 9. razred

V Čudogradu živi en sam čarovnik. Ta čarovnik se lahko spremeni v vilo ali pa desno od sebe ustvari vilo. Vila se lahko spremeni v zaporedje napoja in zmaja ali pa v zaporedje napoja, čarovnika in zmaja.

Preobrazbe so prikazane v tabeli:

Prej	Potem
	
	
	
	

Te čarobne preobrazbe se lahko zgodijo poljubno mnogokrat in v poljubnem vrstnem redu. To pomeni, da se lahko vsak čarovnik in vsaka vila kadar koli spremenita.

Katerega izmed spodnjih stanj v Čudogradu **ni mogoče dobiti**, če začnemo z enim samim čarovnikom?

A) 

B) 

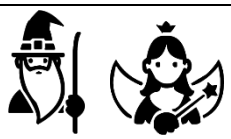
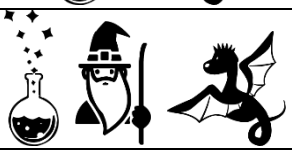
C) 

D) 

Rešitev

V Čudogradu ni mogoče dobiti stanja B.

Oštevilčimo čarobne transformacije od 1 do 4:

	Prej	Potem
1		
2		
3		
4		

Možnost A lahko dobimo tako, da začnemo z enim čarovnikom in uporabimo pretvorbe 1, 4, 2 in 3, v tem vrstnem redu.

Možnost C lahko dobimo tako, da začnemo z enim čarovnikom in uporabimo pretvorbe 2, 2, 3, 4 in 1, v tem vrstnem redu. (Seveda je to le eno izmed zaporedij pretvorb, s katerimi dobimo to stanje.)

Možnost D lahko dobimo tako, da začnemo z enim čarovnikom in uporabimo pretvorbe 2, 1, 3 in 3, v tem vrstnem redu.

Da možnost B ni mogoča, lahko hitro ugotovimo tako, da opazimo, da pravila preoblikovanja vedno hkrati ustvarijo napoj in zmaja. Zato bo število napojev v Čudogradu vedno enako številu zmajev, kar pa ne velja za možnost B.

Računalniško ozadje

V tej nalogi se posamezen simbol v skladu s pravili preslika v zaporedje simbolov. Tabela prikazuje produkcijska pravila, ki jih je potrebno upoštevati pri ustvarjanju veljavnih besed te gramatike.

Domine

državno, 6. do 9. razred

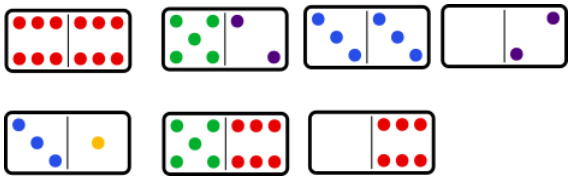
Vsaka domina je razdeljena na dva dela, na vsakem delu pa je lahko od 0 do 6 pik. Domine sestavljamo tako, da združujemo dele z istim številom pik. Pri tem lahko domino tudi obrnemo (da se zamenjata levi in desni del). Iz domin želimo sestaviti čim daljšo vrsto.

Primer: dve domini lahko sestavimo, kot kaže spodnja slika.

To vrsto dveh domin lahko podaljšamo na tri domine, če:

- na levo stran vrste dodamo novo domino, ki ima en del s šestimi pikami,
- na desno stran vrste dodamo novo domino, ki ima en del brez pik.

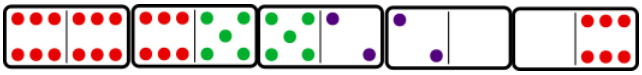
Na voljo imamo naslednjih sedem domin:



Koliko domin je v najdaljši vrsti, ki jo lahko sestavimo iz podanih domin?

Rešitev

Najdaljša vrsta iz podanih domin vključuje 5 domin. Ena od možnih rešitev je na sliki.



Dejansko lahko v rešitvi uporabimo le pet domin na zgornji sliki (vrsto iz njih lahko sestavimo tudi na drugačen način), preostalih dveh domin pa ne moremo dodati v to vrsto. Ena od preostalih domin ima na obeh delih 3 pike, zato je ne moremo nikjer kombinirati s temi petimi dominami v rešitvi (nimamo domine s tremi pikami). Podobno velja tudi za drugo domino, ki ima na enem deli tri pike, na drugem pa eno piko: tudi ena sama pika se ne pojavi v naši vrsti domin. Preostali dve domini lahko sestavimo le v novo vrsto dolžine 2. Torej je najdaljša vrsta iz podanih domin dolžine 5.

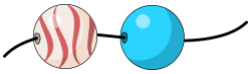
Računalniško ozadje

V računalništvu lahko eno domino predstavimo kot objekt, ki ima natančno določene lastnosti (število pik na vsakem delu) in tudi obnašanje (pravila za sestavo dveh domin). Podobno kot domine lahko kot objekte predstavimo tudi druge stvari, osebe ali pojme, s katerimi manipuliramo v programu. Tak način programiranja imenujemo objektno usmerjeno programiranje.

Mornarske ogrlice

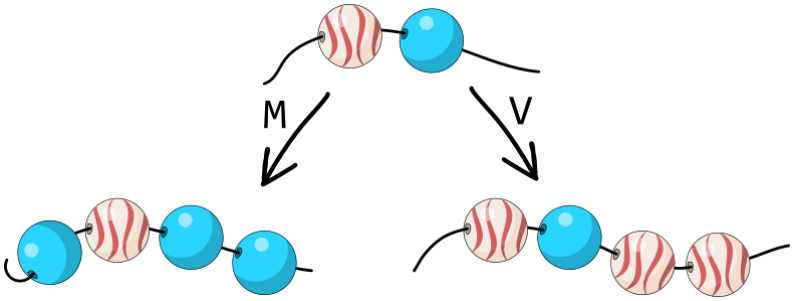
državno, 6. do 9. razred

Mornarsko ogrlico izdelamo iz valovitih in modrih korald. Začnemo z dvema koraldama, valovito in modro, ki ju nataknemo na vrvico:



Ogrlico podaljšamo tako, da

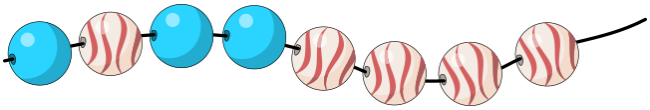
- dodamo modro koraldo na vsako stran ogrlice (akcija M)
- ali pa dodamo dve valoviti koraldi na desno stran ogrlice (akcija V).



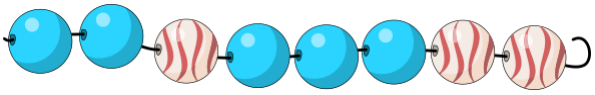
Ti dve akciji poljubno ponavljamo, dokler ne dobimo ogrlice želene dolžine.

Katera ogrlica **ni** mornarska ogrlica?

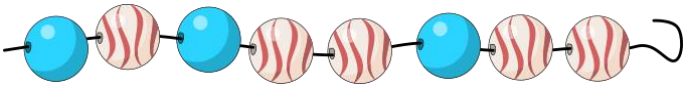
A)



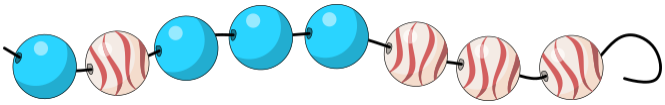
B)



C)

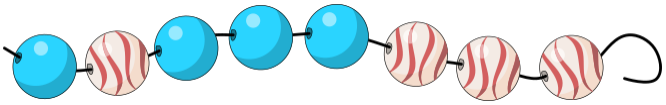


D)



Rešitev

Pravilen odgovor je D.


ni mornarska ogrlica.

Do rešitve lahko pridemo na več načinov. Za vsako ogrlico lahko najprej poiščemo obe začetni koraldi, nato pa s pomočjo akcij M in V poskušamo dobiti končno ogrlico. Ogrlico A lahko

naredimo tako, da začnemo z drugo in tretjo koraldo, nato pa izvedemo akcije M, V in V (dodamo na vsako stran eno modro koraldo in na desno stran dodamo dvakrat po dve valoviti koraldi). Ogrlico B sestavimo tako, da začnemo s tretjo in četrto koraldo, nato pa izvedemo akcije M, M in V (dvakrat dodamo na vsako stran po eno modro koraldo, nato pa na desni dodamo še dve valoviti koraldi). Ogrlico C začnemo z drugo in tretjo koraldo, ki jima dodamo koralde z akcijami V, M in V (najprej dodamo dve valoviti koraldi na desno stran, nato dodamo po eno modro koraldo na vsako stran in na koncu še dve valoviti na desno stran). Pri ogrlici D pa sta začetni koraldi lahko le druga in tretja. Nato lahko izvedemo akcijo M in dobimo prve štiri koralde. Pete, modre koralde, pa ne moremo dobiti z nobeno od navedenih akcij (niti M niti V). Torej ogrlica D ni narejena po mornarskih pravilih.

Ta pristop k reševanju problema pa se izkaže kot neučinkovit, če imamo zelo dolge ogrlice ali pa imajo ogrlice več možnih začetnih parov korald. V tem primeru bi bil bolj enostaven pristop, kjer bi končno ogrlico s pomočjo obratnih akcij M in V krajšali, dokler ne bi prišli do dveh začetnih korald. Obratna akcija M je odvzemanje po ene modre koralde na vsaki strani ogrlice, obratna akcija V pa je odvzemanje dveh valovitih korald na desni strani ogrlice. Če takih akcij ne moremo izvesti, ali če nas izvedene akcije ne pripeljejo do dveh začetnih korald, potem ogrlica ni narejena po mornarskih pravilih.

Tretji način, ki bi dobro deloval pri naših možnih odgovorih, pa je, da pogledamo število posameznih korald v ogrlici. Glede na podan začetek in možna pravila, mora biti število modrih korald vedno liho. Enako velja tudi za število valovitih korald. (Verjetno lahko hitro vidite, zakaj je tako?) To velja za ogrlice A, B in C. Ogrlica D pa ima sodo število modrih in sodo število valovitih korald, zato ne ustreza mornarskim pravilom. (Verjetno ni potrebno posebej poudariti, da je tretji pristop uporaben le pri podanih rešitvah v tej nalogi, ne pa v splošnem.)

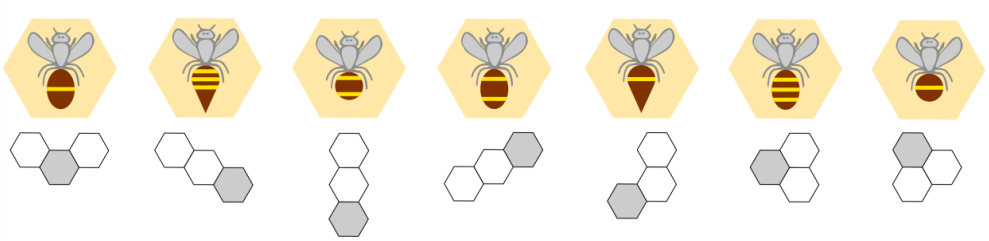
Računalniško ozadje

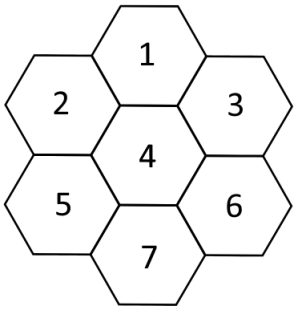
Pri ogrlici lahko koralde dodajamo (ali odvzemamo) le na obeh koncih ogrlice, ne moremo dodati (ali odvzeti) koralde nekje na sredini ogrlice. V računalništvu je ogrlici podobna podatkovna struktura, ki se imenuje *dvojna vrsta* (angl. *double-ended queue* ali *deque*). Dvojno vrsto lahko uporabimo za shranjevanje zgodovine brskalnika, za razvrščanje zahtev za tiskanje, ali pa za preverjanje veljavnosti matematičnih izrazov. Preverjanje ujemanja oklepajev lahko na primer naredimo zelo podobno, kot smo preverjali mornarske ogrlice.

Čebelnjak

državno, 6. do 9. razred, srednja šola

Vsaka čebela ima v panju posebno mesto, kot določajo naslednja pravila (čebela je v sivi celici):



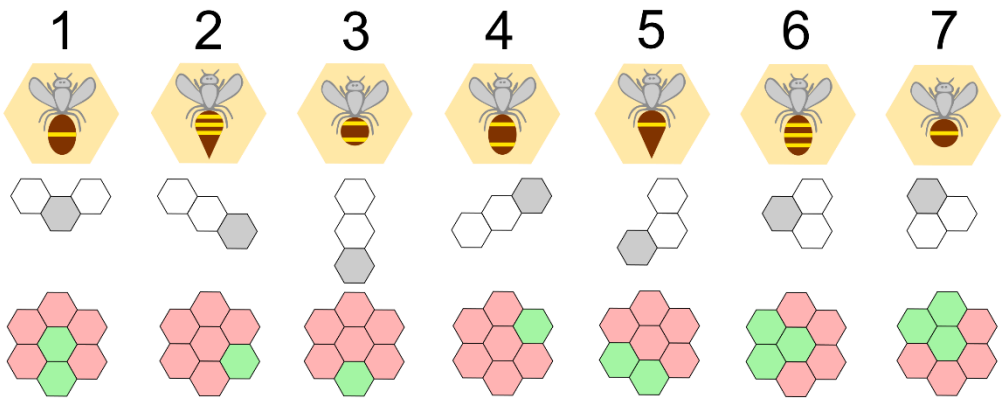


Čebele se razpostavijo tako, da je vsaka na mestu, skladno s pravilom zanjo. Na katerem mestu v panju je najbolj desna čebela,  ?

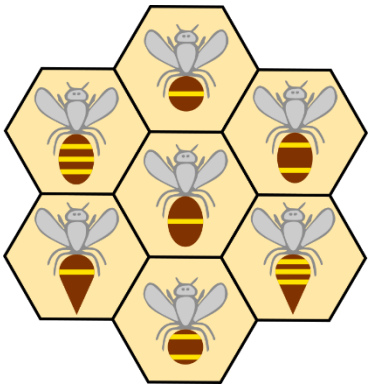
Rešitev

Čebela  je v panju na mestu 1.

