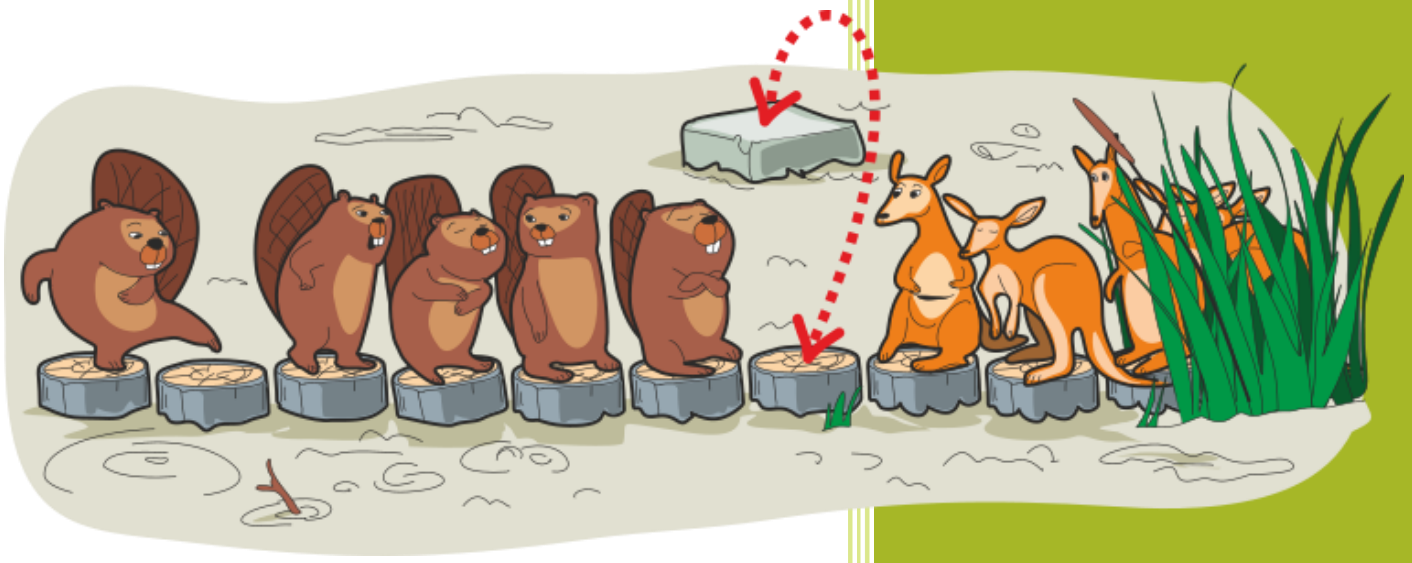
Bober 2021/22

Bebras

ACM Slovenija

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

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



Naloge in rešitve

ACM Slovenija

CC BY SA

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

Alenka Kavčič (UL FRI)
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)

Avtor slike na naslovnici

Vaidotas Kinčius, Litva

Kazalo nalog

Gujeolpan 1

2. razred

ZA KOREJSKO JED GUJEOLPAN POSTAVIMO NA KROŽNIK DEVET RAZLIČNIH SESTAVIN.

KATERI KROŽNIK JE ENAK KROŽNIKU NA ZGORNJI SLIKI, ČE GA ZAVRTIMO?

A) 

B) 

C) 

Rešitev

PRAVILEN JE ODGOVOR C).

PRI ODGOVORU A) MANJKA ENA  IN JE ENA  VEČ.

PRI ODGOVORU B) JE MED  IN  SESTAVINA  NAMESTO  .

Računalniško ozadje

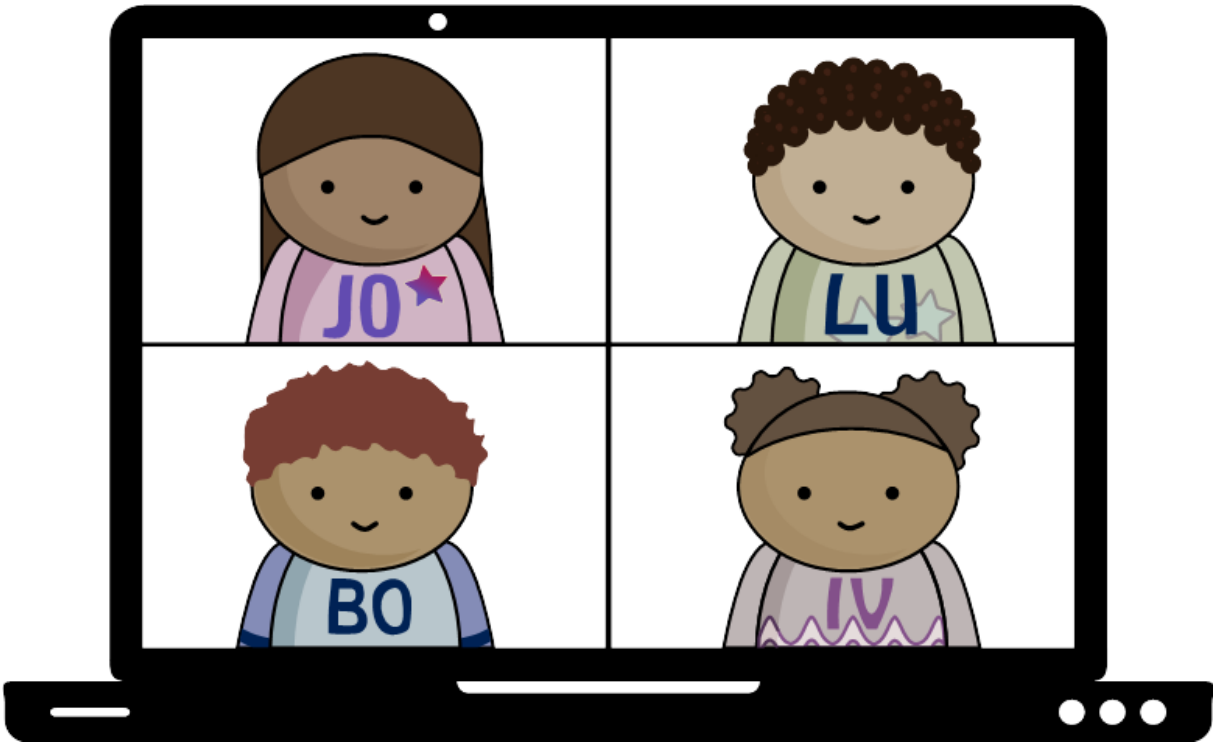
Pri nalogi gra za preverjanje enakosti različno zapisanih reči. S tem se računalnikarji pogosto spopadamo.

Zrcalna slika

2. razred

PRI VIDEO KLEPETU VIDIMO SVOJO SLIKO ZRCALNO, KOT V OGLEDALU. SLIKE OSTALIH VIDIMO NORMALNO.

PRIJATELJI BO, JO, LU IN VI SO NA VIDEO KLEPETU. EDEN OD NJIH TI JE POSLAL SLIKO SVOJEGA ZASLONA. KATERI?



Rešitev

ZASLON, KI GA GLEDAMO, UPORABLJA VI, ZATO JE NJENA SLIKA ZRCALNA. TO NAJLAŽJE VIDIMO, ČE POGLEDAMO NAPIS NA NJENI MAJICI.

Računalniško ozadje

Zanimivo, ne? Večina uporabnikov programov za klepet po videu se niti ne zaveda, da svojo sliko vidijo drugače, kot jo vidijo drugi.

Igrače

2. in 3. razred

BOBER JE DOBIL NAVODILA ZA SESTAVLJANJE IGRAČ. VSAKO IGRAČO SESTAVI PO NAVODILIH V POSAMEZNEM **STOLPCU**.

PRI KATERIH DVEH IGRAČAH SE **NI** DRŽAL NAVODIL?

Rešitev

DRUGA IN TRETJA IGRAČA IMATA NA TREBUHU NAPAČNA VZORCA, PRVA IN ČETRTA IGRAČA PA STA V SKLADU Z NAVODILI.

Računalniško ozadje

Ko rešujemo nalogo v bistvu opazujemo delovanje nekega preprostega programa.

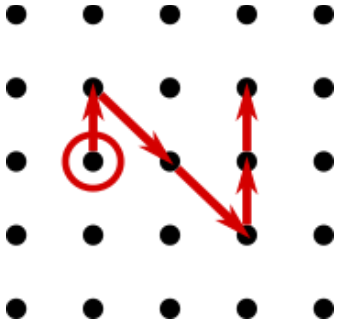
Od pike do pike

2. in 3. razred

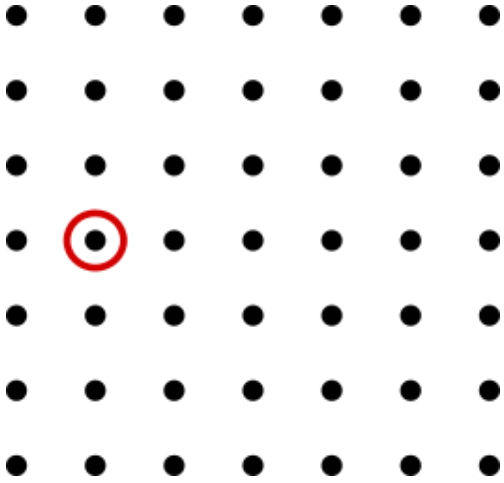
KO NA ROBOTU PRITISNEŠ TIPKE

,

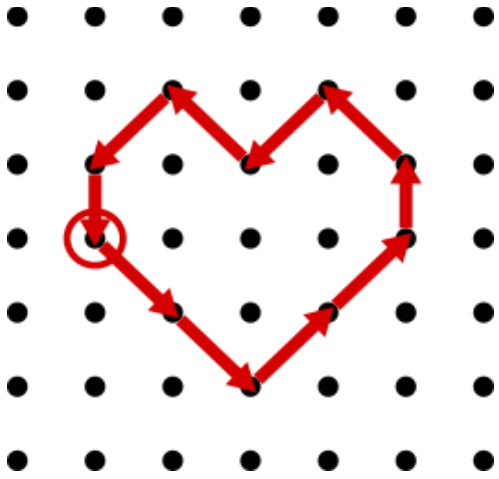
SE PREMAKNE TAKO:



NARIŠI PREMIKE ROBOTA, ČE NA ROBOTU PRITISNEŠ TIPKE:



Rešitev

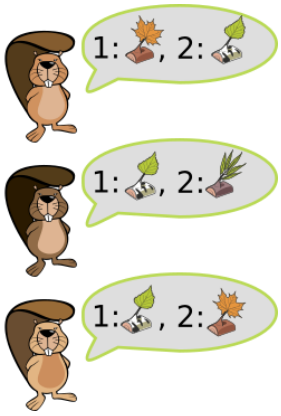


Računalniško ozadje

Tudi to je naloga iz spremljanja nekega preprostega programa.

Želje 1

2. in 3. razred



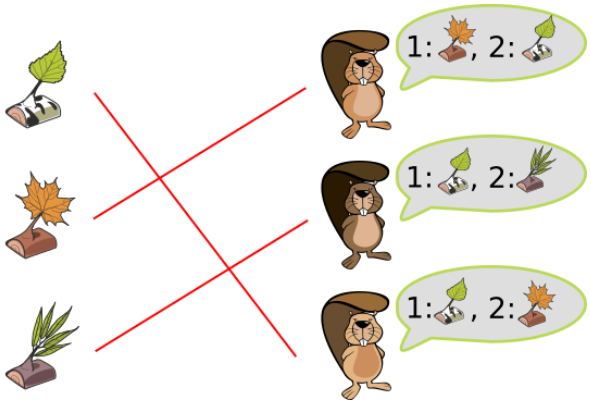
DRUŽINA BOBROV IMA TRI DARILA ZA SVOJE TRI OTROKE. VSAK BOBER NAJPREJ POVE, KATERO DARILO SI NAJBOLJ ŽELI IN KATERO JE NJEGOVO DRUGO NAJLJUBŠE.

DARILA MORAJO BITI RAZDELJENA TAKO, DA:

- ČIMVEČ BOBROV DOBI SVOJE NAJLJUBŠE DARILO.
- OSTALI BOBRI DOBIJO SVOJE DRUGO NAJLJUBŠE DARILO.

POVEŽI BOBRE Z DARILI, KI JIH BODO DOBILI.

Rešitev



PRVI BOBER LAHKO DOBI SVOJE NAJLJUBŠE DARILO, SAJ NOBEN OD PREOSTALIH BOBROV NIMA ENAKE PRVE ŽELJE.

KER SI DRUGI IN TRETJI BOBER NAJBOLJ ŽELITA DOBITI ISTO DARILO, OBEMA NI MOGOČE IZPOLNITI TE ŽELJE. KER JE PRVI BOBER DOBIL SVOJE NAJLJUBŠE DARILO, TO NI VEČ NA VOLJO. TRETJI BOBER ZATO TEGA DARILA NE MORE DOBITI, ČEPRAV MU JE VŠEČ. DARILO, KI JE VŠEČ DRUGEMU BOBRU, JE ŠE NA VOLJO, ZATO DRUGI BOBER DOBI TO DARILO, TRETJI BOBER PA DOBI SVOJE NAJLJUBŠE DARILO.

Računalniško ozadje

Problem, ki ga rešujemo, je podoben problemu stabilnih zakonov. Takšne naloge se ne pojavljajo le v računalništvu temveč še marsikje. Leta 2012 je bila Nagrada Švedske banke za ekonomske vede v spomin Alfreda Nobela (ki ji pravijo tudi Nobelova nagrada za ekonomijo) podeljena raziskovalcem, ki so uporabili podoben postopek za resnične probleme s področja ekonomije.

Gujeolpan 2

3. razred

MasterChef je svojim učencem pokazal, kako pripravi tradicionalno korejsko jed gujeolpan. To je jed, postrežena v osmerokotni posodi, ki je sestavljena iz različnih sestavin, razporejenih okoli pšeničnih palačink. Od učencev pričakuje, da bodo tudi oni pripravili jed na isti način, kot na spodnji sliki.

Katera od naslednjih posod ima enako razporejene sestavine kot MasterChefova? Upoštevaj, da je posodo mogoče zavrteti.

A)

B)

C)

D)

Rešitev

Pri odgovoru A) manjka ena  in je ena  več.

Pri odgovoru B) je  med  , namesto  .

Odgovor C) je pravilen.

Pri odgovoru D) manjka ena  in je ena  več.

Računalniško ozadje

Pri nalogi gre za preverjanje enakosti različno zapisanih reči. S tem se računalnikarji pogosto spopadamo.

Ribe

3. razred

Ribe plavajo v vrsti:

Ko Marko pove dve številki mest A in B, odplavajo stran vse ribe, ki so levo od A ali desno od B.

Če Marko pove števili 2 in 17, odplavajo stran riba na 1. mestu in ribe na mestih 18, 19 in 20. Ostanejo še:

Preden Marko izreče naslednji ukaz, ribe v tej vrsti na novo oštevilčimo od 1 do 16.

Začnimo z začetno vrsto 20 rib. Marko reče:

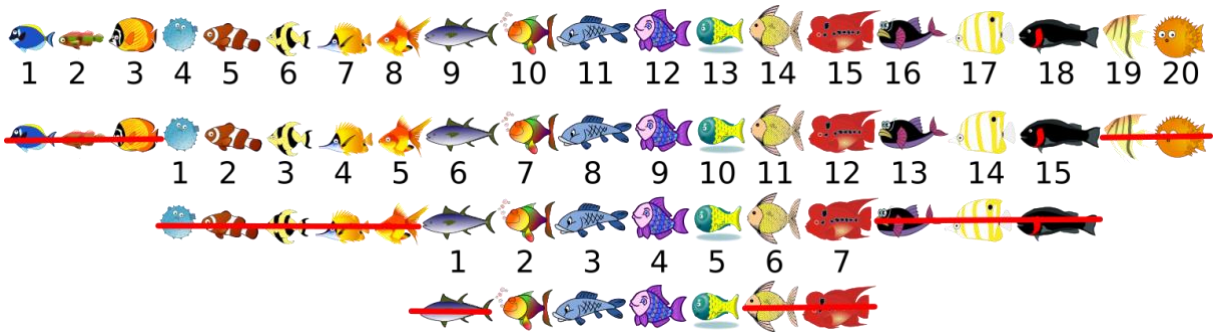
- Mesti 4 in 18.
- Mesti 6 in 12.
- Mesti 2 in 5.

Prečrtaj ribe, ki bodo odplavale stran:

Rešitev

Nalogo najlažje rešimo s štetjem. Za lažjo razlago bomo ribe oštevilčili.

- Na začetku imamo 20 rib (v prvi vrsti na spodnji sliki). Po ukazu »mesti 4 in 18« odplavajo prve tri ribe in zadnji dve.
- Ostane 15 rib (glej drugo vrsto). Po »mesti 6 in 12« odplava proč prvih pet rib in zadnje tri.
- Ostane 7 rib, kot je prikazano v tretji vrsti. Po ukazu »mesti 2 in 5« odplavajo stran prva in zadnji dve ribi v vrsti.
- Kot je vidno v zadnji vrsti, ostanejo 4 ribe, ki so bile na začetku na mestih 10 do 13.



Računalniško ozadje

Takole odstranjujemo dele tabel, enega izmed tipov podatkov v programih.

Darila

3. in 4. razred

Štirje prijatelji so se dogovorili, da si bodo darila letos izmenjali po naslednjih pravilih:

Pravilo 1: Nihče ne da darila samemu sebi.

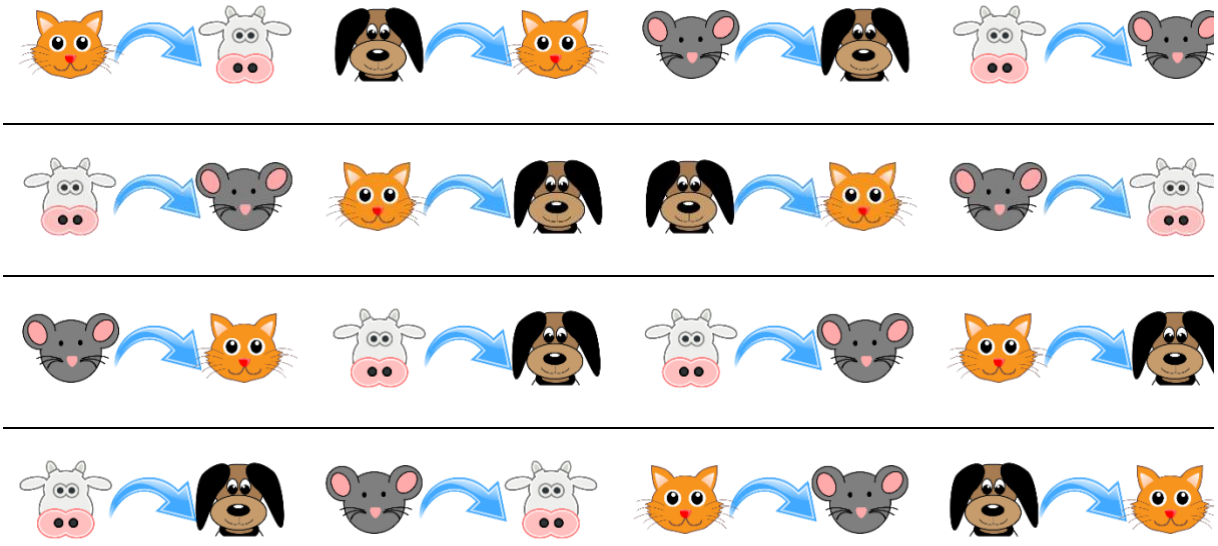
Pravilo 2: Vsak prijatelj podari eno darilo.

Pravilo 3: Vsak prijatelj prejme eno darilo.

Dva prijatelja bi si izmenjala darili tako: 

Muca Mija da darilo miški Maji in miška Maja da darilo muci Miji.

Štirje prijatelji si lahko darila izmenjajo na različne načine. Kateri od spodnjih **NI** skladen z dogovorom?



Rešitev

Tretji: krava podari dve darili, kužek pa dve darili prejme, s tem pa sta kršeni tako 2. kot 3. pravilo.

Računalniško ozadje

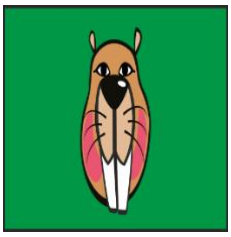
Pri programiranju velikokrat razmišljamo o tem, katerim pogojem zadošča kakšna reč.

Smešni filtri

3. in 4. razred

V aplikaciji za obdelavo fotografij lahko uporabiš štiri filtre:

Ko je nek bober uporabil filtra »dodaj rdečilo« in »preoblikuj obraz«, izgleda fotografija tako:

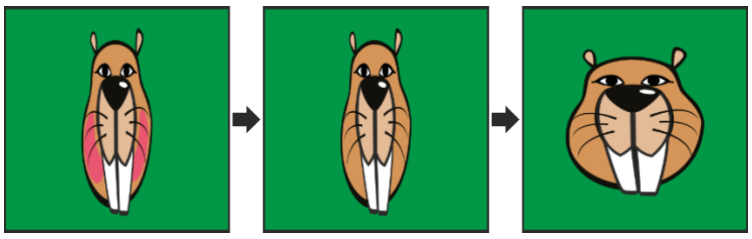


Kakšna je bila začetna fotografija?