Nalogo bi lahko reševali tako, da bi za vse možnosti preverili, ali ustrezajo pravilom. Vendar pa bi za to porabili zelo veliko časa (in truda). Boljši pristop je, da najprej podrobneje pogledamo podana pravila. Na spodnji sliki so za vsako čebelo glede na njeno pravilo prikazane celice v panju, kamor jo lahko postavimo (zelene celice):



Vidimo, da lahko nekatere čebele postavimo v eno samo celico v panju. Na primer, čebela 2 je lahko le v 6. celici panja, ker njeno pravilo treh zaporednih diagonalnih celic lahko le na en način postavimo v panj.



Nalogo rešimo tako, da najprej v panj postavimo tiste čebele, ki so lahko le na enem mestu v panju. To so čebele 2, 3 in 4. Čebela 1 je lahko v panju na dveh različnih mestih (4 in 7), a je mesto 7 že zasedeno s čebelo 3, zato je čebela 1 na mestu 4. Podobno velja tudi za čebelo 5, ki pride na mesto 5 v panju. Čebela 6 je v panju lahko na treh različnih mestih (2, 4 in 5), a sta mesti 4 in 5 že zasedeni, zato pride ta čebela na mesto 2. Podobno je tudi čebela 7 lahko na treh različnih mestih (1, 2 in 4), a ker sta mesti 2 in 4 že zasedeni, ji ostane le še mesto 1. Celotna rešitev je torej naslednja:

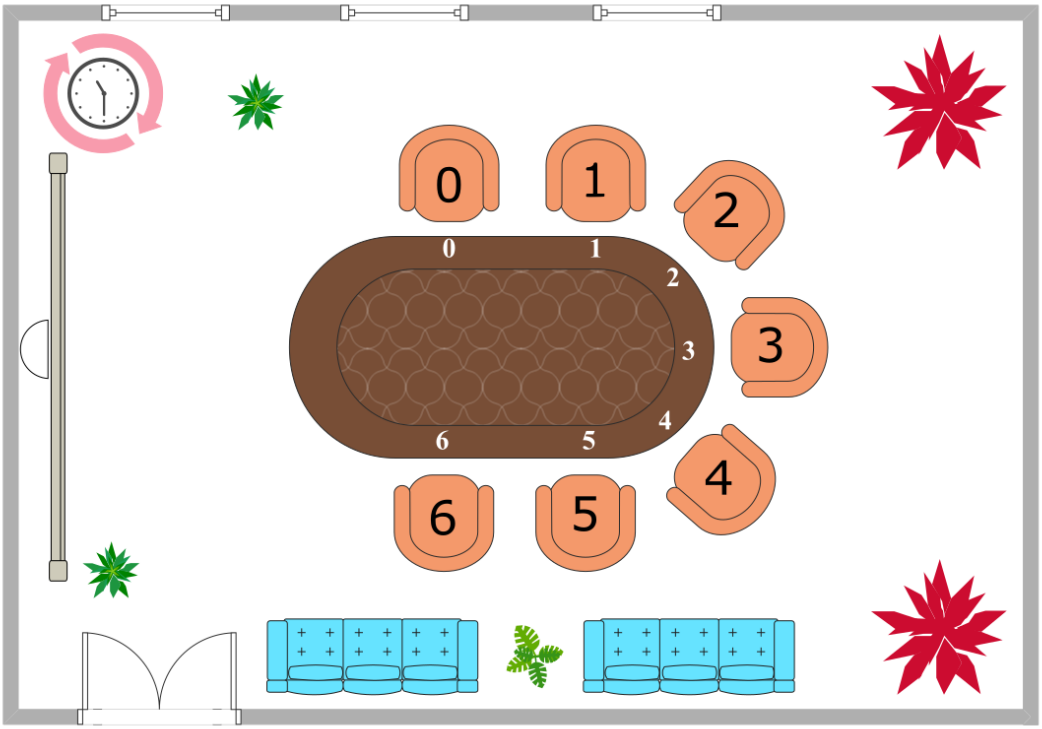
Računalniško ozadje

V nalogi smo morali sedem čebel razporediti na sedem mest v panju. Število vseh kombinacij razporeditve čebel je zelo veliko. Če upoštevamo podana pravila, kje v panju je lahko posamezna čebela, se število vseh možnosti zelo zmanjša. Vendar jih je še vedno veliko (preveč), da bi lahko enostavno rešili problem. Ključ do uspešnega pristopa k reševanju problema je v pravilnem zaporedju pravil: če najprej upoštevamo najbolj omejujoča pravila, ostane le malo različnih možnosti, ki jih moramo pregledati, da najdemo rešitev.

Filmski večer

državno, 6. do 9. razred, srednja šola

Gospa Naja je organizirala filmski večer za sedem učencev, ki obiskujejo filmski krožek. Za dogodek je rezervirala šolsko sejno sobo z ovalno mizo, sedaj pa mora določiti sedežni red.



Vse učence je najprej zapisala v tabelo:

Ime	Mark	Hana	Ema	Jon	Janez	Maja	Sara
Rojen	5. 1.	8. 2.	30. 1.	9. 9.	24. 12.	16. 4.	2. 12.

Nato jim je dodelila sedeže, ki so označeni od 0 do 6 (glej sliko). V izogib pritožbam je določila stroga pravila za dodelitev sedeža za mizo:

- Vsoto učenčevega rojstnega dneva in meseca deli s 7. Učenec dobi stol s številko, ki je enaka ostanku po deljenju. Tako bi na primer Janez, čigar vsota dneva in meseca njegovega rojstva je enaka 24 + 12 = 36, ostanek pri deljenju s 7 pa je 1, dobil stol številka 1.
- Če je stol s to številko že zaseden, se učencu dodeli številka prvega prostega stola, gledano v smeri urinega kazalca. Na primer, če učencu pripada stol 2 in je ta že zaseden, se mu dodeli stol 3. Če sta zasedena oba stola, 2 in 3, dobi učenec stol 4. V primeru zasedenosti stola 6, se učenec prestavi na stol 0.
- Učenci se posedajo po vrsti, kot so zapisani v tabeli (prvi se usede Mark).

Na katerem stolu bo sedela Sara?

Rešitev

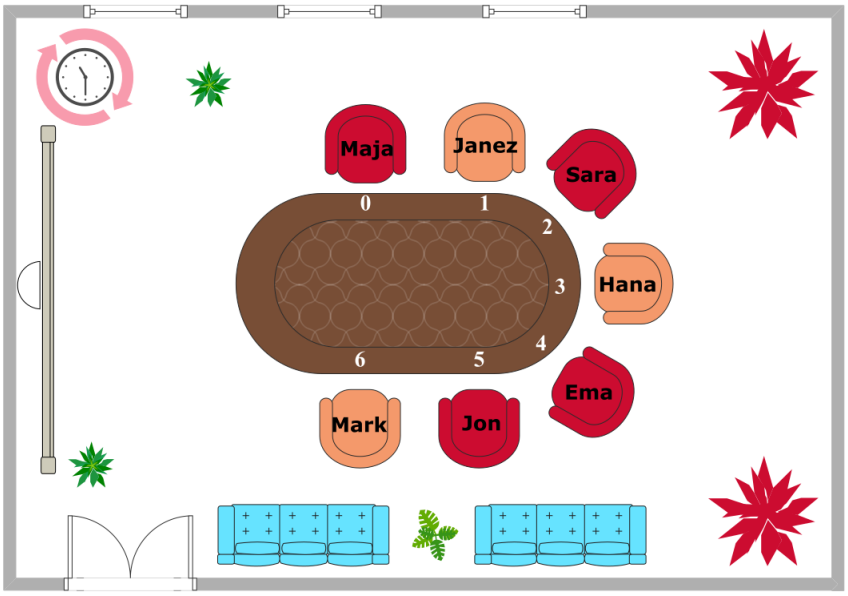
Sara bo sedela na stolu številka 2.

Do rešitve pridemo tako, da najprej za vsakega učenca izračunamo začetno številko stola po prvem pravilu (seštejemo dan in mesec rojstva, rezultat delimo s 7 in gledamo ostanek):

Ime	Mark	Hana	Ema	Jon	Janez	Maja	Sara
Rojen	5. 1.	8. 2.	30. 1.	9. 9.	24. 12.	16. 4.	2. 12.
vsota (dan + mesec)	5 + 1 = 6	8 + 2 = 10	30 + 1 = 31	9 + 9 = 18	24 + 12 = 36	16 + 4 = 20	2 + 12 = 14
ostanek pri deljenju s 7	6	3	3	4	1	6	0
dodeljen stol	6	3	4	5	1	0	2

V tabeli vidimo, da naj bi Hana in Ema obe sedeli na stolu 3. Ker lahko na enem stolu sedi le en učenec, dobi stol 3 Hana, ki ima prednost glede na starost. Ema tako dobi naslednji prosti stol, to je 4. Naslednji po starosti je Jon, ki bi moral dobiti stol 4, a je ta že zaseden, zato dobi stol 5. Tudi Majin na začetku dodeljeni stol (številka 6) je že zaseden, saj na njem že sedi Mark. Zato Maja dobi naslednji prosti stol (v smeri urinega kazalca), to je stol 0. Tako je sedaj zaseden tudi Sarin začetni stol, zato Sara zasede edini še prosti stol, to je 2.

Spodnja slika prikazuje končni sedežni red. Učenci na oranžnih stolih sedijo na začetno dodeljenih stolih po prvem pravilu. Z rdečo pa so označeni stoli, kjer se je moral učenec premakniti vsaj za en stol, ker je bil njegov na začetku dodeljeni stol že zaseden (drugo pravilo).



Računalniško ozadje

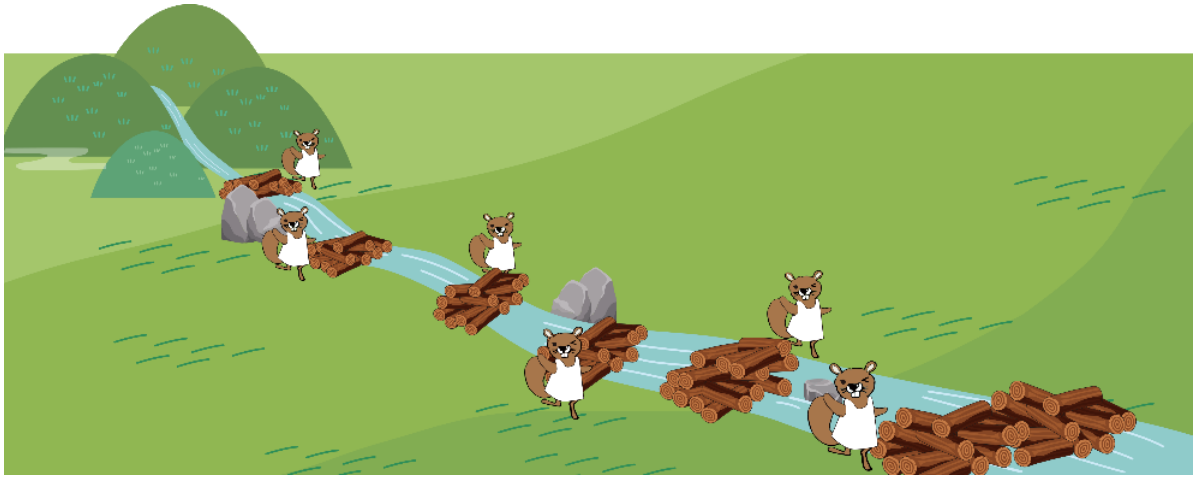
Podoben način, kot ga je uporabila gospa Naja v naši nalogi, se uporablja tudi v računalništvu za hitrejši dostop do shranjenih podatkov.

Podatke shranimo v *zgoščevalni tabeli* (angl. *hash table*) tako, da vsak podatek s pomočjo *zgoščevalne funkcije* (angl. *hash function*) preslikamo v vrednost na določenem intervalu in to vrednost uporabimo kot indeks za hitrejši dostop do shranjenega podatka. Če zgoščevalna funkcija dva različna podatka preslika v isto vrednost, imamo še posebno pravilo za razrešitev tega spora: če ima izračunano vrednost že nek drug podatek, moramo izračunano vrednost na nek način spremeniti.

V naši nalogi je zgoščevalna funkcija preslikala ime učenca v število od 0 do 6 tako, da je izračunala ostanek, ki ga dobimo, ko vsoto dneva in meseca rojstva učenca delimo s 7. Če je izračunana vrednost bila že dodeljena nekemu učencu, smo za razrešitev spora izračunano vrednost povečali za ena (po modulu 7).

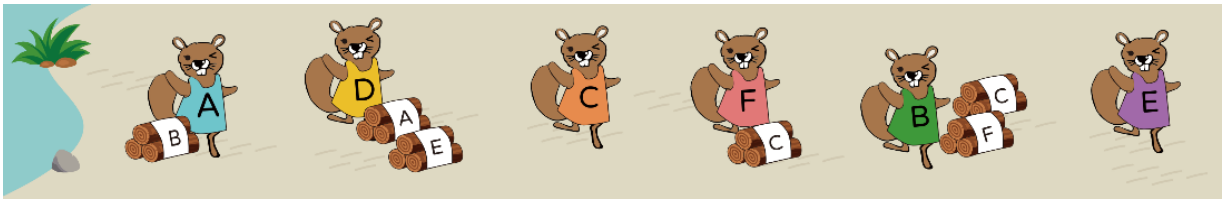
Jezovi

državno, 6. do 9. razred, srednja šola



Šest bobrov (A, B, C, D, E, F) je zgradilo vsak svoj jez na potoku. Po močnem neurju je potok zelo narasel in podrl vse jezove, hlode iz jezov pa je odneslo nizvodno. Na srečo so skrbni bobri vsak hlod označili z imenom graditelja jezu; na primer, bober A je vse hlode v svojem jezu označil z A.

Po neurju so se bobri zbrali ob potoku. Vsak je prinesel najdene hlode, da jih vrne lastnikom, in želel poiskati hlode iz svojega jezu. Spodnja slika prikazuje bobre in hlode, ki so jih našli.



Glede na to, katere hlode je pobral vsak od bobrov, kateri od naštetih bi lahko bil vrstni red jezov gledano nizvodno (od izvira navzdol)?

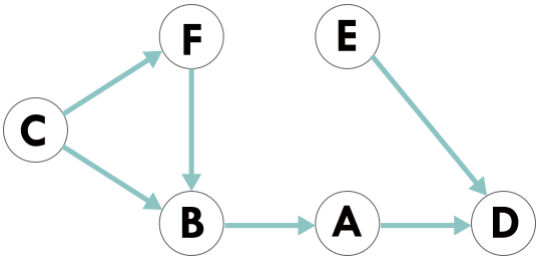
A) A → B → C → D → E → F
B) C → B → F → A → D → E
C) C → F → B → D → A → E
D) E → C → F → B → A → D

Rešitev

Pravilen odgovor je D: E → C → F → B → A → D.