A) 
B) 
C) 
D) 

Rešitev

Bober na izvorni sliki je imel velike zobe, saj ni bil uporabljen filter povečanja zob. Torej sliki A in C ne moreta biti začetni sliki. Prav tako je imel bober na začetni sliki brke, saj jih ima tudi na obdelani sliki (filter dodajanja brkov ni bil uporabljen oz. sploh ne obstaja), zato tudi slika B ne more biti začetna slika. Torej je začetna slika D. To lahko preverimo tudi tako, da odstranimo oba dodana filtra. Najprej odstranimo dodano rdečilo in nato še preoblikovanje obraza ter dobimo sliko D:



Kolački

3. do 7. razred

V pekarni izdelujejo kolačke, okrašene s prelivom, posipom in sadjem. Vrsto preliva, posipa in sadja zamenjajo po vsakem kolačku.

Prelivi se menjajo v naslednjem vrstnem redu:

zelen → bel → rdeč → moder → zelen → bel → rdeč → moder → zelen → bel → ...

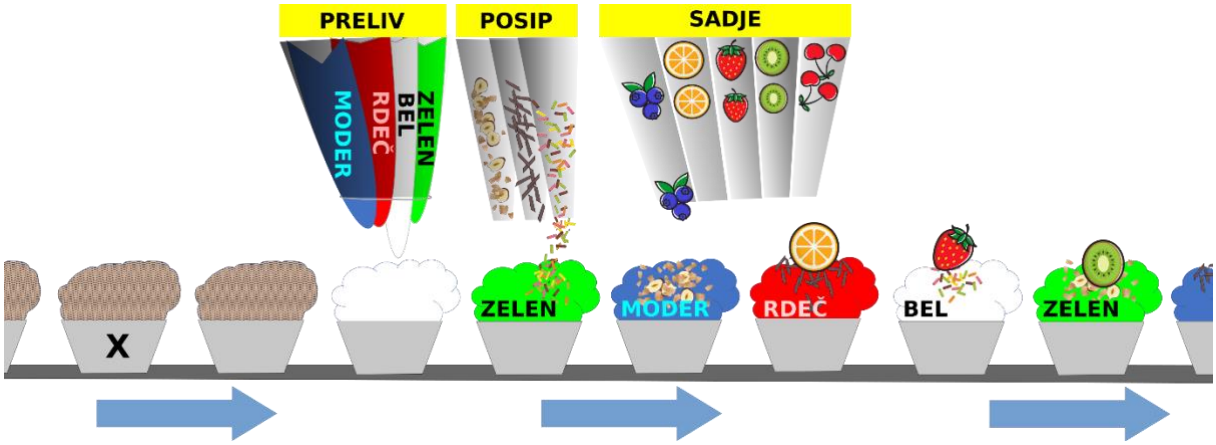
Posipi se menjujejo v naslednjem vrstnem redu:

mrvice → koščki čokolade → oreščki → mrvice → koščki čokolade → oreščki → ...

Sadje se menjuje v tem vrstnem redu:

češnja → kivi → jagoda → pomaranča → borovnice → češnja → kivi → jagoda → ...

Spodnja slika prikazuje kolačke na produkcijski liniji v nekem trenutku, ko se pomikajo od leve proti desni.



Kakšen bo kolaček, označen z X?

A) Rdeč z mrvicami in pomarančo

B) Bel s koščki čokolade in kivijem

C) Moder z oreščki in jagodo

D) Moder z mrvicami in pomarančo

Rešitev

Kolaček, označen z X, bo moder z mrvicami in pomarančo.

Zadnji dokončani kolaček je rdeč s koščki čokolade in pomarančo. Iščemo 5. kolaček, ki bo izdelan za njim. Prvi bo moder z oreščki in borovnicami. Drugi bo zelen z mrvicami in češnjo. Tretji bo bel s koščki čokolade in kivijem. Četrti bo rdeč z oreščki in jagodo. Peti, ki ga iščemo, pa bo moder z mrvicami in pomarančo.

Računalniško ozadje

Tudi to je ena od nalog, v kateri sledimo izvajanju programa.

Kocke

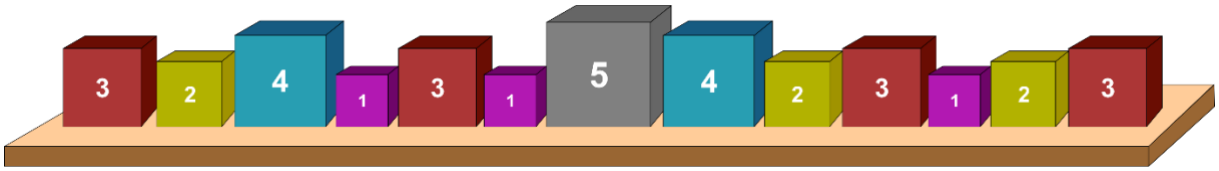
4. in 5. razred

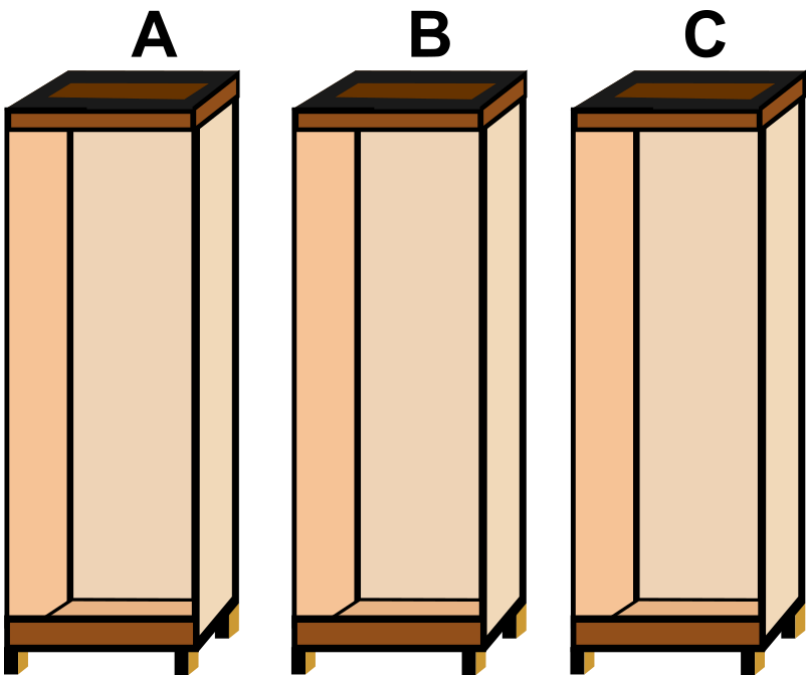
Bober ima kocke, na katerih je napisana njihova teža. Zložiti jih mora v tri omare. Pri tem:

- na nobeno kocko ne postavi težje kocke, saj bi se lahko spodnja zlomila;
- v vsako omaro naloži največ 15 kg.

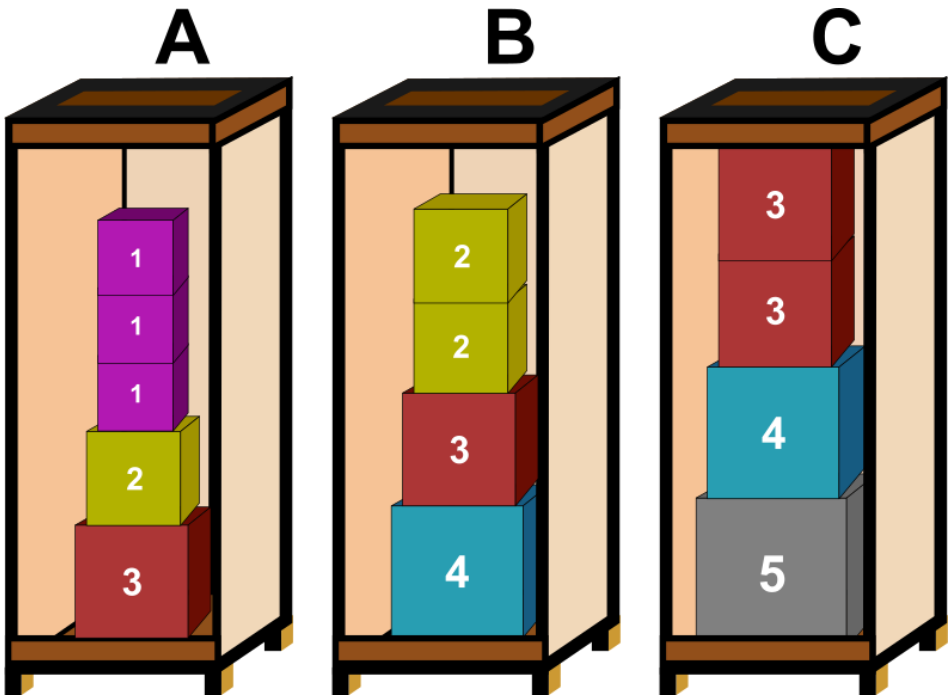
Kocke jemlje s police po vrsti od leve proti desni. Za vsako kocko preveri, ali jo sme odložiti v A; če ne gre, preveri B in na koncu C.

Pomagaj pospraviti kocke. V omare nariši, kako bodo kocke pospravljene.





Rešitev



Z zlaganjem začnemo s prvo kocko, težko 3 kg, in jo postavimo v omaro A. Sledi kocka, ki je težka 2 kg. Ker ni težja od prejšnje, jo postavimo v omaro A na prvo kocko. 4 kg kocka, ki sledi, ne more iti v omaro A, saj je pretežka, zato jo pospravimo v omaro B. Naslednja kocka tehta 1 kg, pospravimo jo lahko v prvo omaro. Tako nadaljujemo, dokler niso pospravljene vse kocke.

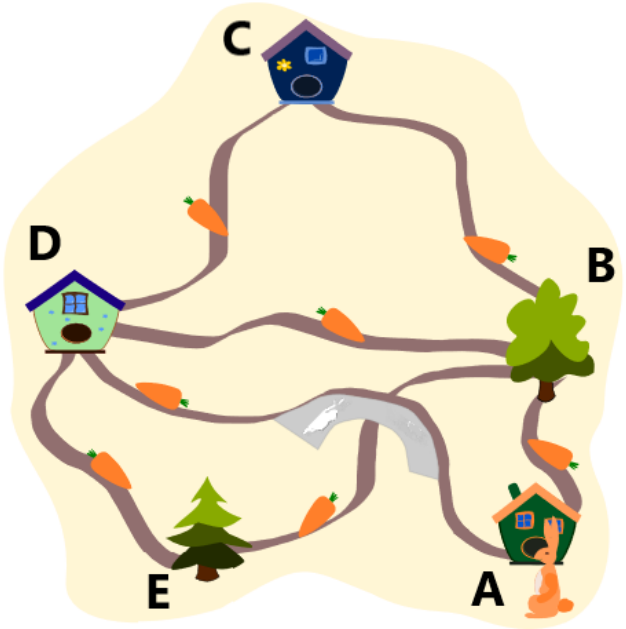
Računalniško ozadje

V nalogi izvajamo preprost program.

Pobiranje korenja

4. in 5. razred

Mali zajček živi v hiši A. Odločil se je, da bo pobral vse korenje. Sprehod po vsaki poti mu vzame 1 minuto.



V kakšnem vrstnem redu naj obišče hiše in drevesa, da čimprej pobere vse korenje in se vrne domov?

A) A B C D E B D A
B) A D E B A D C B A
C) A B D E D C B A
D) A D E B C D A

Rešitev

Pravilni odgovor je odgovor A, saj se zajček sprehodi po vseh poteh natanko enkrat in pri tem pobere vse korenje. Ker se zajček po vsaki poti sprehodi le enkrat, je ta pot najhitrejša.

Pot B je daljša od poti A, saj vsebuje 1 dodatno pot.

Pri poti C se zajček ne sprehodi po poteh AD in BE, zato ne pobere vsega korenja. Poleg tega se dvakrat sprehodi po poteh AB in DE ter s tem podaljša pot.

V primeru D pa se zajček ne sprehodi po poteh DB in AB, zato ostaneta 2 korenčka nepobrana.

Računalniško ozadje

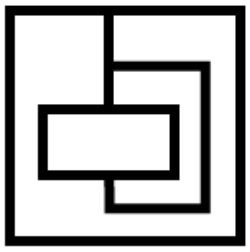
Iskanje najboljše poti ali obhoda glede na neke kriterije je pogosta naloga računalniških programov – in tudi udeležencev tekmovanja Bober.

Poplavljanje

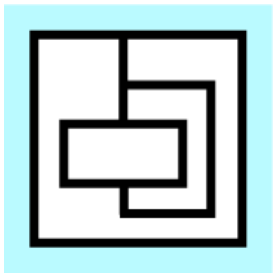
4. in 5. razred

Kadar voda poplavi grad, to poteka tako:

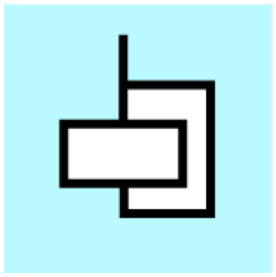
- Voda najprej poplavi zunanjost gradu.
- Po eni uri se vsak zid z vodo na eni strani pod pritiskom poruši.
- Voda poplavi novo območje, ki ni več omejeno s stoječimi zidovi.
- Vsi trenutno stoječi zidovi z vodo na eni strani se po drugi uri porušijo in voda dodatno poplavi območje gradu.



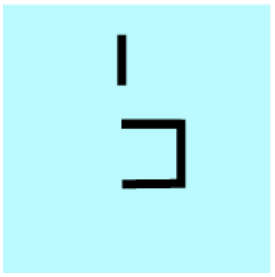
Ta postopek se ponavlja, dokler voda ne zalije celotnega območja.



Na začetku



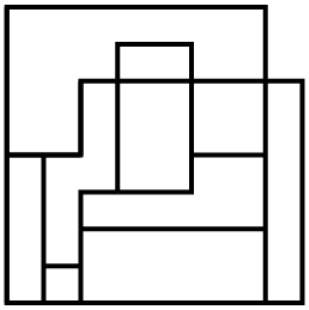
Po eni uri



Po dveh urah

Celoten grad je poplavilo v 2 urah.

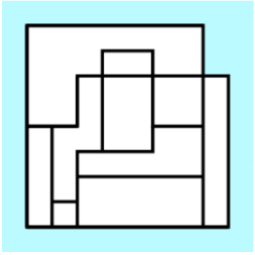
V kolikšnem času bi poplavilo celoten grad na desni sliki?



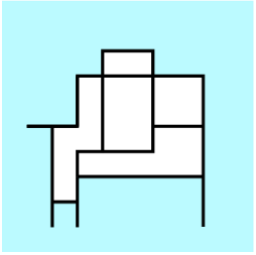
Rešitev

Grad bi v celoti poplavilo v 3 urah. Postopek bi lahko opisali z naslednjimi slikami:

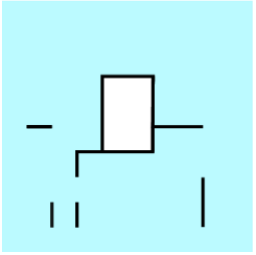
Na začetku:



Po eni uri:



Po dveh urah:



Po treh urah:

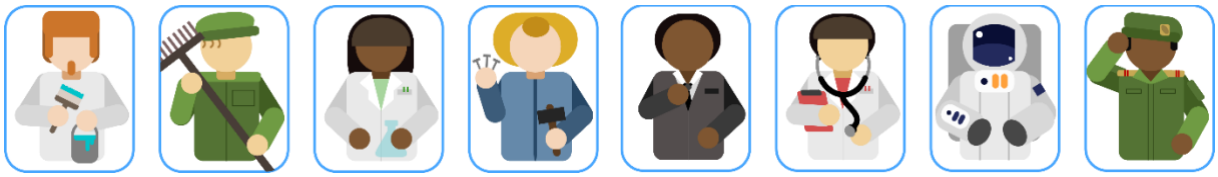
Računalniško ozadje

"Poplavljanje" je pogost postopek, povezan z algoritmom iskanja v širino ali, tudi, iskanja najkrajših poti.

Ugani kdo

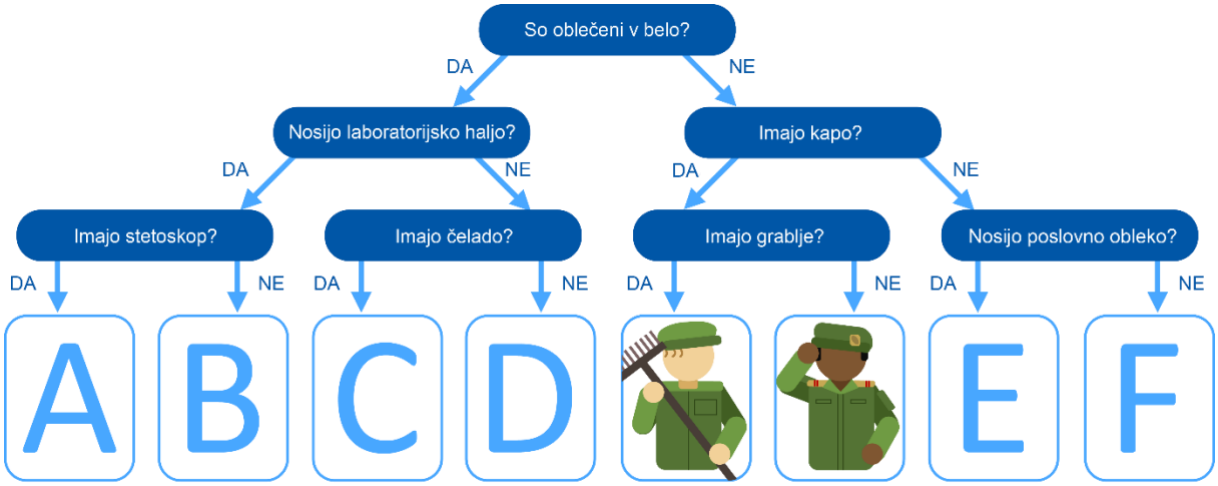
4. in 5. razred

Jaka in Nika imata 8 kart, na katerih so predstavljeni različni poklici (od leve proti desni): slikar, vrtnar, znanstvenik, mizar, poslovnež, zdravnik, astronavt, vojak.



Jaka izbere eno od kart, Nika pa mora s postavljanjem različnih vprašanj ugotoviti poklic na izbrani karti. Nika lahko zastavi le eno vprašanje naenkrat in Jaka lahko na vprašanje odgovori le z DA ali NE.

Nika poskuša ugotoviti izbrano karto s pomočjo spodnjega diagrama.

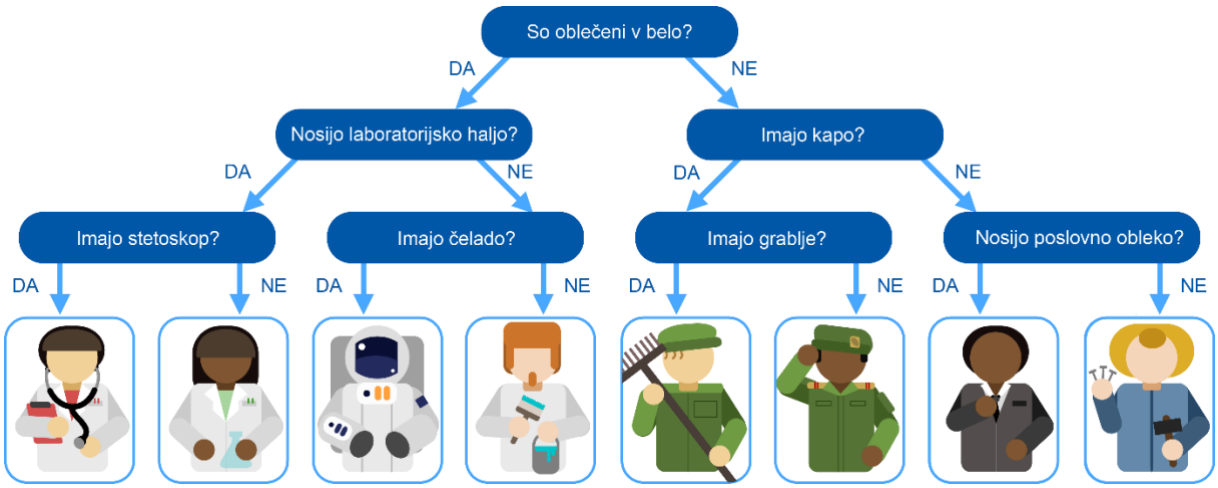


Kateri poklici so na mestih A, B, C, D, E in F?