Ker je hlode odneslo nizvodno, lahko sklepamo, da če je bober X našel hlode od bobra Y, mora biti jez bobra Y nad (tj. v bolj zgornjem toku) jezom bobra X.

Pri določanju pozicije vsakega bobra si lahko pomagamo z grafom, v katerem dva bobra povežemo s puščico, če je jez prvega bobra nad jezom drugega bobra (puščica kaže v smeri drugega bobra).



Podatke o relativnih pozicijah bobrov najdemo na sliki naloge: na primer, bober A je našel hlode iz jezu bobra B, kar pomeni, da je jez bobra B nad jezom bobra A, kar v grafu označimo s puščico od vozlišča B do vozlišča A.

Kot lahko vidimo v grafu, nimamo nobenih puščic do vozlišča C in vozlišča E. Torej je prvi jez na potoku lahko le jez bobra C ali bobra E. Torej odgovor A ni pravilen.

Tudi odgovor B ni pravilen, saj bi moral biti jez bobra F nad jezom bobra B (puščica od F proti B v grafu).

Odgovor C je tudi nepravilen, ker mora biti jez bobra A nad jezom bobra D (puščica od A proti D v grafu).

Le odgovor D ustreza našemu grafu: E je nad D, C je nad B in F, F je nad B, B je nad A in A je nad D.

Računalniško ozadje

Usmerjen aciklični graf sestavljajo vozlišča, ki so povezana z usmerjenimi povezavami (puščicami) brez ciklov. V naši nalogi smo bobre prikazali kot vozlišča, usmerjene povezave pa prikazujejo, v kateri smeri je odneslo hlode iz jezov. Postavljanje bobrov v pravi vrstni red od zgornjega do spodnjega toka, imenujemo topološko urejanje, to je ureditev vozlišč po vrsti glede na smer povezav.

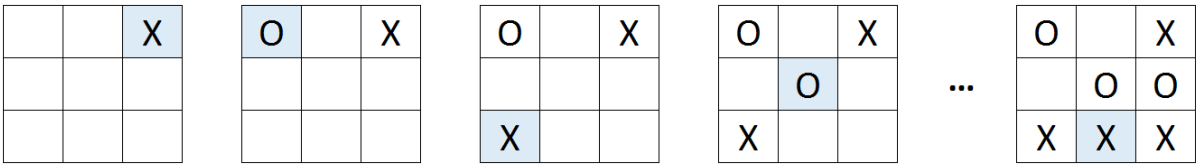
V računalništvi se topološko urejanje med drugim uporablja za razvrščanje nalog, ki so medsebojno odvisne.

Križci in krožci

državno, 6. do 9. razred, srednja šola

Če slučajno ne poznate igre križci in krožci, si najprej poglejmo njena preprosta pravila. Igrata dva igralca, ki na mreži velikosti 3 x 3 v polja izmenično rišeta križce X (en igralec) oziroma krožce O (drugi igralec). Zmaga tisti, ki prvi postavi tri zaporedne znake (X ali O) vodoravno, navpično ali po diagonali. Če so zapolnjena že vsa polja in ni zmagovalca, se igra konča neodločeno.

Primer igre prikazujejo spodnje slike: prva poteza prvega igralca je na skrajno levi sliki (kvadratek z zadnjo potezo na vsaki sliki je pobarvan), sledi prva poteza drugega igralca na drugi sliki in tako naprej do zadnje (desne) slike, kjer je prikazana zadnja, zmagovalna poteza.



Sliko na desni bomo poimenovali rezultat zaključene igre. Seveda ni vsaka mreža z vpisanimi X in O veljaven rezultat zaključene igre glede na zgoraj podana pravila igre.

Katera slika prikazuje veljaven rezultat zaključene igre?

A)

X	O	X
O	X	O
O	O	X

B)

X	O	X
O	X	
O	X	X

C)

X	X	O
	O	X
O	O	X

D)

X	O	X
O	X	O
O	X	

Rešitev

Pravilen odgovor je C.

Na sliki C je igralec O postavil tri v vrsto, nato pa se je igra končala.

Odgovor A ni pravilen, saj je zmagal igralec X (imamo tri X v vrsti), a je na poljih več znakov O kot znakov X, kar ni možno glede na pravila. Ker zmagovalec vedno zapiše zadnji znak, ima ta igralec lahko enako ali več zapisanih znakov kot nasprotni igralec, nikakor pa jih ne more imeti manj.

Odgovor B tudi ni pravilen, saj je na poljih pet znakov X in le trije O. Ker igralca izmenično zapisujeta znake, je lahko razlika med številom znakov enega in drugega igralca ob zaključku igre le 0 ali 1.

Odgovor D tudi ni pravilen, saj zmagovalca ni, igra pa se je končala, preden so bila zapolnjena vsa polja.

Računalniško ozadje

Nalogo smo rešili tako, da smo za vse štiri odgovore preverili, če so možni glede na pravila igre.

Iz podanih pravil igre smo lahko sestavili pravila, kako mora izgledati veljaven rezultat zaključene igre. Na primer:

1) Razlika med številom O-jev in X-ov mora biti 0, 1 ali -1.
2) Če noben igralec ni zmagal, morajo biti zapolnjena vsa polja.
3) Zmagovalec mora imeti ali enako število znakov ali en znak več kot igralec, ki je izgubil.
4) Zaključena igra ima lahko le eno zmagovalno zaporedje treh znakov.

Če slika ne ustreza vsem štirim pravilom, ne more biti veljaven rezultat zaključene igre.

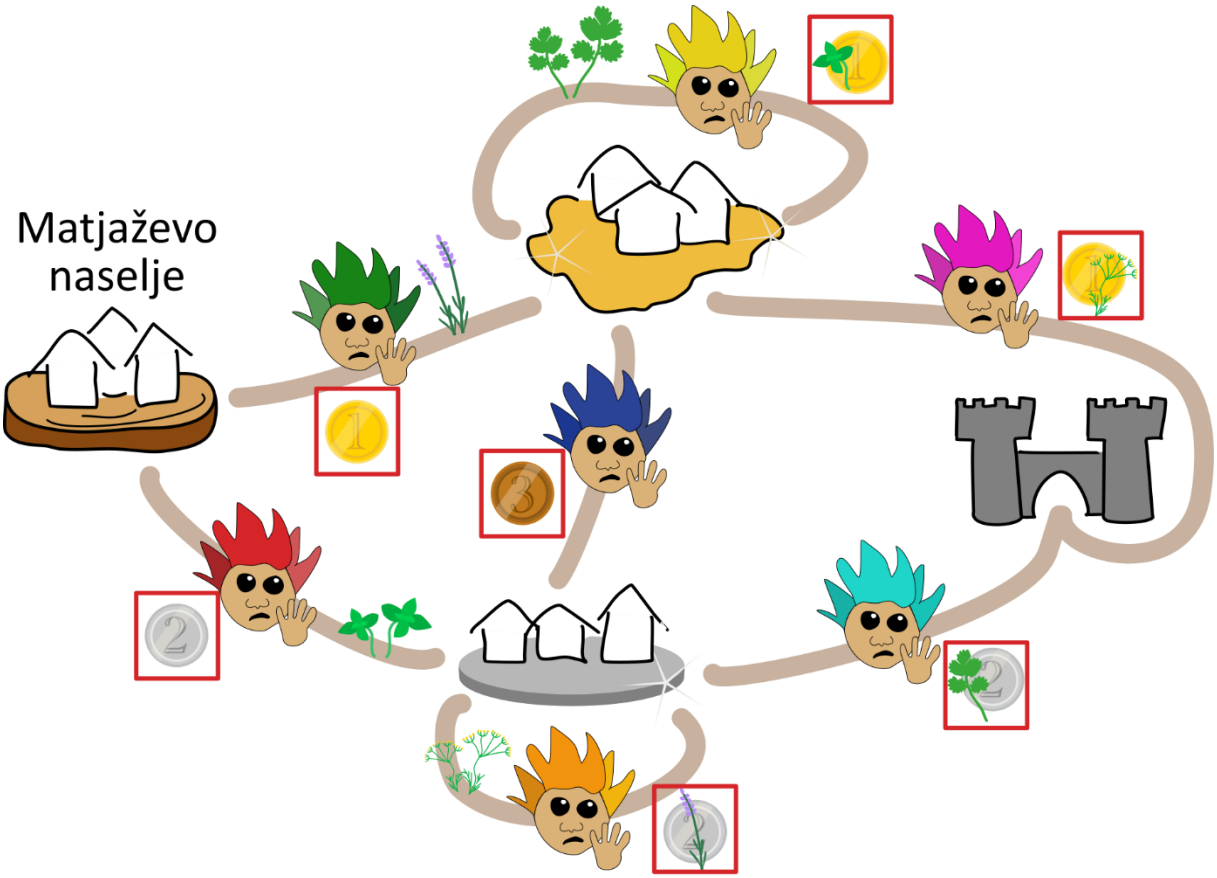
V računalništvu so pravila zelo pomembna, še posebej pri procesiranju podatkov. Tako imamo pravila, ki določajo format datoteke s sliko, številke kreditnih kartic, telefonske številke in podobno. Ko želimo s programom za prikaz slik odpreti datoteko, program najprej preveri, če je vsebina datoteke veljavna glede na pravila formata slike. Le če vsebina ustreza pravilom, program prebere podatke in prikaže sliko.

Lakomni škrati

državno, 6. do 9. razred, srednja šola

Matjaž mora priti iz svojega domačega naselja do gradu, a poti ga vodijo skozi gozd, v katerem prebivajo lakomni škrati. Ti čakajo popotnike na poteh v gozdu in od njih zahtevajo en kovanec (zlatnik ali srebrnik ali bakreni kovanec), da jih spustijo naprej. Nekateri škrati poleg kovanca zahtevajo še dodatno darilo, in sicer baziliko, koriander, koper ali sivko.

Vsa ta zelišča rastejo ob poteh do posameznih naselij, zato jih lahko Matjaž nabere v gozdu. Matjaž ima tudi zemljevid, na katerem so označene poti, kje rastejo zelišča in katere kovance in morebitna zelišča zahteva posamezen škrat na poti (glej spodnjo sliko — zahteve škratov so v kvadratkih). Škrati zelo ljubosumno čuvajo svoje poti, zato ni mogoče, da ob poti naberemo zelišča, ne da bi predhodno plačali škratu za prehod poti.



Matjaž že doma pripravi vse kovance za plačilo škratom. Shrani jih v tulec, iz katerega lahko vedno vzame le vrhnji kovanec. Zato morajo biti kovanci v tulcu v takem zaporedju, kot jih bo potreboval na poti.

Za nabrana zelišča vzame Matjaž na pot majhen prazen nahrbtnik. Nahrbtnik je tako ozek, da lahko iz njega vsakič vzame le zelišče na vrhu (tisto, ki ga je nazadnje pospravil v nahrbtnik).

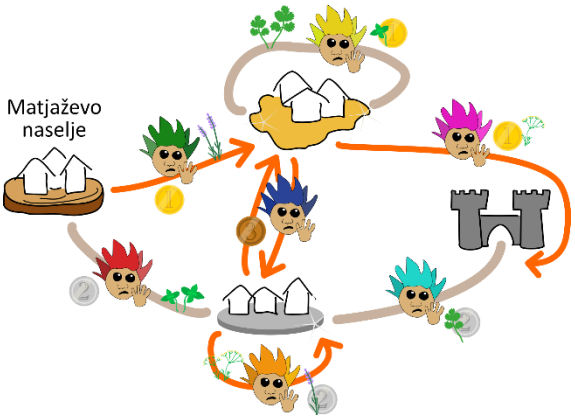
Kako naj Matjaž napolni tulec s petimi kovanci, da bo lahko plačal lakomnim škratom na svoji poti do gradu? Pri tem ne sme pozabiti, da mora na poti nabrati tudi zelišča!

Rešitev

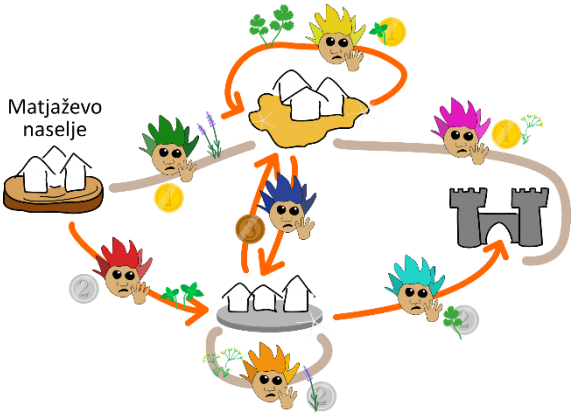
Matjaž lahko tulec napolni na dva načina. Prvi je, da v tulec shrani srebrnik, bakreni kovanec, zlatnik, bakreni kovanec in srebrnik. Druga možnost pa je, da vanj shrani zlatnik, bakreni kovanec, srebrnik, bakreni kovanec in zlatnik.

Obe rešitvi sta prikazani spodaj:

Matjaž gre najprej do zlatega naselja (zgoraj v sredini). Na poti plača en zlatnik in nabere sivko. Nato gre do srebrnega naselja (v sredini spodaj) in na poti plača en bakren kovanec. Nato se sprehodi okoli srebrnega naselja, za kar plača škratu en srebrnik in sivko, a ob poti nabere koper. Nato se vrne v zlato naselje, kar ga ponovno stane en bakren kovanec. Od tam pa gre lahko na grad, saj ima zlatnik in koper, ki ju zahteva škrat na tej poti do gradu.



Matjaž gre najprej do srebrnega naselja in na poti plača en srebrnik ter spotoma nabere baziliko. Nato gre do zlatega naselja in za pot plača en bakren kovanec. Za pot skozi gozd pri zlatem naselju mora škratu plačati zlatnik in podariti baziliko, a lahko ob poti nabere koriander. S plačilom enega bakrenega kovanca se vrne do srebrnega naselja, od tam pa nadaljuje do gradu, saj ima vse, kar zahteva škrat na tej poti do gradu: srebrnik in koriander.



To sta edini možni rešitvi, saj na poti do gradu potrebuje koper ali koriander. Če želi nabrati koper, mora najprej imeti sivko, ki jo lahko nabere le na poti do zlatega naselja. Ker ima v tulcu le pet kovancev (kar pomeni, da se lahko sprehodi po največ petih poteh), je edina možna rešitev ta, ki je prikazana prva.