	A	B	C	D	E	F
A)	astronavt	slikar	zdravnik	znanstvenik	vrtnar	mizar
B)	zdravnik	znanstvenik	vrtnar	vojak	poslovnež	mizar
C)	zdravnik	znanstvenik	astronavt	slikar	poslovnež	mizar
D)	slikar	zdravnik	astronavt	znanstvenik	vrtnar	vojak

Rešitev

Pravilni odgovor je: zdravnik, znanstvenik, astronavt, slikar, poslovnež, mizar.



Nika lahko po treh vprašanjih s pomočjo diagrama ugotovi, kateri poklic se nahaja na izbrani karti.

- A: oseba, ki je oblečena v belo, nosi laboratorijsko haljo in ima stetoskop, je zdravnik.
- B: oseba, ki je oblečena v belo, nosi laboratorijsko haljo in nima stetoskopa, je znanstvenik.
- C: oseba, ki je oblečena v belo, ne nosi laboratorijske halje in ima čelado, je astronavt.
- D: oseba, ki je oblečena v belo, ne nosi laboratorijske halje in nima čelade, je slikar.
- E: oseba, ki ni oblečena v belo, nima kape in nosi poslovno obleko, je poslovnež.
- F: oseba, ki ni oblečena v belo, nima kape in ne nosi poslovne obleke, je mizar.

Računalniško ozadje

V umetni inteligenci takšnemu diagramu rečemo klasifikacijsko drevo. Sestavljanje klasifikacijskega drevesa na podlagi podatkov je eden najstarejših algoritmov strojnega učenja.

Chez Connie

4. in 5. razred

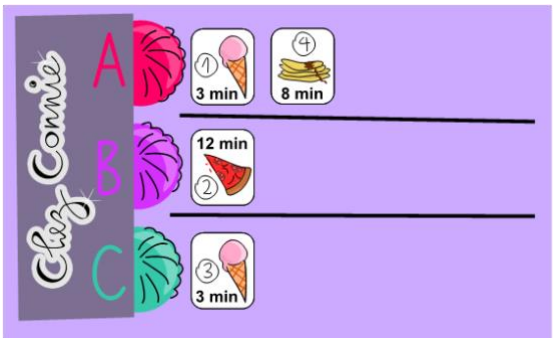
Pravkar se bo odprla restavracija Chez Connie in bobri čakajo, da naročijo eno od treh poslastic:

- sladoled, pripravljen v 3 minutah;
- palačinke, pripravljene v 8 minutah;
- pico, pripravljeno v 12 minutah.

Connie želi organizirati naročila, da bodo gostje hitro postreženi. Naročila zapiše na oštevilčene lističe. Prvo današnje naročilo izgleda takole:

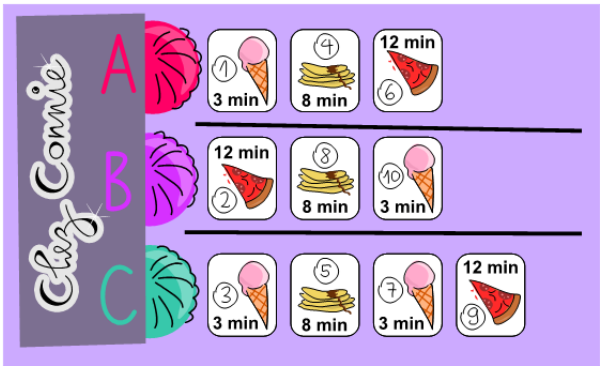
Connie deli lističe okencem A, B in C tako, da naročilo vedno dodeli k prvemu prostemu okencu. Če bosta istočasno prosti dve okenci, dobi naročilo tisto, ki je prej po abecednem redu.

Prva štiri naročila je že dodelila. Razporedi naslednjih 6:



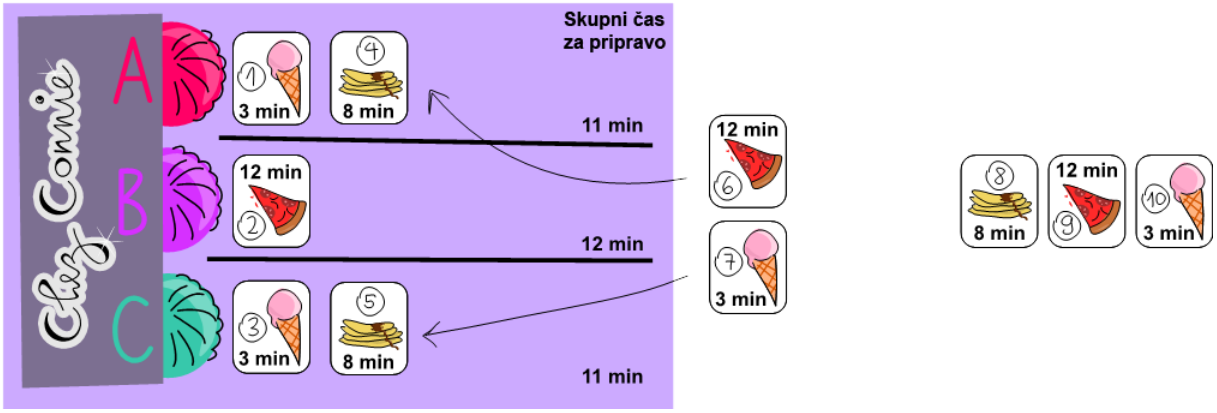
Rešitev

Pravilna rešitev je na sliki.



Za razdelitev naročil po okencih mora pri vsakem okencu upoštevati skupni čas pred tem dodeljenih naročil. Po 3 minutah se sprostita okenci A in C. Četrto naročilo je dodelila okencu A, saj je to po abecedi pred C. Peto naročilo dodeli k okencu C.

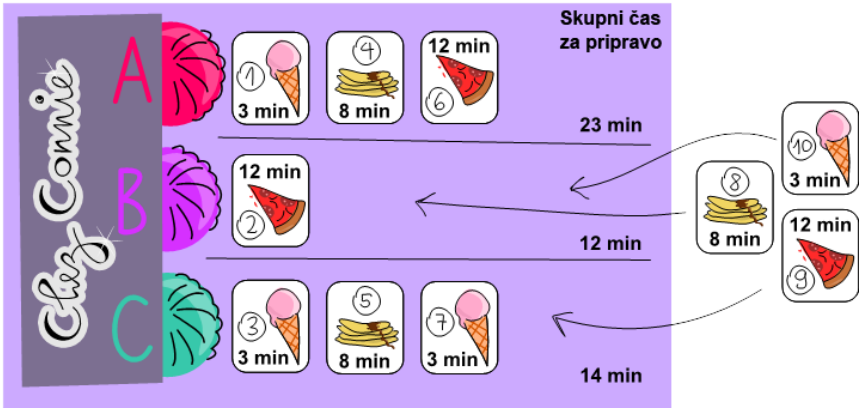
Po 11 minutah sta ponovno prosti okenci A in C, zato k okencu A razporedi naročilo 6 in k okencu C naročilo 7.



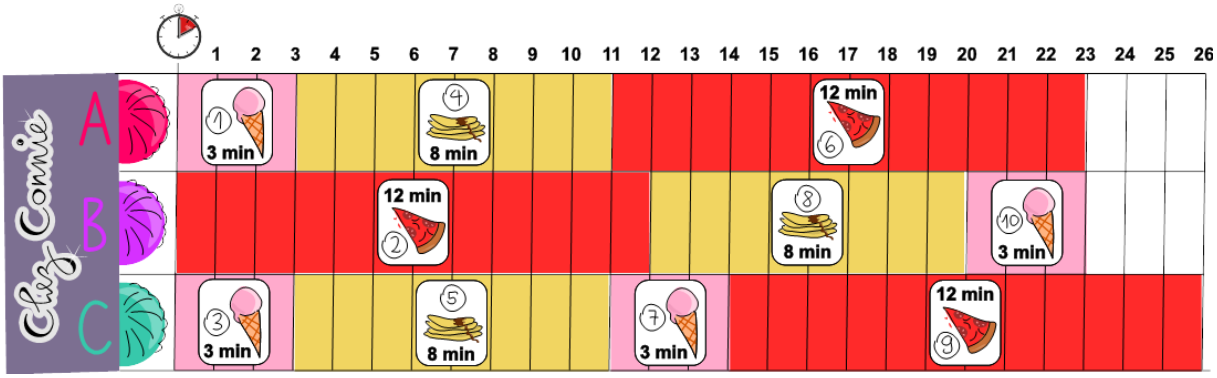
Po 12 minutah se sprosti okence B in Connie mu dodeli naročilo 8.

Po 14 minutah je naročilo 9 razporejeno k okencu C.

Po 20 minutah je naročilo 10 razporejeno k okencu B.



Za boljši pregled lahko Connie uporabi tudi tabelo, ki po minutah prikazuje zasedenost posameznega okenca:



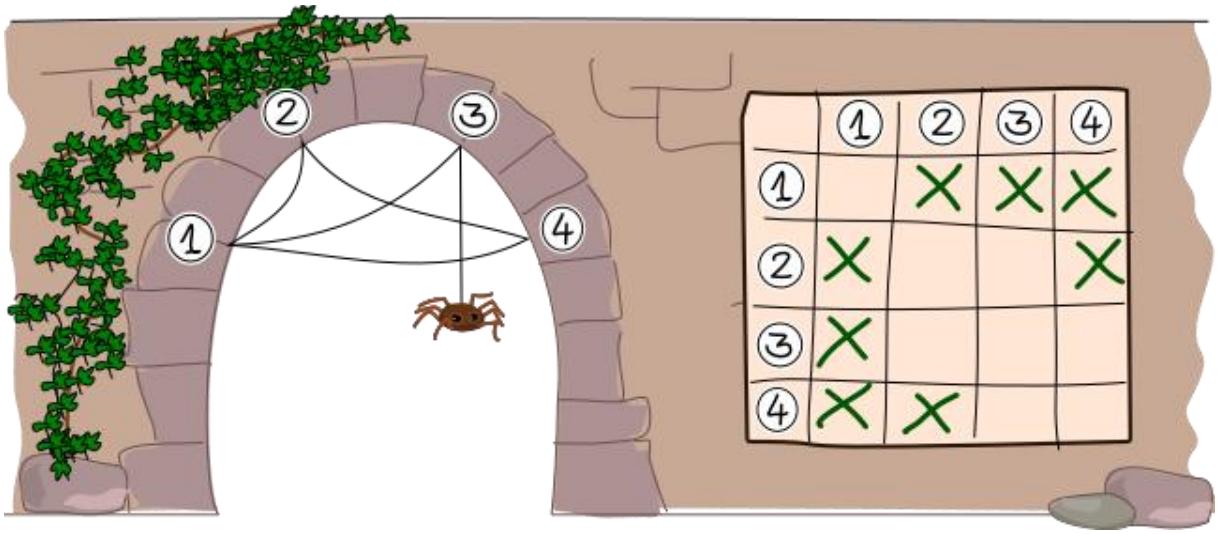
Računalniško ozadje

Naloga se ukvarja s problemom dodeljevanja virov (scheduling). Procesorji sodobnih računalnikov imajo več jeder in zmorejo hkrati opravljati več opravil. Operacijski sistem jim zato dodeljuje delo.

Vzorec pajkove mreže 1

4. in 5. razred

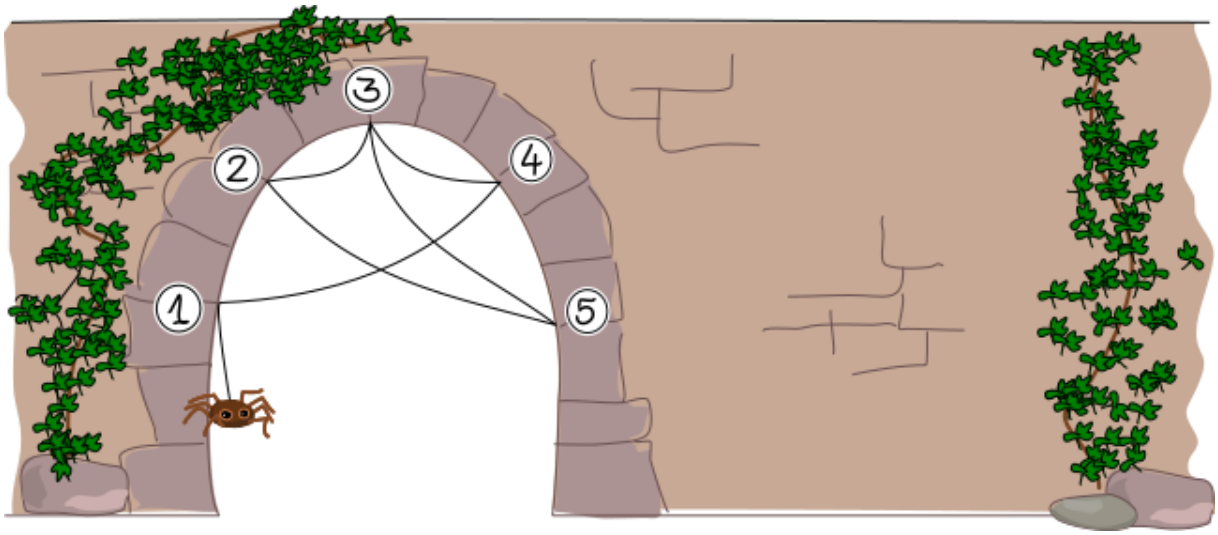
Pajek želi zgraditi čim več različnih pajčevin, zato je iznašel način za izdelavo načrta svoje pajčevine.



Načrt izdela tako, da oštevilči končne točke pajčevine od 1 do **N** in uporabi polja v mreži v skladu z naslednjimi pravili:

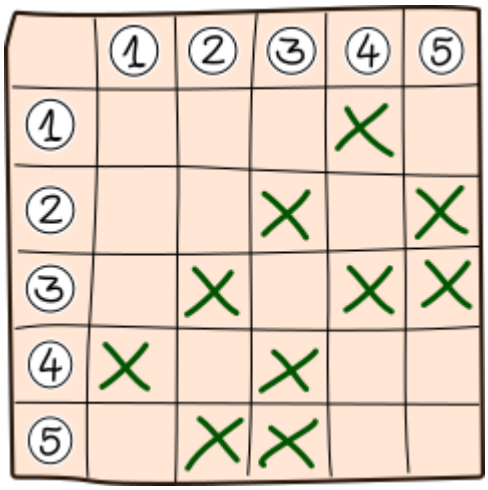
- Če obstaja nit, ki povezuje končno točko A s končno točko B, je polje v stolpcu A in vrstici B označeno z »X«.
- Nit, ki povezuje končno točko A s končno točko B, povezuje tudi končno točko B s končno točko A.

Pajek je zdaj zgradil to pajčevino:

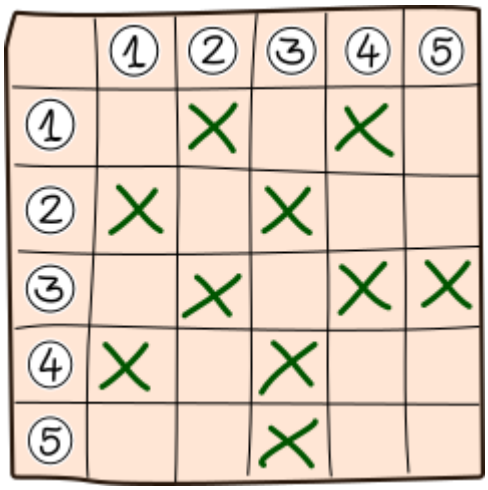


Kako izgleda načrt te pajčevine?

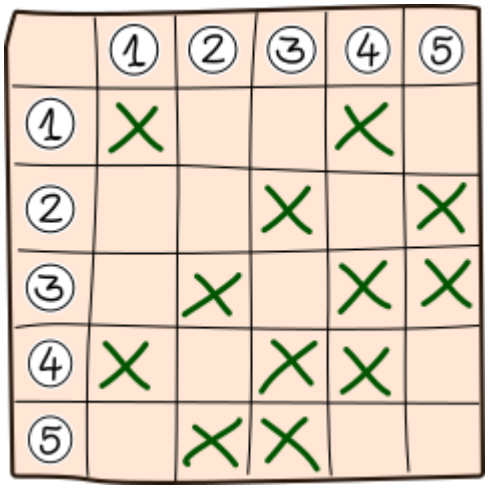
A)



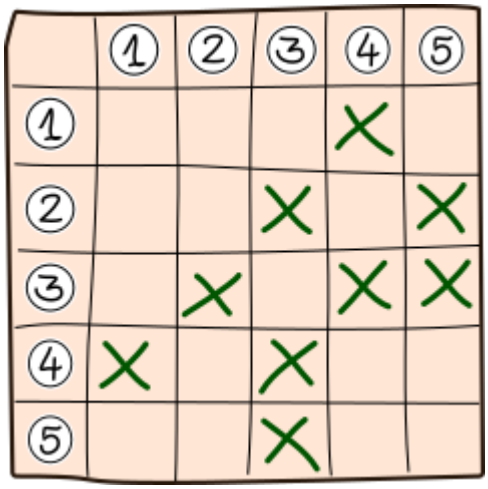
C)



C)



D)



Rešitev

Pravilni odgovor je A.

Odgovor B ni pravilen, saj končna točka 3 ni povezana s končno točko 5.

Odgovor C ni pravilen, saj končni točki 1 in 4 nista povezani sami s seboj.

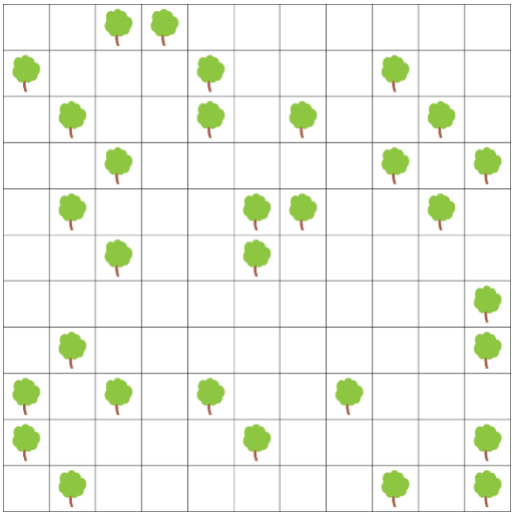
Odgovor D ni pravilen, saj je končna točka 2 povezana s končno točko 5 in obratno končna točka 5 s končno točko 2, kar pa označeno v mreži.

Računalniško ozadje

Tabelice v nalogi so ena od možnih predstavitev "grafa". Graf je množica objektov, med katerimi so nekatere povezani – na primer sošolcev, pri čemer povezave predstavljajo pare prijateljev.

Marko skače

4. do 9. razred

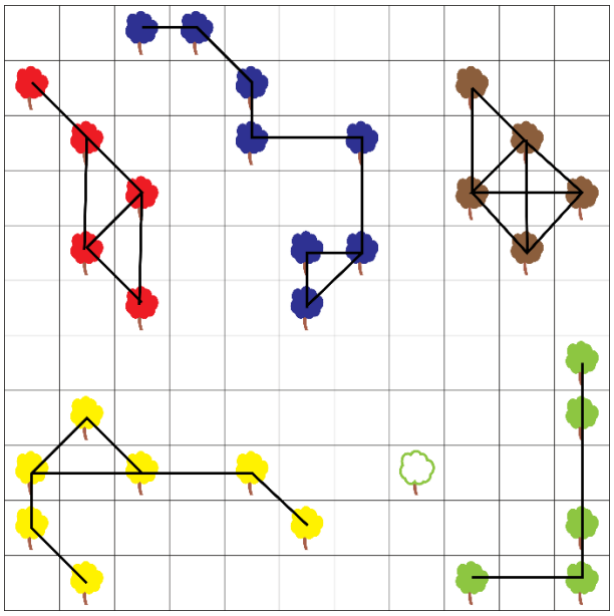


Marko je majhna opica, ki živi v parku in ne more skakati zelo daleč: skoči lahko največ dve polji daleč v vodoravni ali navpični smer, ali pa na diagonalno polje (glej sliko).

Marko vsako jutro spleza na eno izmed dreves in nato ves dan skače po drevesih, do katerih lahko pride, ne da bi vmes stopil na tla. Če si dobro izbere začetno drevo, bo lahko obiskal veliko dreves; če slabo, manj.

Slika parka je desno. Kakšno je največje število dreves, ki jih lahko obišče v enem dnevu?

Rešitev



Marko lahko v enem dnevu obišče skupine dreves, ki so predstavljene z različnimi barvami.

Največja povezana skupina dreves obsega 8 dreves, to so modro obarvana drevesa na sredini slike zgoraj. Vse skupine dreves so med seboj oddaljene več kot dve polji, zato Marko med skupinami ne more skakati.