Če pa želi nabrati koriander, mora najprej imeti baziliko. To lahko nabere po poti do srebrnega naselja. Tudi v tem primeru je ob omejitvi petih kovancev edina možna rešitev zapisana kot druga.

Računalniško ozadje

Naselja, grad in poti med njimi bi lahko prikazali kot graf. Vsaka pot ima določeno ceno (kovanec, ki ga je potrebno plačati škratu) in lahko omogoča, da nekaj dodamo v nahrbtnik ali vzamemo iz njega; nahrbtnik torej deluje kot sklad.

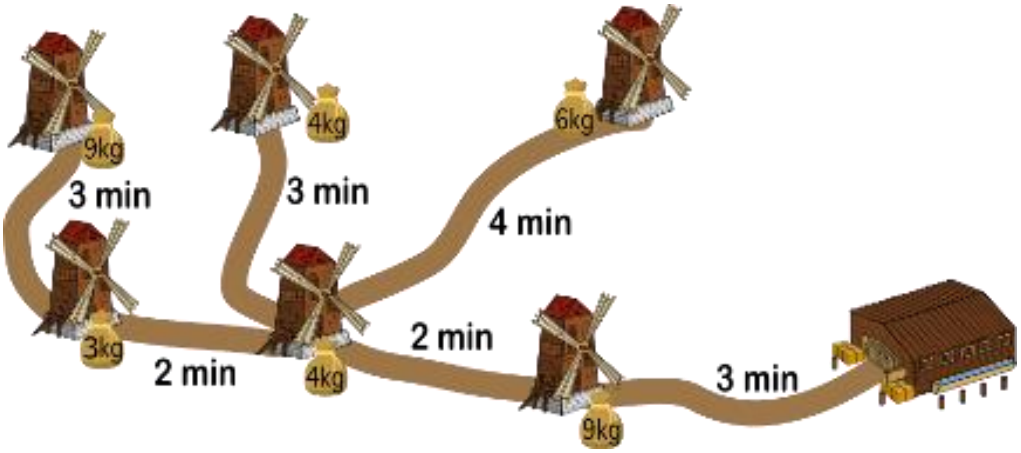
V računalništvu je sklad podatkovna struktura, ki deluje po principu »prvi noter, zadnji ven« (tj. kot kup stvari, ki jih nalagamo eno na drugo in s kupa lahko vedno vzamemo le stvar, ki je na vrhu, sicer bi se kup podrl). Podpira dve operaciji, in sicer dodajanje objekta na sklad (angl. *push*) in odstranjevanje vrhnjega objekta s sklada (angl. *pop*).

Na ta graf iz naloge lahko gledamo tudi kot na *skladovni avtomat* (angl. *push-down automaton*). Skladovni avtomat ima različna stanja (v nalogi so to naselja in grad), vhodno abecedo (v nalogi so to trije različni kovanci), skladovno abecedo (v nalogi so to štiri različna zelišča), množico prehodov med stanji (v nalogi so to poti), začetno stanje (v nalogi je to Matjaževo domače naselje) in množico končnih stanj (v nalogi je to grad). Posebnost skladovnega avtomata v naši nalogi pa je, da vse poti vodijo v obe smeri, kar v splošnem pri skladovnem avtomatu ne velja vedno.

Mlini

državno, 6. do 9. razred, srednja šola

Ceste povezujejo šest mlinov s skladiščem, kot kaže slika. Mlinarji ob koncu dneva postavijo žaklje z moko pred vrata svojih mlinov, da jih pobere Vili in prenese v skladišče še pred nočjo. Vili je močan in lahko naenkrat nese tudi več vreč z moko, a vseeno ne zmore bremen, ki so težja od 15 kg. Na sliki so zapisani tudi časi, ki jih Vili potrebuje za pot na vsakem odseku ceste.



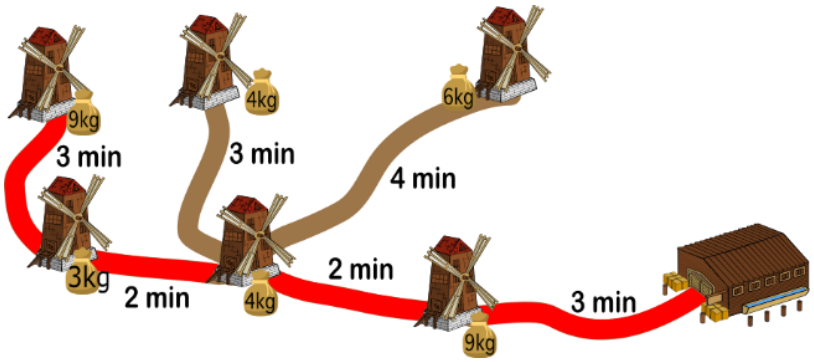
Vili starta iz skladišča in bi rad delo čimprej opravil. Najmanj koliko časa potrebuje, da prenese vse žaklje v skladišče?

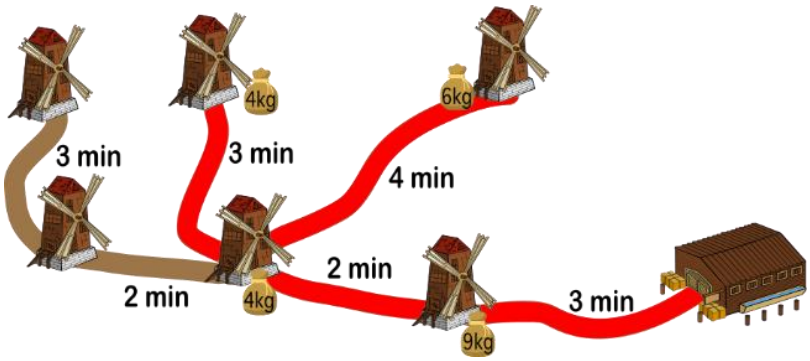
Rešitev

Vili potrebuje najmanj 50 minut.

Vili mora skupaj prenesti 35 kg moke. Ker lahko naenkrat nese le 15 kg moke, bo moral na pot vsaj trikrat, da bo v skladišče prenesel vse žaklje. Če želi skrajšati skupni čas poti, bo moral na eni poti pobrati žaklje iz sosednjih mlinov.

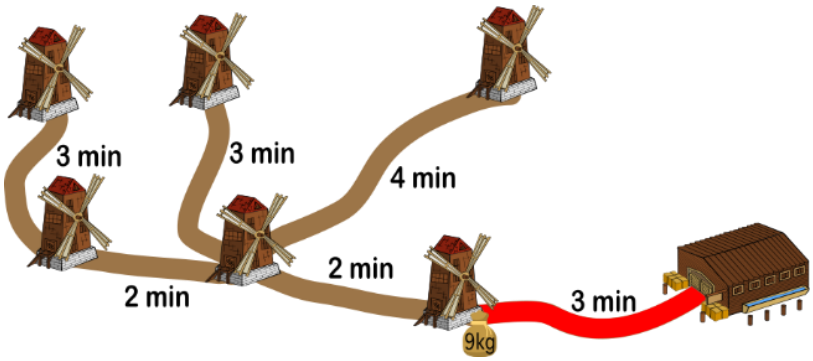
Na primer, ko pobere 9 kg žakelj iz mlina levo zgoraj, lahko spotoma pobere še 3 kg žakelj v sosednjem mlinu. Tako bo skupaj nesel 12 kg. Ker vsi ostali žaklji tehtajo več kot 3 kg, na tej poti ne bo mogel pobrati nobenega žaklja več. Za to pot bo porabil 20 minut (3 + 2 + 2 + 3 = 10 min v eno smer in ravno toliko tudi nazaj).





Da pobere še žaklje v preostalih štirih mlinih, bo moral na naslednji poti prenesti tri najbolj oddaljene žaklje: 6 kg iz mlina zgoraj desno in dvakrat po 4 kg iz srednjih dveh mlinov. Tako bo skupaj nesel 14 kg. Za to drugo pot bo porabil 24 minut (3 + 2 + 4 + 4 + 3 + 3 + 2 + 3).

Za tretjo pot pa mu ostane le še 9 kg žakelj v mlinu, ki je najbližje skladišču. Za to pot bo porabil še 6 minut (3 + 3)



Če seštejemo čase vseh treh poti (20 + 24 + 6), dobimo skupen čas, ki ga Vili potrebuje za delo, to je 50 minut.

Opazimo lahko, da vrstni red poti, ki jih naredi Vili, ni pomemben (npr. lahko bi najprej opravil pot do mlina, ki je najbližji skladišču), pomembno je le združevanje posameznih žakljev z moko, ki jih prenese na posamezni poti.

Računalniško ozadje

Tudi ta problem je optimizacijski problem, kjer želimo med vsemi možnimi rešitvami poiskati najboljšo rešitev. Navadno je rešitev povezana z iskanjem največje ali najmanjše vrednosti funkcije. V naši nalogi, na primer, iščemo najmanjši čas.

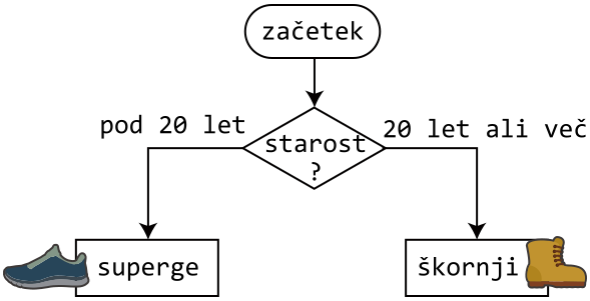
Tudi reševanje problemov z računalniki je povezano z optimizacijo, saj navadno želimo, da računalnik opravi naloge v čim krajšem času in z uporabo čim manj virov (kot je na primer spomin).

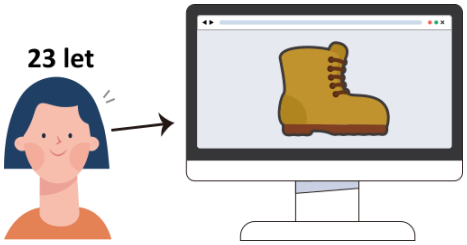
Rešitev te naloge bi lahko poiskali tako, da bi preverili vse možnosti za prenos žakljev v skladišče in med njimi poiskali tisto, ki zahteva najmanj časa. Tak pristop imenujemo *tehnika grobe sile* (angl. *brute force technique*). Primerna je za reševanje manjših problemov, kjer ni prav veliko možnih rešitev (in zato lahko preverimo vse). Pri večjih problemih, kjer je tudi veliko možnih rešitev, pa je tak postopek zelo dolgotrajen, zato uporabimo drug pristop, ki problem reši bolj učinkovito, na primer požrešni algoritem ali pa dinamično programiranje.

Modna oblačila

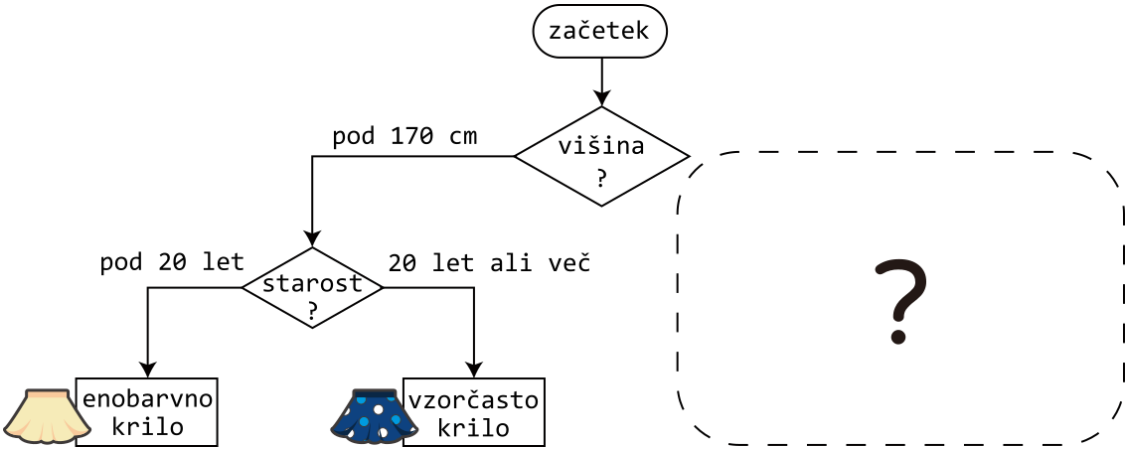
državno, 6. do 9. razred, srednja šola

Spletna trgovina z modnimi oblačili uporablja priporočilni sistem, ki temelji na informacijah o strankah. Če na primer stranka želi kupiti čevlje in vpiše v sistem svoje podatke, sistem sledi spodnjim pravilom in stranki priporoča škornje.





Nekega dne pa je sistemski administrator po nerodnosti izbrisal del pravil priporočilnega sistema. Pravila, ki so še ostala, so prikazana na sliki.

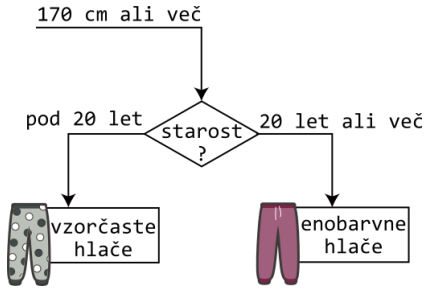


Na srečo je administrator našel zapise o predhodnih priporočilih sistema:

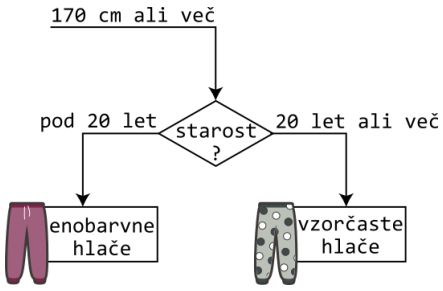
stranka	18 let 165 cm	32 let 168 cm	28 let 185 cm	18 let 172 cm	12 let 170 cm
priporočilo	enobarvno krilo	vzorčasto krilo	vzorčaste hlače	enobarvne hlače	enobarvne hlače

Na podlagi predhodnih priporočil je nato sestavil manjkajoča pravila. Kateri manjkajoči del pravil je pravi?