Kako rešimo takšno nalogo? Izberemo si drevo in ga pobarvamo. Z enako barvo pobarvamo vsa drevesa, na katera je mogoče skočiti s tega drevesa. In vsa, na katera je mogoče skočiti z dreves, ki smo jih pravkar pobarvali. In vsa, na katera je močno skočiti s teh... Ko ne gre več, vzamemo drugo barvico, izberemo eno od dreves, ki še niso pobarvana, in ponovimo postopek.

Računalniško ozadje

V opisu ozadja prejšnje naloge smo omenili grafe. Tudi v tej nalogi vidimo graf: drevesa so točke grafa in dve drevesi sta povezani, če je možno skočiti z enega na drugega. Če posamični deli grafa niso povezani med seboj, rečemo, da je graf sestavljen iz več nepovezanih komponent.

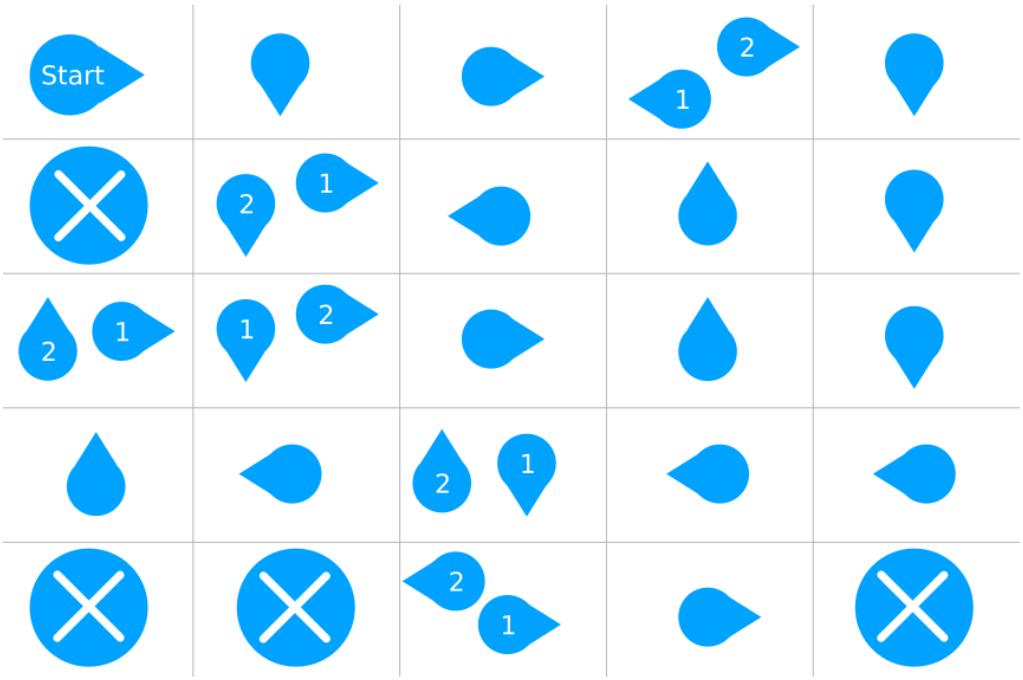
Iskanje zaklada

5. razred

Tvoji prijatelji pirati so našli zemljevid, vendar ne vedo točno, kaj pomeni. Vse, kar vidijo, so čudne oblike. Ko si skrbno preučil zemljevid, si na zadnji strani našel ta navodila:

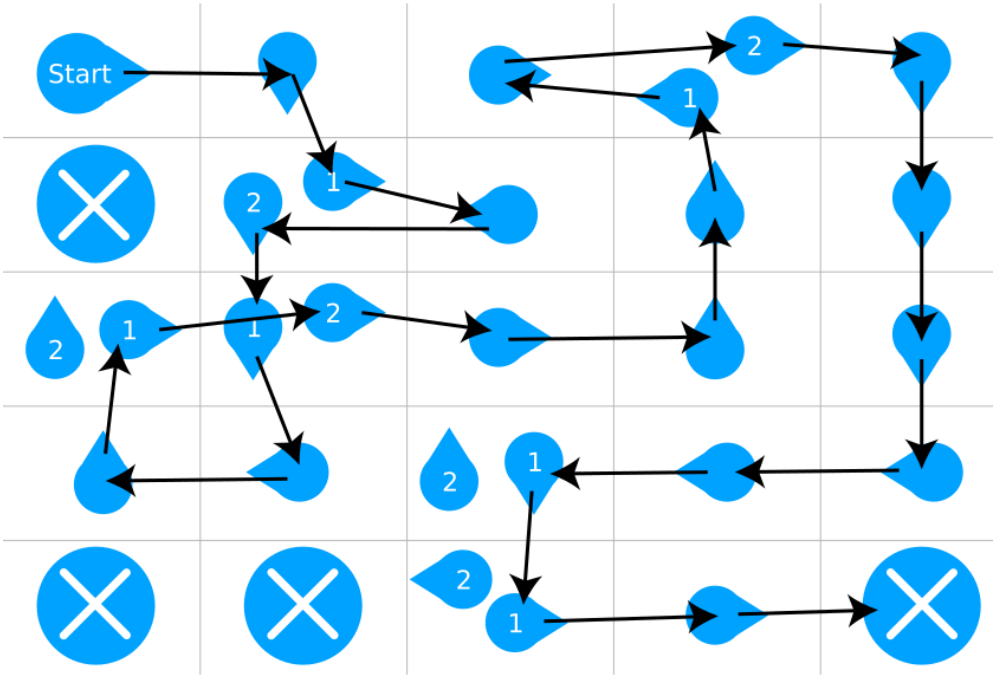
Simbol				
Pomen	Začni tukaj in sledi puščici do naslednjega polja.	Nadaljuj na naslednje polje v smeri puščice.	Ko to polje prečkaš prvič, sledi tej puščici.	Ko to polje prečkaš drugič, sledi tej puščici.

Na katerem polju z oznako  se nahaja zaklad?



Rešitev

Zaklad se nahaja v desnem spodnjem polju. Spodaj je narisana pot do rešitve. Na poljih, kjer imamo po dve kapljici, prvič sledimo kapljici s številko 1, drugič pa kapljici s številko 2.



Računalniško ozadje

Pri nalogi gre za sledenje programu, ki ima poleg tega še določena notranja stanja, ki si jih moramo zapomniti.

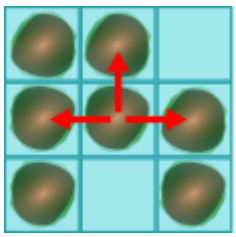
Kengurujki

5. razred

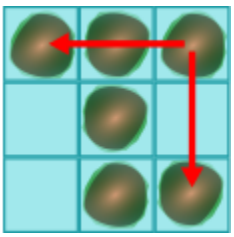
Kengurujka Maja skače po otočkih do prijateljice Klare na drugi stran močvirja. Skače lahko

- s kratkim skokom na sosednji otoček
- z dolgim skokom, tako da preskoči katerokoli sosednje polje, če pri tem pristane na otočku.

Primer kratkega skoka

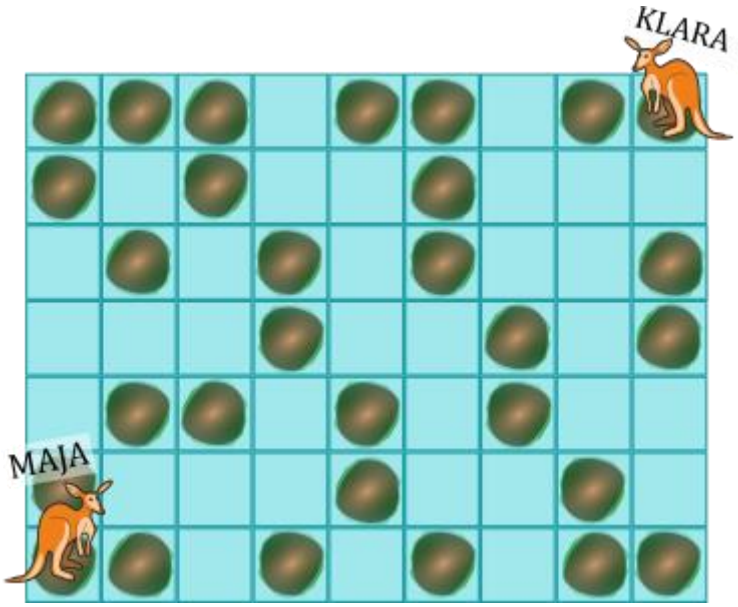


Primer dolgega skoka

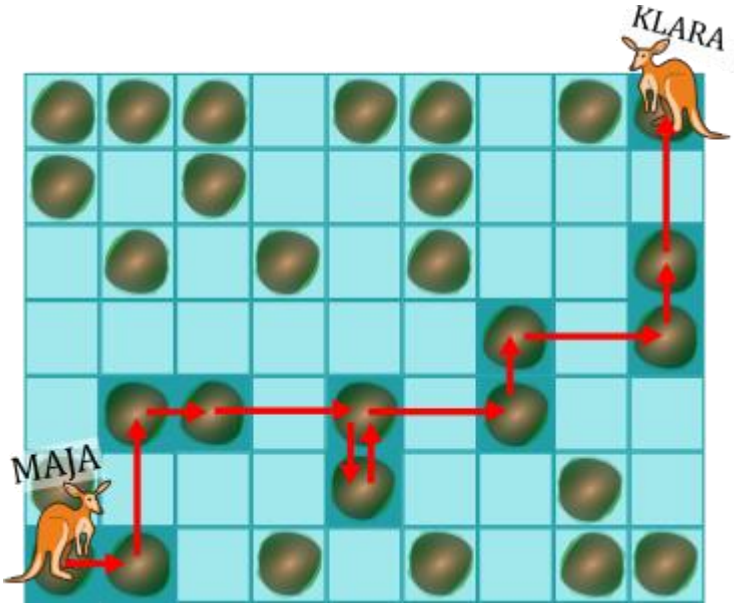


Diagonalno Maja ne more skakati. Ker se z dolgim skokom bolj utrudi in so bolj nevarni, ne more dvakrat zapored narediti dolgega skoka.

Vriši najkrajšo pot, po kateri bo Maja prišla do Klare.



Rešitev



Majino pot do Klare kaže slika. Ostale možne rešitve vključujejo ponovni skok na otoček, na katerem je že bila, in so zato daljše.

Maja lahko najprej skoči bodisi gor ali desno. Če skoči gor, lahko skoči samo nazaj na izhodišče, zato to ni ustrezna pot. Maja zato najprej skoči na desni otoček (prva puščica).

S tega otočka lahko skoči bodisi dolg skok gor ali dolg skok desno. Če izbere dolg skok desno, obstane na tem otočku, saj z njega ne more narediti nobenega kratkega skoka, zato gre Maja gor (druga puščica).