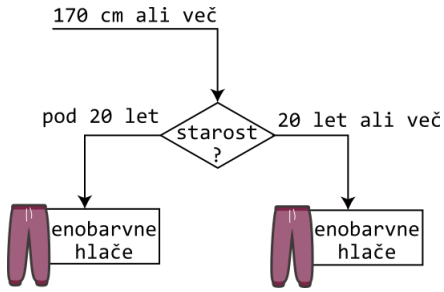
A)



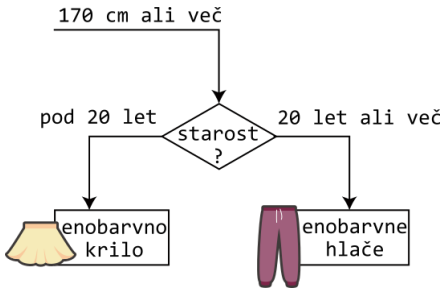
B)



C)



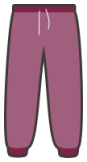
D)



Rešitev

Pravilen odgovor je B.

Prvi pogoj preveri strankino višino, torej manjkajo pravila le za stranke, ki so večje od 170 cm (vključno z višino 170 cm). V tabeli predhodnih priporočil lahko vidimo, da je sistem vsem takim strankam priporočal hlače:

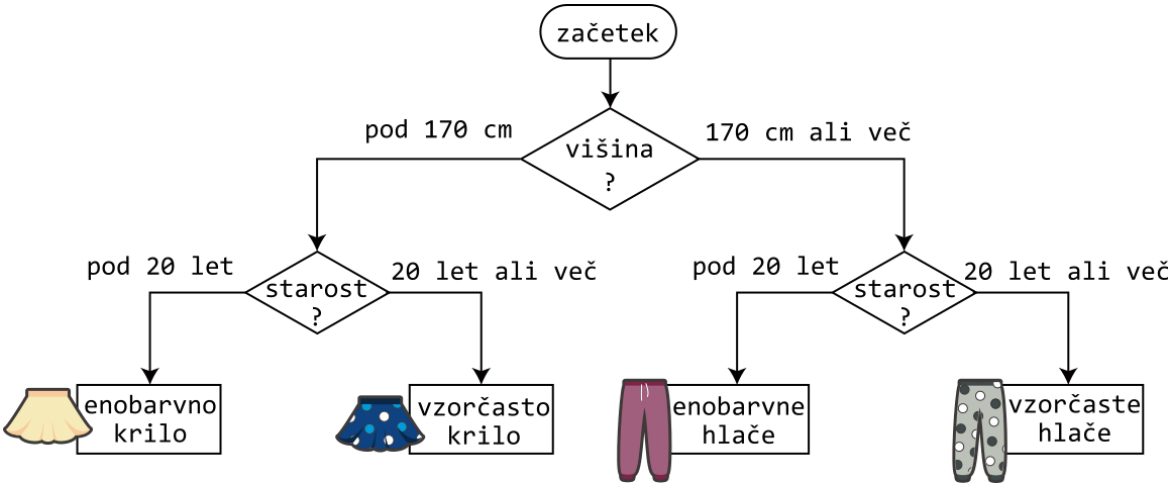
stranka	28 let 185 cm	18 let 172 cm	12 let 170 cm
priporočilo	vzorčaste hlače 	enobarvne hlače 	enobarvne hlače 

Torej odgovor D ni pravilen.

Ker je sistem vsem strankam, mlajšim od 20 let, priporočal enobarvne hlače, tudi odgovor A ni pravilen.

Ker je sistem 28-letni stranki priporočal vzorčaste hlače, tudi odgovor C ne more biti pravilen.

Sistem priporoča različne hlače glede na starost stranke, torej odgovor B ustreza predhodnim priporočilom. Pravila priporočilnega sistema so tako naslednja:



Računalniško ozadje

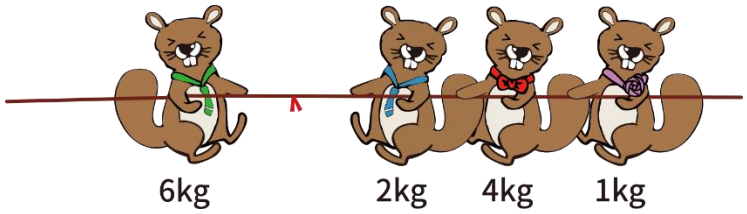
Diagram, s kateri smo ponazorili pravila priporočilnega sistema, imenujemo diagram poteka. V računalništvu ga uporabljamo za prikaz algoritma, delovnega toka ali procesa. Diagram poteka sestavljajo pravokotniki, ki predstavljajo akcije, in rombi, ki predstavljajo odločitve, medsebojno pa so povezani s puščicami, ki določajo zaporedje izvajanja. Začetek je je označen z zaobljenim pravokotnikom.

Vlečenje vrvi

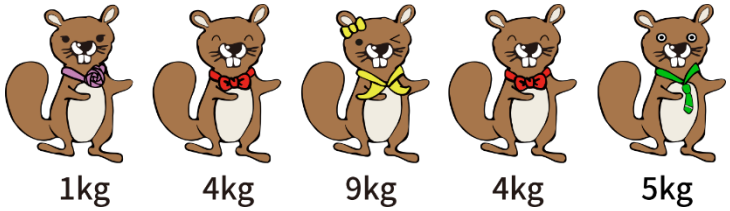
državno, 6. do 9. razred, srednja šola

Bobri tekmujejo v vlečenju vrvi. Za bolj pošteno tekmovanje so ekipe sestavili na podlagi teže posameznih tekmovalcev. Ekipe sestavijo tako, da je razlika v teži dveh ekip čim manjša.

Na primer, razlika v teži obeh ekip na spodnji sliki je 1 kg (bober v levi ekipi ima 6 kg, desno ekipo pa sestavljajo trije bobri: 2 kg, 4 kg in 1 kg, torej skupaj 7 kg).



Na tekmovanje se je prijavilo pet bobrov:



Koliko znaša najmanjša razlika v teži dveh ekip, ki ju lahko sestavimo iz prijavljenih bobrov?

Rešitev

Najmanjša razlika v teži dveh ekip je 3 kg.

Skupna teža vseh prijavljenih bobrov je 23 kg (1 + 4 + 9 + 4 + 5 = 23). Idealno bi bilo, če bi obe ekipi tehtali enako, to je 11,5 kg. Ker pa bobra ne moremo razrezati (in v ekipo vključiti samo en del bobra), mora biti skupna teža vsake ekipe celo število.

S to predpostavko bi bili najbolj po teži izenačeni ekipi z 11 kg in 12 kg. Vendar pa ne moremo najti kombinacije katerih koli od petih bobrov, ki bi skupaj tehtali 11 kg (oziroma 12 kg).

Torej sta skupni teži ekip lahko le 10 kg (11 – 1 = 10) in 13 kg (12 + 1 = 13), razlika med njima pa je 3 kg. Taki ekipi lahko sestavimo na več načinov: ekipo 10 kg lahko sestavljata dva bobra (1 kg in 9 kg) ali pa trije bobri (1 kg, 4 kg in 5 kg).

Računalniško ozadje

Problem, ki smo ga reševali v tej nalogi, je v računalništvu poznan kot problem nahrbtnika, ki je kombinatorični optimizacijski problem. Imamo več predmetov in za vsakega podano težo in vrednost. Nahrbtnik, ki ima omejeno največjo težo vsebine, želimo napolniti s takimi predmeti, da bo skupna vrednost predmetov v nahrbtniku čim večja. V poenostavljeni različici problema so vsi predmeti enako vredni in nas za vsak predmet zanima le to, ali ga postavimo v nahrbtnik

ali ne. Optimalno rešitev takega problema lahko dobimo z uporabo metod dinamičnega programiranja.

Ekipe v naši nalogi lahko obravnavamo kot nahrbtnik, kjer moramo za vsakega bobra ugotoviti, ali ga vključimo v ekipo ali ne, pri tem pa je skupna teža ekipe omejena na polovico skupne teže vseh bobrov.

Podobne probleme srečamo tudi v vsakdanjem življenju, na primer, kako v enem večeru preizkusiti čim več atrakcij v zabaviščnem parku ali pa kako prepotovati čim večjo razdaljo z omejenim proračunom.

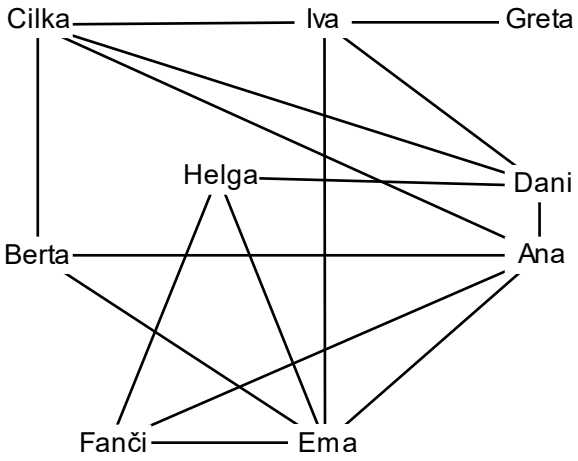
Vzdevek

državno, 6. do 9. razred, srednja šola

V ponedeljek je prišla nova učiteljica matematike. Ana, Dani in Iva so ji takoj nadele vzdevek, ki pa ga tu ne bomo napisali, ker ni lepo, da tako govorimo o učiteljicah. Sploh o učiteljicah matematike.

Če neka učenka sliši, da vzdevek uporablja vsaj pol njenih prijateljic, ga začne naslednji dan uporabljati tudi sama. Kdo je prijatelj s kom, pa kaže slika.

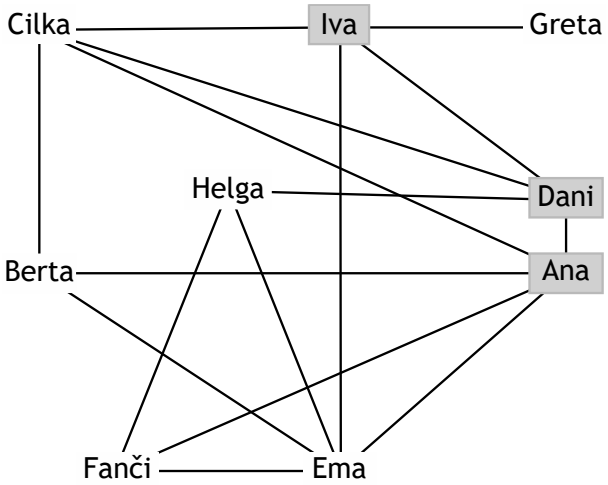
Na kateri dan so vzdevek uporabljale vse učenke na sliki?



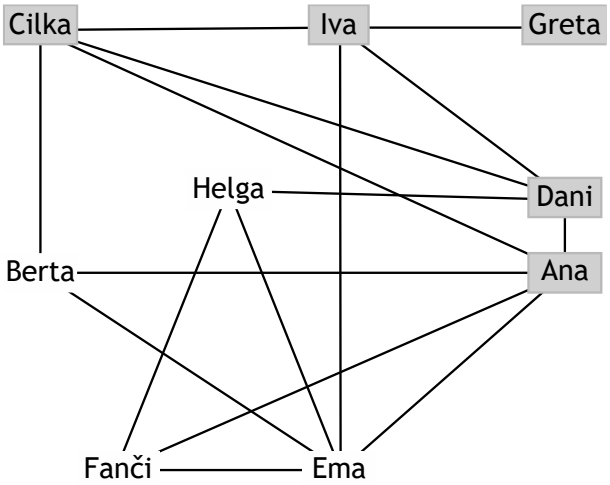
Rešitev

V petek. Vzdevek se širi takole.

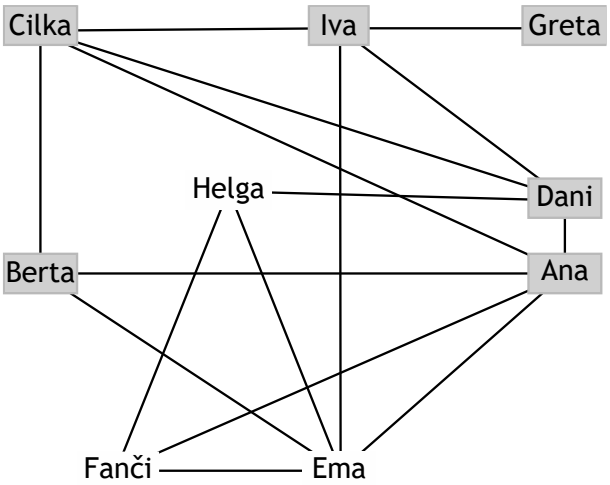
Ponedeljek



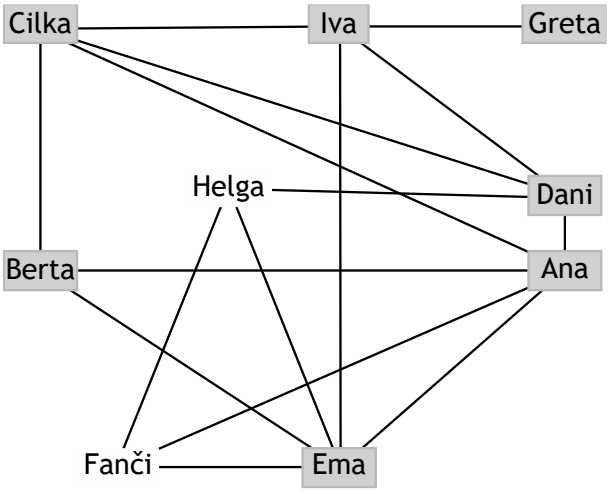
Torek


Cilka
Iva
Greta
Helga
Dani
Berta
Ana
Fanči
Ema

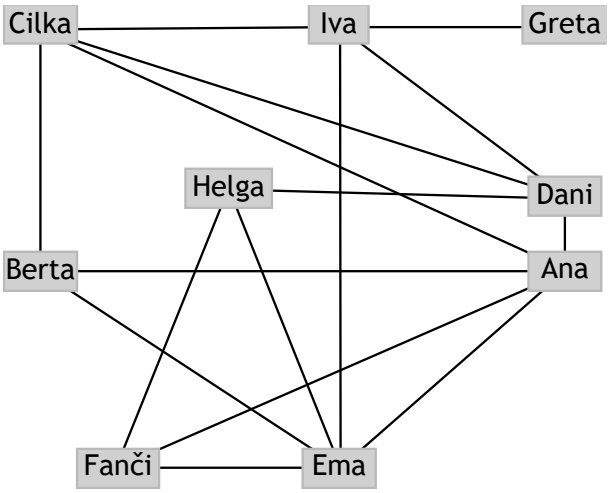
Sreda


Cilka
Iva
Greta
Helga
Dani
Berta
Ana
Fanči
Ema

Četrtek



Petek



Računalniško ozadje

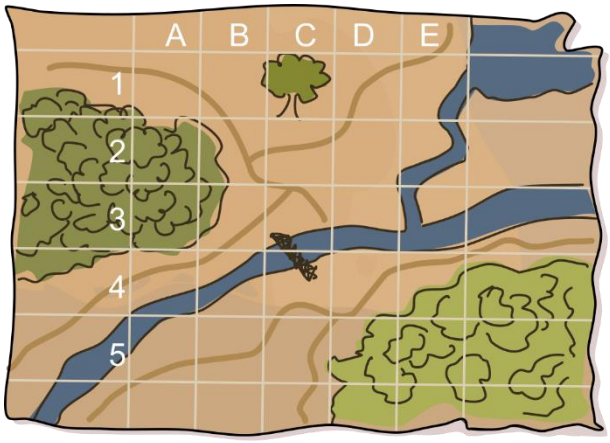
S pojavom velikih družbenih omrežij je preprosteje — in tudi pomembneje — raziskovati, kako se širijo ideje, (dez)informacije in vplivi v družbi. To zanima tako sociologe kot tudi podjetja in druge ustanove, ki želijo čim hitreje razširiti določeno informacijo — ali pa zajeziti njeno širjenje.

Model, ki ga uporablja ta naloga, se imenuje pragovni model in temelji na ideji, da vsak posameznik nadaljuje širjenje takrat, ko ga opazi pri dovolj velikem številu svojih stikov.

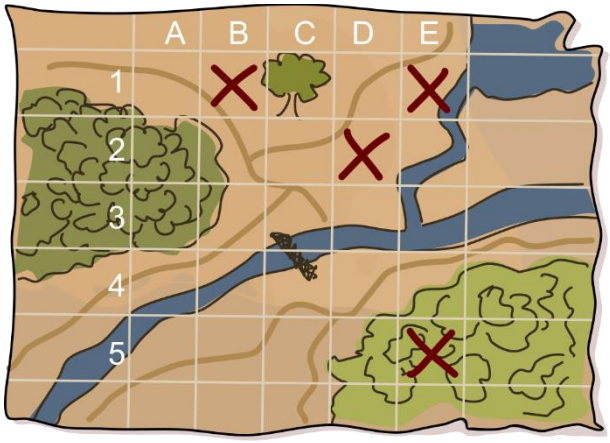
Zemljevid

državno, 6. do 9. razred, srednja šola

Matevž je našel dve dobri skrivališči za shranjevanje svojih dragocenosti. Da bi si ju zapomnil, ju želi na spodnjem zemljevidu označiti z znakoma "X". A če bo zemljevid našla Karmen, bo vedela, kje naj jih išče!



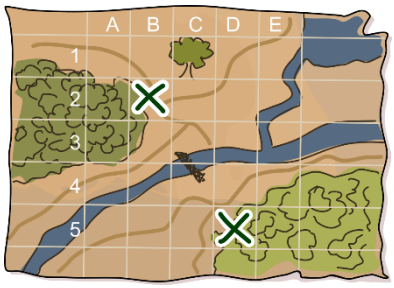
Da bi zmedel Karmen, Matevž naključno doda nekaj znakov "X" v druge kvadrate zemljevida in poskrbi, da je skupno število znakov "X" v vsaki vrstici in vsakem stolpcu sodo. (Opomba: tudi 0 je sodo število.) Nato izbriše obe črki "X", ki kažeta njegovo skrivališče. Tako nastane zemljevid:



Kje sta Matevževi skrivališči?

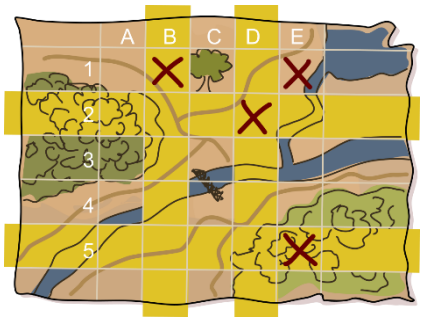
Rešitev

Skrivališči sta na mestih B2 in D5:



Če ju želimo najti, si ogledamo končni zemljevid in ugotovimo, da v dveh vrsticah in dveh stolpcih število znakov "X" ni sodo: v vrsticah 2 in 5 ter v stolpcih B in D.

Na sliki so vrstice in stolpci označeni z rumeno barvo:

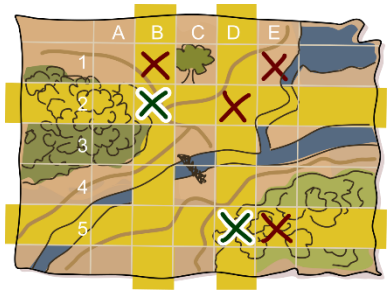


Dva izbrisana "X", ki kažeta na pravi skrivališči, morata biti nekje v označenem območju.

Na presečišču rumenih vrstic in stolpcev so 4 kvadrati. Zgornji desni kvadratek na presečišču (D2) je že označen z "X": tam ne more biti skrivališča. Da bi v drugi vrstici obnovili sodo število znakov "X", ga moramo dodati v zgornji levi rumeni kvadratek preseka (B2): zdaj poznamo eno skrivališče, to je B2.

Drugo skrivališče lahko določimo na naslednji način: ko smo našli in označili prvo skrivališče, ostane liho število "X" le v vrstici 5 in v stolpcu D, torej mora biti drugo skrivališče v D5.

Spodnja slika prikazuje zemljevid, preden je Matevž izbrisal "X", s sodim številom "X" v vsaki vrstici in stolpcu:



Računalniško ozadje

Za preverjanje pravilnosti podatkov v vrsticah in stolpcih smo uporabili pariteto: preverili smo, v katerih vrsticah in stolpcih je vsota števila križcev soda, in tako ugotovili, v katerih je prišlo do napake.

Bobrove igre

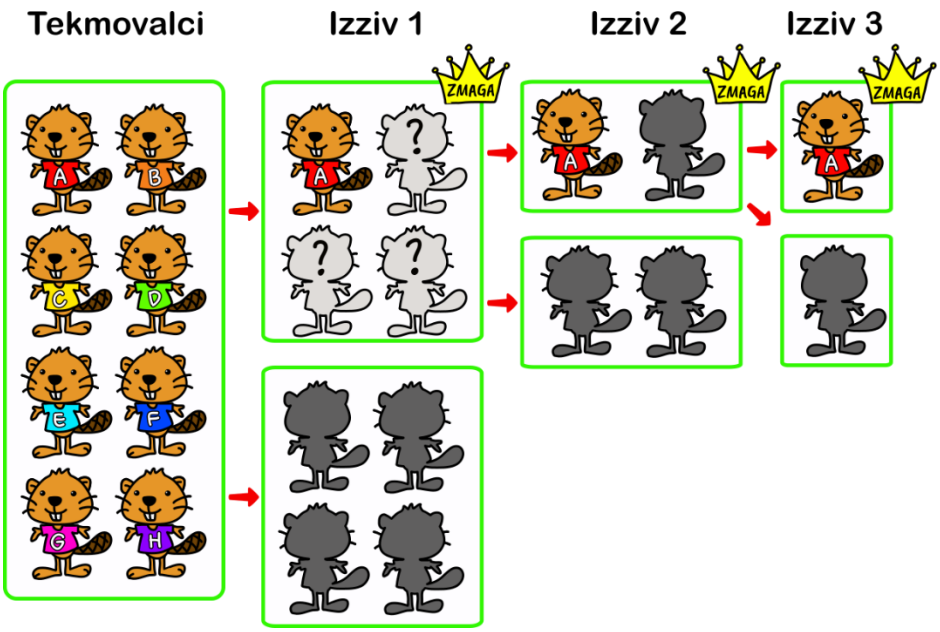
državno, 8. in 9. razred, srednja šola

Bobrove igre so letno tekmovanje, ki ga sestavljajo trije izzivi, v katerih zmaga ekipa, par oziroma posameznik z najvišjim seštevkom točk.

V prvem izzivu se vsi tekmovalci razdelijo v dve ekipi po štiri, kot določi žreb. Zmagovalna ekipa se v drugem izzivu s ponovnim žrebom razdeli v dva para. Zmagovalca drugega izziva se za zmago pomerita v finalnem tretjem izzivu. Zmagovalec iger je tisti tekmovalec, ki zmaga v vseh treh izzivih.

Ada je bila srečna zmagovalka letošnjih Bobrovih iger. Za vsak izziv so posamezne točke podane v tabeli.

Ime	Ada	Bojan	Cilka	Daša	Erik	Franci	Greta	Hajdi
Izziv 1	15	16	19	18	17	20	19	19
Izziv 2	20	27	30	24	28	24	30	30
Izziv 3	10	14	11	15	16	13	9	12



Kateri trije bobri so bili skupaj z Ado v prvem izzivu?

Rešitev

Ostali trije člani Adine ekipe so Daša, Franci in Greta.

Tretji izziv je individualna tekma. Ker je Greta edini igralec, ki ima manj točk kot Ada, se je Ada v tretjem izzivu pomerila z Greto, torej sta morali biti v drugem izzivu v isti ekipi.

Njuna vsota točk v drugem izzivu je 50. Ta vsota mora biti višja od vsote točk druge dvočlanske ekipe. Igralca Daša in Franci sta edina, katerih vsota točk v drugem izzivu je manjša od 50. Zato morata biti v isti ekipi kot Ada in Greta v prvem izzivu. Ker zdaj poznamo te člane, nam pravzaprav ni treba preverjati članov druge ekipe.

To torej pomeni, da je v prvem izzivu ekipa (Ada, Daša, Franci, Greta) imela 72 točk, medtem ko je ekipa (Bojan, Cilka, Erik, Hajdi) imela 71 točk. Zato zmaga Adina ekipa. V drugem izzivu je imela ekipa (Ada, Greta) 50 točk, ekipa (Daša, Franci) pa 48 točk. Ponovno zmaga Adina ekipa. V tretjem izzivu zmaga Ada z 10 točkami proti Greti z 9 točkami. Tako je Ada skupna zmagovalka.

Računalniško ozadje

Če bi za reševanje te naloge uporabili kombinatoriko, bi morali za iskanje ustrezne rešitev preveriti 35 kombinacij, to pa bi nam vzelo kar nekaj časa.

Računalničarji zato v podobnih primerih poskušamo najti bolj učinkovito (in hitrejšo) metodo za rešitev problema. Pri tej nalogi smo hitro ugotovili, da se jo najhitreje reši, če začnemo pri končnem izidu kot prikazuje naša razlaga rešitve. To metodo, ki jo imenujemo vzvratno iskanje, pogosto uporabljamo, kadar moramo upoštevati kakšne omejitve.

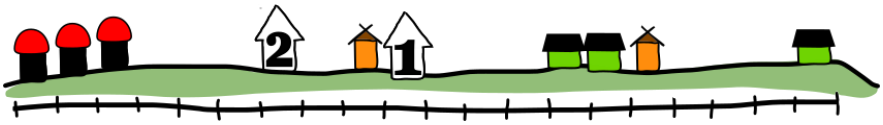
Širjenje vasi

državno, 8. in 9. razred, srednja šola

Z leti so se vasi Zeljendol  , Jagodno  in Korenčkovje  precej razširile in hiše vseh treh vasi so se začele že prepletati. Vaščani pa vseeno vedo, kateri vasi pripada posamezna hiša. O pravilu vemo tole.

- Upoštevajo X hiš, ki so najbližje novi hiši.
- Nova hiša bo pripadla vasi, kateri pripada večina izmed teh hiš.
- Če je rezultat neodločen, se med vasmi, ki imajo večino, odločijo glede na to, katera izmed hiš v teh vaseh je najbližja novi hiši.
- Če je rezultat še vedno neodločen, žrebajo.

Žal ne poznamo vrednosti X. Lahko pa sklepamo. Zgradili bodo hišo 1 in nato še hišo 2. Hiša 2 bo pripadla vasi Jagodno. Kakšna je najmanjša možna vrednost X, da bo res tako?



Rešitev

Da hiša 2 spada v Jagodno, mora biti X najmanj 5.

Nalogo lahko rešimo tako, da za različne vrednosti X po vrsti preverimo, pod katere vasi spadata obe novi hiši.

Če je X = 1, se hiša 1 dodeli Korenčkovju, prav tako pa tudi hiša 2, saj je obema novima hišama najbližja sosednja hiša iz Korenčkovja.

Tudi pri X = 2 obe novi hiši spadata pod Korenčkovje. Hiša 1 pride pod Korenčkovje, ker ima dve različni najbližji hiši (ena iz Korenčkove in druga iz Zeljnik), a je hiša iz Korenčkovja najbliže. Zato tudi hiša 2 spada pod Korenčkovje, saj sta njej najbližji dve hiši iz te vasi.

Če je X = 3, se hiša 1 dodeli v Zeljendol, saj sta iz te vasi dve izmed treh najbližjih hiš. Tri najbližje sosede hiše 2 pa so vsaka iz druge vasi, zato hiša 2 spada pod Korenčkovje, iz katere je tudi njej najbližja soseda.

Pri X = 4 ima hiša 1 dve sosedi iz Korenčkovja in dve iz Zeljendola. Ker je hiši 1 najbližja hiša iz Korenčkovja, bo tej vasi pripadala tudi hiša 1. Podobno ima tudi hiša 2 po dve sosedi iz iste vasi (dve iz Korenčkovja in dve iz Jagodnega), a najbližja soseda je iz Korenčkovja, zato tudi hiša 2 spada pod Korenčkovje.

Pri X = 5 pa gledamo pet najbližjih sosednjih hiš in za hišo 1 so to dve iz Korenčkovja, dve iz Zeljendola in ena iz Jagodnega. Ker je najbližja soseda iz Korenčkovja, tudi hiša 1 spada v Korenčkovje vas. Hiša 2 pa ima dve sosedi iz Korenčkovja in tri sosede iz Jagodnega, zato je dodeljena v Jagodno.

Tudi pri X = 6 in X = 7 spada hiša 2 pod Jagodno, a to ni najmanjša vrednost X, ki jo iščemo v nalogi.

Računalniško ozadje

Pravilo, s katerim novo hišo dodelimo določeni vasi, je primer klasifikacijskega algoritma z imenom k-najbližjih sosedov (angl. k-nearest neighbors ali kNN). Algoritem kNN uvrsti vsak nov element tako, da pogleda k najbolj podobnih elementov, ki so že bili uvrščeni. V naši nalogi je vlogo k prevzela vrednost X.

V strojnem učenju se algoritem kNN pogosto uporablja, ker je enostaven za implementacijo in ne zahteva prilagajanja kompleksnega modela danim podatkom.

Varnostni sistem državno, 8. in 9. razred, srednja šola

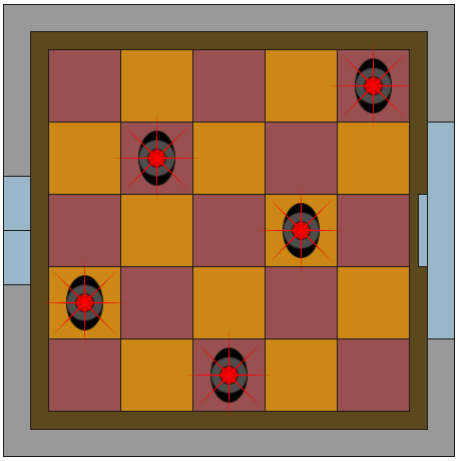
V banki BPB (Bobrova poslovna banka) želijo postaviti nov varnostni sistem v sobi z bančnim sefom. Nabavili so pet detektorjev. Vsak detektor pošilja laserske žarke v osmih smereh do stene sobe.

Če vlomilec prekriža pot laserskemu žarku iz katerega koli detektorja, se sproži alarm. Vendar se alarm sproži tudi v primeru, če je na poti laserskega žarka drug detektor.

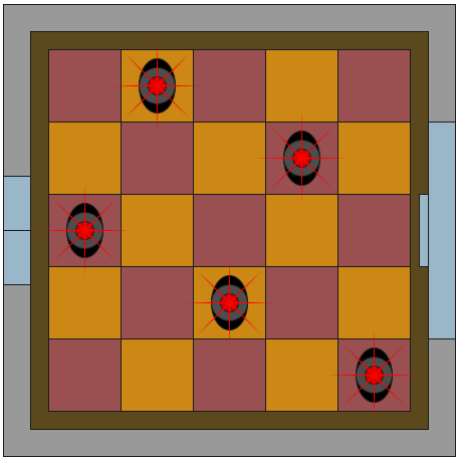
V BPB želijo postaviti vseh pet detektorjev na kvadratne ploščice v sobi tako, da z laserji pokrijejo vse preostale ploščice v sobi, a da se detektorji medsebojno ne aktivirajo.

Predlagali so štiri različne možnosti postavitve, a ena od njih ne ustreza zahtevam BPB. Katera?

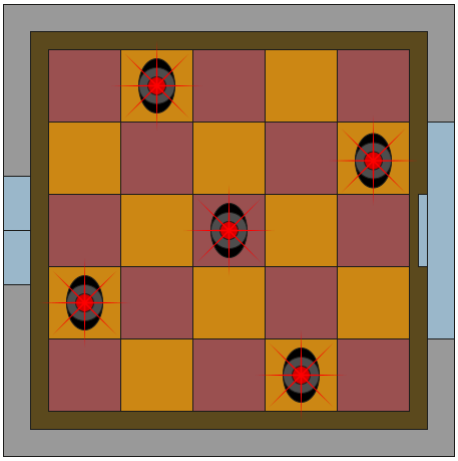
A)



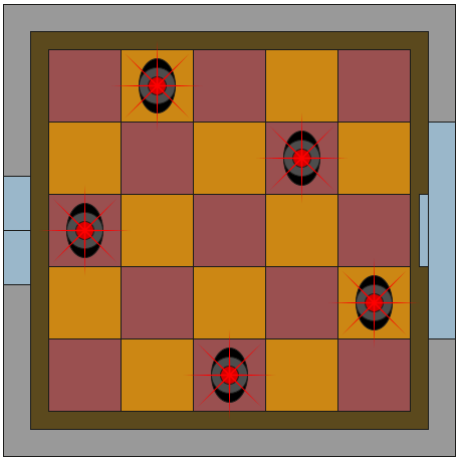
B)



C)



D)



Rešitev

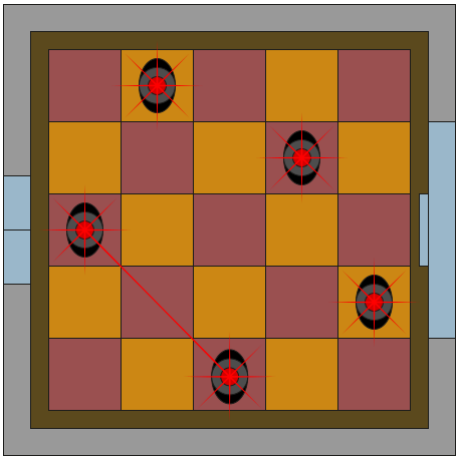
Pravilen odgovor je D.

Ko iščemo rešitev, ki ustreza zahtevam BPB, takoj opazimo, da je v vsaki vrstici in v vsakem stolpcu ploščic lahko le en detektor. Ker detektorji pošiljajo laserske žarke vodoravno in navpično, bi se v primeru več detektorjev v isti vrstici (vodoravno) ali stolpcu (navpično) sprožil alarm. Če torej detektorje postavimo tako, da je vsak od njih v svoji vrstici in stolpcu, smo z laserji pokrili celo sobo in tako prestregli morebitnega vlomilca.

Vendar detektorji pošiljajo laserske žarke tudi diagonalno, zato moramo pri njihovi postavitvi paziti tudi na to, da se medsebojno ne ovirajo. To pomeni, da dveh detektorjev ne smemo postaviti na isto diagonalo.

V splošnem lahko rečemo, da mora pri pravilni rešitvi vsak par različnih detektorjev zadostovati naslednjim pogojem. Če prvi detektor stoji v stolpcu *x* in vrstici *y* ter drugi detektor stoji v stolpcu *a* in vrstici *b*, potem mora veljati:

- *x* in *a* ne smeta biti enaka zaradi vodoravnih laserjev,
- *y* in *b* ne smeta biti enaka zaradi navpičnih laserjev,
- razlika med *x* in *a* (v absolutni vrednosti) ne sme biti enaka razliki med *y* in *b* (v absolutni vrednosti) zaradi diagonalnih laserjev.



Vsi ti pogoji veljajo za odgovore A, B in C. Pri odgovoru D pa sta dva detektorja (eden v stolpcu 1 in vrstici 3, drugi v stolpcu 3 in vrstici 5) na isti diagonali, kar bi sprožilo alarm. Poleg tega sta na isti diagonali tudi detektorja v stolpcu 2 in 5.

Računalniško ozadje

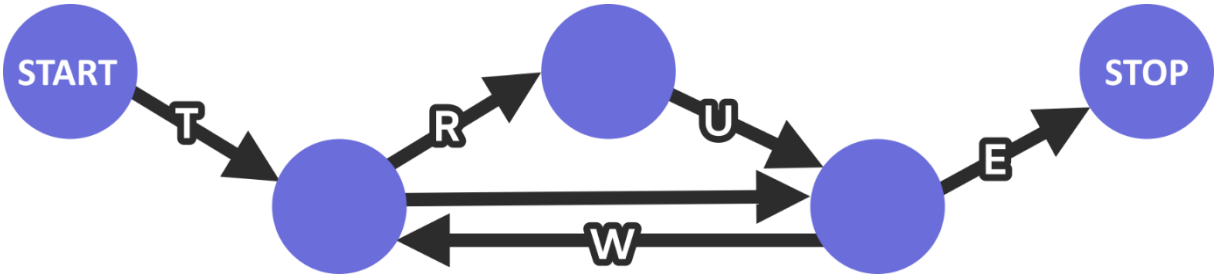
Problem, ki smo ga reševali v nalogi, je nekoliko spremenjen problem osmih dam na šahovnici, ki je znan primer problema, kjer moramo pri rešitvi zadostiti vsem podanim omejitvam. Podobne probleme srečamo pri samodejnem planiranju, računalniškem jezikoslovju (pri razpoznavanju jezika) ali razvrščanju virov (npr. kdo bo delal s katerimi delovnimi sredstvi). Čeprav so taki problemi videti preprosti, se lahko hitro zelo zakomplicirajo in za njihovo uspešno reševanje potrebujemo računalnik. Računalnik poskuša rešitev sestaviti po korakih in ko naleti na kršitev enega od pravil, se vrne korak nazaj in poskusi z drugo opcijo. To ponavlja, dokler ne dobi rešitve, ki ustreza vsem podanim omejitvam. Takemu postopku rečemo *sestopanje* (angl. *backtracking*).

Besede

državno, srednja šola

Breda sestavlja različne »besede« s pomočjo diagrama na spodnji sliki. Vedno začne v krogu z oznako START in sledi puščicam, dokler ne pride do kroga z oznako STOP. Vsako črko, ki jo sreča na poti (na puščici, ki ji je sledila), si zapiše in tako ustvari »besedo«.

Sestavi lahko, na primer, besede TE, TRUE ali pa TWWWE.



Koliko različnih besed z 8 črkami lahko sestavi?

Rešitev

Lahko sestavi 9 različnih besed dolžine 8.

Pri sestavljanju besede mora vedno uporabiti T kot prvo črko in E kot zadnjo črko. Torej moramo poiskati različne besede dolžine 6 iz črk R, U in W.

Problem lahko rešimo tako, da najprej rešimo manjši podproblem: na koliko različnih načinov lahko ustvarimo besede določene dolžine?

dolžina besede	število različnih načinov	besede
1	1	W (sledimo prazni puščici, nato W in spet prazni puščici)
2	2	WW, RU
3	3	WWW, WRU, RUW
4	4	WWWW, WWRU, RUWW, WRUW
5	6	WWWWW, WWWRU, WWRUW, WRUWW, RUWWW, RUWRU
6	9	WWWWWW, WWWWRU, WWWRUW, WWRUWW, WRUWWW, RUWWWW, WRUWRU, RUWRUW, RUWWRU

Breda lahko torej sestavi naslednjih 9 različnih besed: TWWWWWWE, TWWWWRUE, TWWWRUWE, TWWRUWWE, TWRUWWWE, TRUWWWWE, TWRUWRUE, TRUWRUWE, TRUWWRUE.

Računalniško ozadje

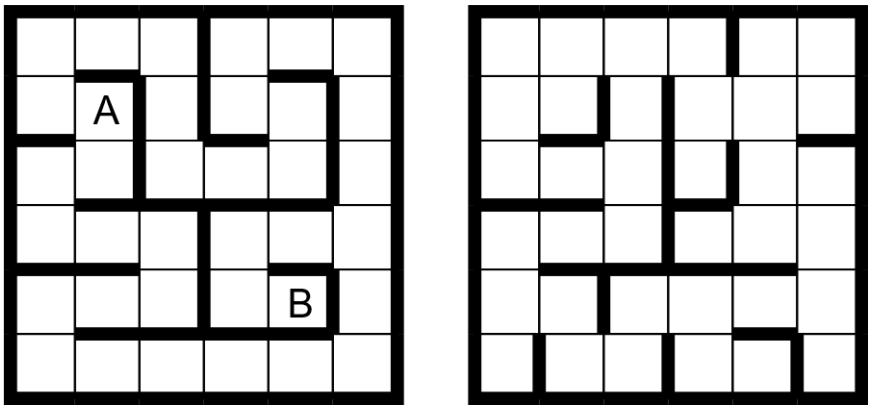
Računalniki so zelo dobri pri preiskovanju vseh možnosti. V naši nalogi smo se morali precej potruditi, da smo sestavili vse možne besede in da pri tem nismo dvakrat zapisali iste besede. Za računalnike pa je to, da ustvari zaporedje črk z določenimi omejitvami, zelo enostavna naloga, saj je diagram v naši nalogi pravzaprav končni avtomat.

Sisteme, podobne temu v nalogi, lahko uporabimo za generiranje gesel za tekmovalce na Bobru ali pa za generiranje oznak na avtomobilskih tablicah.

Borgovska ladja

državno, srednja šola

Seven se nahaja na borgovskem kvadru. Ta ima dve nadstropji; njuna zemljevida sta na sliki. Seven ne more skozi steno, lahko pa se prežarči (*beama*) skozi tla oz. strop.



Pot z enega na drugo polje ji vzame eno sekundo. Pot skozi strop (oz. tla) na istoležno polje v prvem oz. drugem nadstropju ji vzame pet sekund.

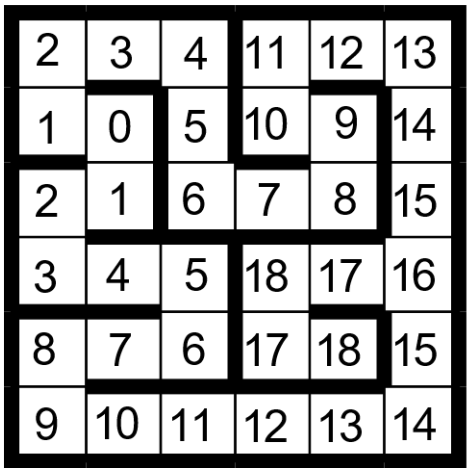
Da razstreli ladjo, mora priti s točke A do točke B. Najmanj koliko časa potrebuje za to?

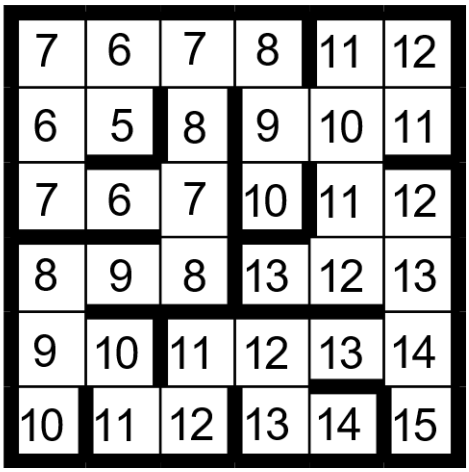
Rešitev

Pot ji v najboljšem primeru vzame 18 sekund.

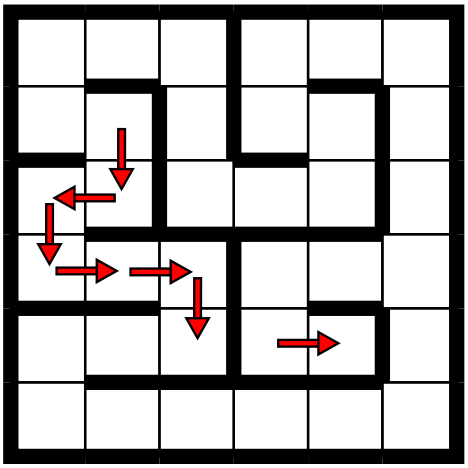
Za vsako polje na ladji, v obeh nadstropjih, zapišimo najkrajši čas, ki ga potrebuje za pot s polja A.

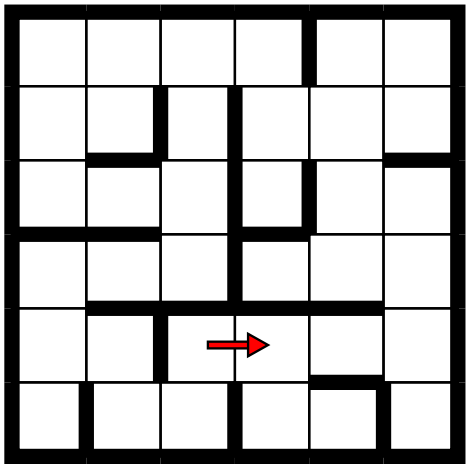
- Do polja A potrebuje 0 sekund.
- Do sosednjih polj potrebuje eno sekundo, do polja nad njim pa pet sekund.
- Isto ponovimo za vsa polja, na katera smo vpisali enice: na njihove sosede (če so še prazni) zapišemo dvojke, na polja nad enkami pa 6.
- Nato ponovimo vajo s polji, na katerih so dvojke ...
- ... in tako naprej, dokler ne pridemo do polja B.
- Pri tem pazimo, da vedno upoštevamo le najkrajši čas, tj. ali prehod s sosednjega polja ali pa iz drugega nadstropja, karkoli je manj.





Izkaže se, da se ji splača le enkrat skočiti v drugo nadstropje. Idealna pot je takšna.





Računalniško ozadje

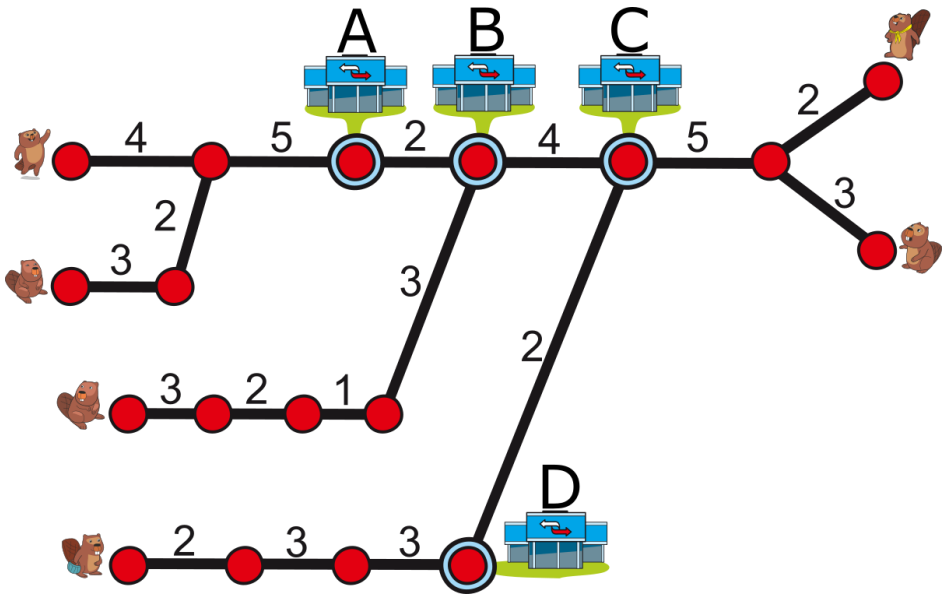
"Poplavljanje" je pogost postopek, povezan z algoritmom iskanja v širino ali, tudi, iskanja najkrajših poti. V našem primeru smo pri poplavljanju morali upoštevati še dodatno omejitev, ki jo predstavlja trajanje prežarčenja skozi strop oziroma tla, ki ga je možno narediti na vsaki točki.

Podzemna železnica

državno, srednja šola

Šest prijateljev živi vsak na svoji strani mesta. Dogovorili so se za srečanje na eni izmed glavnih postaj podzemne železnice, to so A, B, C in D. Vsak od njih se bo s svoje domače postaje pripeljal s podzemno železnico. Vsi prijatelji vstopijo na vlak ob istem času in se želijo čim hitreje srečati na izbrani glavni postaji.

Spodnja slika prikazuje omrežje podzemne železnice z označenimi postajami (rdeči krogci) na vsaki liniji in časi potovanja med posameznimi postajami (v minutah). Na vsaki postaji se vlak zadrži natanko eno minuto, da potniki izstopijo in vstopijo.



Na kateri izmed glavnih postaj se morajo srečati prijatelji, da se čim prej dobijo?

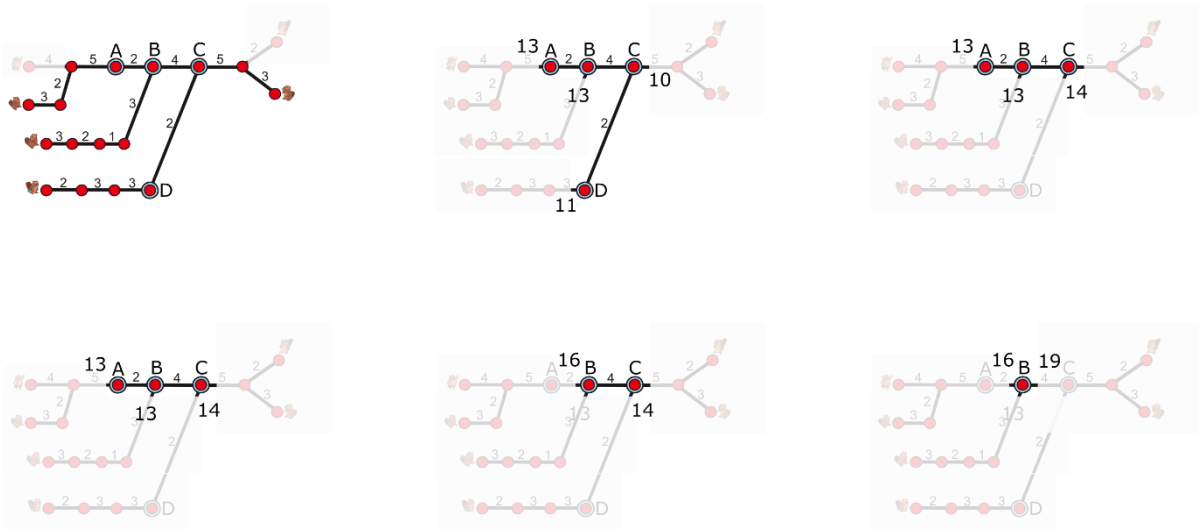
Rešitev

Srečati se morajo na postaji B.

Za rešitev problema moramo za vsakega od prijateljev izračunati čas potovanja od njegove vstopne postaje do vseh štirih glavnih postaj (A, B, C in D). Za vsakega prijatelja torej izračunamo vse možnosti in nato pogledamo, koliko časa porabi zadnji od njih, da prispe na vsako od postaj. To je lahko zelo zamudno delo in nam vzame veliko časa.

Problem pa lahko rešimo tudi z dekompozicijo (kar pomeni, da problem razbijemo na več manjših delov) in abstrakcijo (da se odločimo, katere linije so najbolj pomembne za izračun in ostale enostavno zanemarimo). Ker nas zanima le, koliko časa potrebuje za prihod na neko postajo zadnji od prijateljev, lahko upoštevamo le najdaljši čas potovanja za vsako linijo.

Spodnja slika prikazuje eno od zaporedij korakov k rešitvi te naloge:



Če gledamo zgornja dva prijatelja na levi in čas, ki ga potrebujata do postaje A, potrebuje prvi (zgornji) 11 minut (1 + 4 + 1 + 5 = 11), drugi (spodnji) pa 13 minut (1 + 3 + 1 + 2 + 1 + 5 = 13). Torej moramo od tu dalje upoštevati le drugega prijatelja, ki prispe kasneje (njegov čas potovanja do A je 13 minut).

Podobno velja tudi za oba prijatelja na desni, katera se lahko srečata na postaji C v 10 minutah (zgornji potrebuje 1 + 2 + 1 + 5 = 9 minut, spodnji pa 1 + 3 + 1 + 5 = 10 minut).

Spodnji prijatelj na levi potrebuje do postaje D 11 minut (1 + 2 + 1 + 3 + 1 + 3 = 11), torej bo na postaji D, preden se prva dva prijatelja srečata na postaji A. Zato lahko potuje naprej do postaje C, za kar potrebuje 14 minut (11 + 1 + 2 = 14).

Preostali od prijateljev na levi potrebuje do postaje B 13 minut (1 + 3 + 1 + 2 + 1 + 1 + 1 + 3 = 13).

V 13 oz. 14 minutah so prijatelji lahko na postajah A, B in C, vendar se morajo vsi dobiti na isti postaji. Vidimo, da mora biti to postaja B, ki je vmes med postajama A in C. Do postaje B bodo najdlje potovali prijatelji, ki so se zbrali na postaji C, in sicer še dodatnih 5 minut (1 + 4 = 5), torej bodo najpočasnejši za pot potrebovali 19 minut (14 + 5 = 19).

Dobljeno rešitev lahko preverimo tudi tako, da izračunamo čas potovanja do vsake glavne postaje iz obeh smeri, leve in desne. Čase smo zapisali v tabelo:

Čas potovanja do A	**Čas potovanja do B**	**Čas potovanja do C**	**Čas potovanja do D**
z leve: 13	z leve: 16	z leve: **21**	z leve: 11
z desne: **22**	z desne: **19**	z desne 10	z desne: **24**

Vidimo, da prijatelji potrebujejo 22 minut, da se vsi srečajo na postaji A, 19 minut za srečanje na postaji B, 21 minut za srečanje na postaji C in 24 minut za srečanje na postaji D. Torej je postaja B najboljša izbira za čim hitrejše srečanje.

Računalniško ozadje

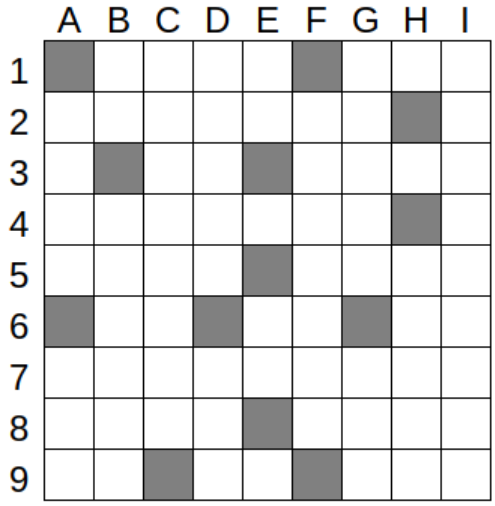
Za reševanje problemov v računalništvu lahko uporabimo dinamično programiranje, pri katerem reševanje kompleksnega problema poenostavimo tako, da ga razbijemo na bolj enostavne podprobleme. Dinamično programiranje se veliko uporablja pri optimizacijskih problemih, kot je na primer iskanje največje ali najmanjše vrednosti nečesa. Tudi pri rešitvi naše naloge smo si pomagali z dinamičnim programiranjem.

V naši nalogi je omrežje podzemne železnice prikazano z grafom. Postaje so vozlišča grafa, proga pa povezava v grafu. Vsak del proge (povezava) med dvema postajama ima pripisan tudi čas potovanja, zato imamo utežen graf, kjer so povezavam pripisane tudi določene vrednosti ali uteži. Iskanje poti v grafu, ki ima najmanjšo vsoto vrednosti, je optimizacijski problem.

Pot robota

državno, srednja šola

Robot začne svojo pot na beli celici mreže na sliki, obrnjen pa je lahko v katero koli od štirih smeri.



Nato robot obišče natanko 8 drugih belih celic, kot sledi:

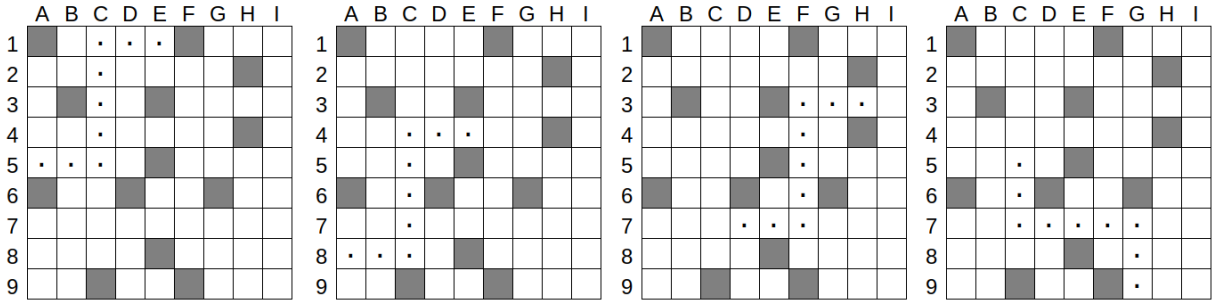
1. Premakne se 2 celici naprej.
2. Obrne se za 90 stopinj v levo (in pri tem ostane na trenutni celici).
3. Premakne se 4 celice naprej.
4. Obrne se za 90 stopinj v desno (in pri tem ostane na trenutni celici).
5. Premakne se 2 celici naprej.

Koliko je vseh različnih začetnih celic, s katerih robot lahko začne svojo pot?

Rešitev

Pravilen odgovor je 8.

Najprej ugotovimo, da imamo 4 različne možnosti za celice, ki jih obišče robot:



Te štiri možne poti najlažje poiščemo tako, da poiščemo 5 zaporednih belih celic v mreži in nato poskusimo pot podaljšati še za 2 celici pravokotno na vsaki strani. Pri tem pazimo, da je

prvi obrat na poti v levo, drugi pa v desno (in ne obratno). Pet zaporednih belih celic iščemo sistematično, na primer od leve proti desni in od zgoraj navzdol. Le tako smo lahko prepričani, da nismo izpustili kakšne od možnosti.

Ker je pot robota simetrična, lahko robot začne na kateri koli strani poti pri vsaki od zgoraj narisanih štirih možnosti. Torej imamo skupaj 8 različnih možnih začetnih celic (A5, E1, A8, E4, D7, H3, C5 in G9).

Računalniško ozadje

Naloga je preprost primer rotacijsko nespremenljivega ujemanja slike (ali predloge), kar je ena od nalog s področja obdelave slik. V nalogi smo določili predlogo (koraki 1 do 5 našega robota), ki jo iščemo, mreža pa je bila slika, v kateri smo iskali predlogo. Ujemanje predloge se uporablja, med drugim, pri razpoznavanju obrazov in obdelavi medicinskih slik.