Ker je naredila dolg skok, mora bit njen naslednji skok kratek. Edino sosednje polje z otočkom je desno, zato skoči tja (3. puščica).

S tega otočka Maja lahko skoči le dolg skok desno (4. puščica).

Ker mora biti naslednji skok kratek, Maja skoči na otoček pod tem (5. puščica) in nazaj (6. puščica). To sta edini možnosti, saj drugega od tu dosegljivega otočka ni.

Ker spet lahko naredi dolg skok, skoči Maja desno (7. puščica).

Od tu lahko skoči samo kratek skok gor (8. puščica).

Ker lahko naredi dolg skok, skoči Maja desno (9. puščica). Edina druga možnost bi bila vračanje nazaj, kar pa ni smiselno.

Maja mora narediti kratek skok, zato lahko skoči le gor (10. puščica).

Ker lahko naredi dolg skok skoči gor h Klari (11. puščica).

Računalniško ozadje

Iskanje najkrajše poti – ali najboljše poti po kakem drugem kriteriju – je ena pogostih nalog računalnikov. Pri tem pogosto ne gre za dejansko pot med nekimi kraji, temveč za, na primer, optimalno zaporedje nekih operacij.

Pikčaste kukavice

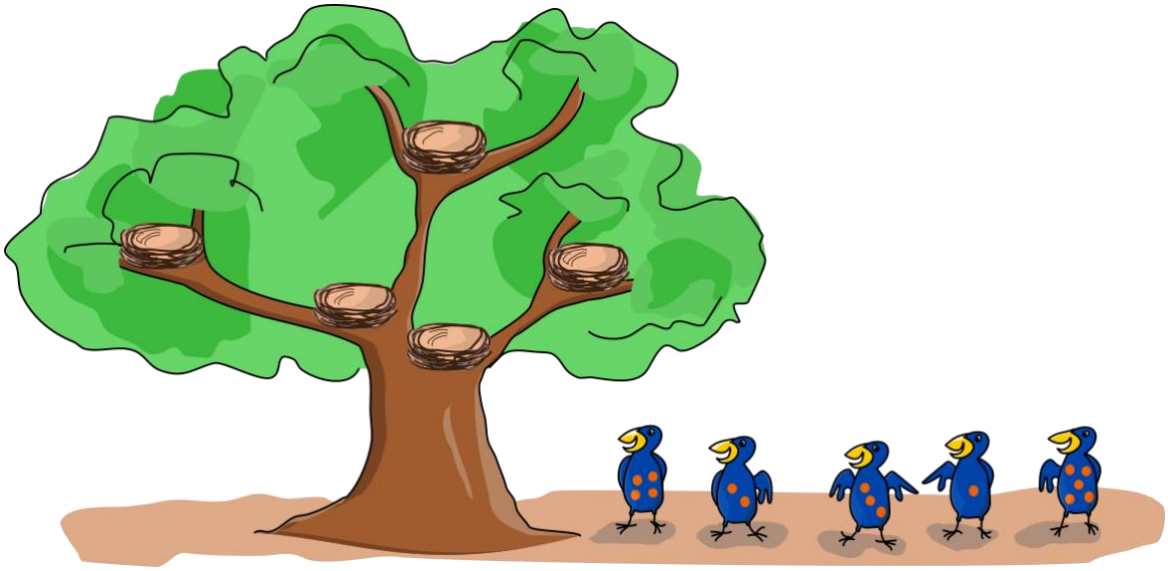
5. razred

Ob drevesu je pet pikčastih kukavic. Druga za drugo – po vrsti od leve proti desni – plezajo na drevo in se naselijo v prazna gnezda. Tista s štirimi pikami je prva. Vsaka pikčasta kukavica si izbere gnezdo po naslednjem pravilu:

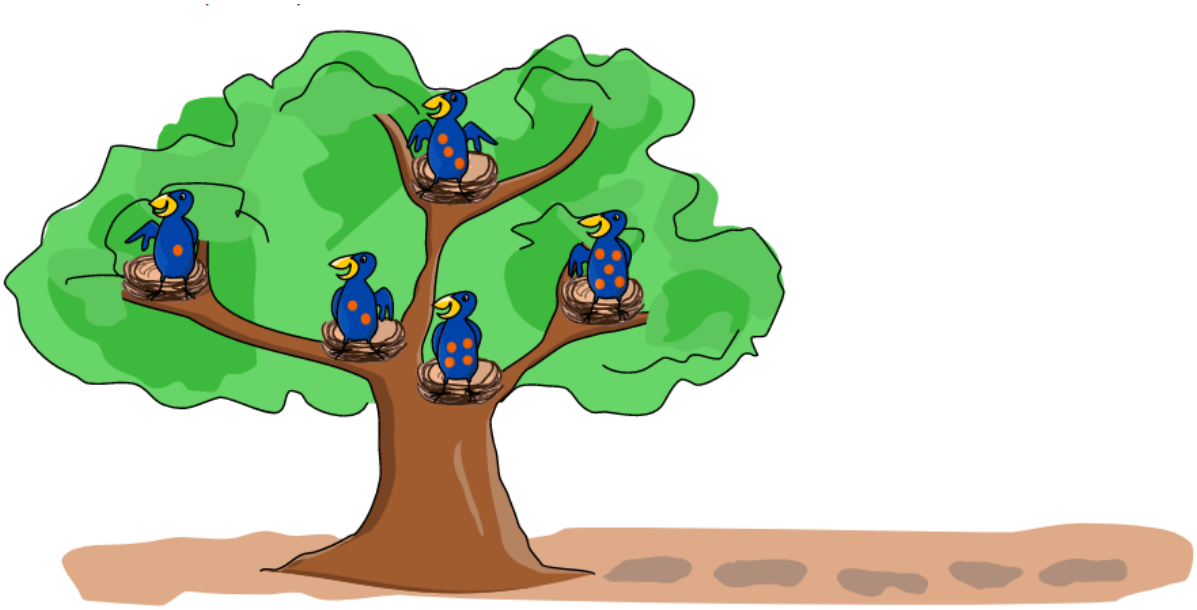
Začne na dnu drevesa in ponavlja naslednje korake, dokler ne najde praznega gnezda:

1. Vzpenja se po drevesu, dokler ne pride do gnezda.
2. Če je gnezdo prazno, se naseli vanj in tam ostane.
3. Če je gnezdo zasedeno, pogleda pikčasto kukavico, ki sedi v njem. Če ima ta:
    - več pik, nadaljuje na levo stran drevesa,
    - manj ali enako pik, nadaljuje na desno stran drevesa.

Kam se bo naselila katera pikčasta kukavica? Poveži vsako kukavico z njenim novim gnezdom.



Rešitev



Prva kukavica, ki ima 4 pike, se bo naselila v prvo gnezdo. Nato prileti kukavica z dvema pikama. Ko pride do prvega gnezda, tam že sedi kukavica s 4 pikami. Ker jih ima torej več, druga kukavica odide na levo. Tam pride do prvega gnezda, ki je prazno, zato se naseli vanj. Tretja prileti kukavica s tremi pikami. Ker je 3 manj kot 4, gre po levi strani drevesa do gnezda, v katerem je že naseljena kukavica z dvema pikama. Ker je to manj pik, kot jih ima sama, nadaljuje na desno stran drevesa. Ko pride do gnezda, je to prazno, zato se naseli vanj. Za njo prileti kukavica z eno piko. Ker ima najmanj pik, bo ob vsakem zasedenem gnezdu nadaljevala na levo, dokler ne pride do prostega gnezda. Zadnja pride še kukavica s petimi pikami. Ob prvem zasedenem gnezdu nadaljuje na desno, saj ima prva kukavica manj pik kot ona.

Računalniško ozadje

V nalogi smo simulirali delovanje urejenih iskalnih dreves. To je eden od načinov, na katerega si računalniki organizirajo podatke glede na določen kriterij (kot je tu število pik), da jih kasneje hitreje najdejo.

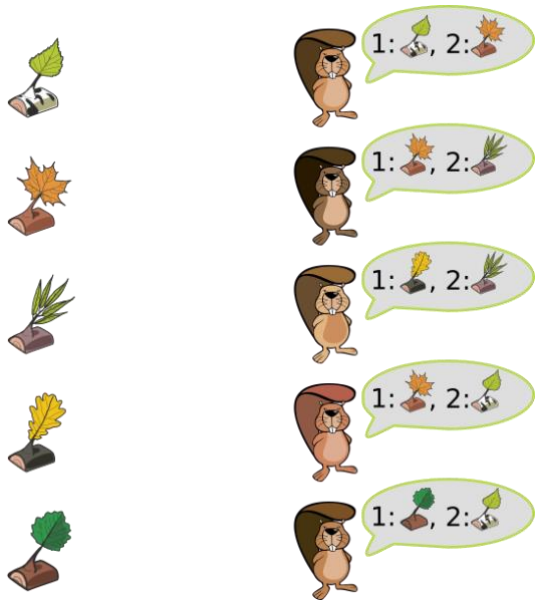
Želje 2

5. razred

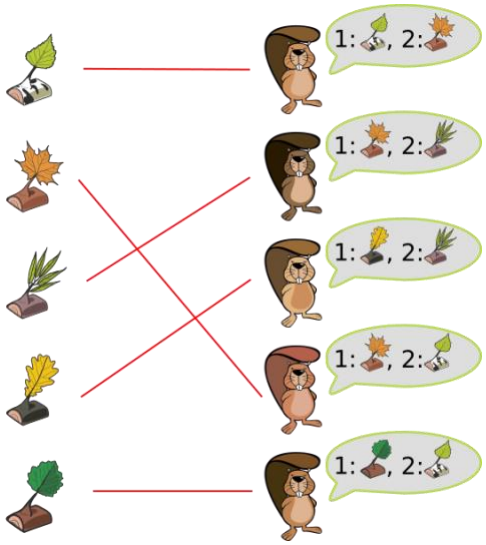
Družina bobrov ima pet daril za svojih pet otrok. Vsak bober najprej pove, katero darilo si najbolj želi in katero je njegovo drugo najljubše.
Darila morajo biti razdeljena tako, da:

- Kar največ bobrov dobi svoje najljubše darilo.
- Ostali bobri dobijo svoje drugo najljubše darilo.

Poveži bobre z darili, ki jih bodo dobili.



Rešitev



Ker si nekateri bobri najbolj želijo enakih daril, vsem te želje ne moremo izpolniti.

Peti bober si najbolj želi peto darilo, ki si ga ne želi nihče drug, zato ga tudi dobi.

Tretji bober si najbolj želi četrto darilo, ki si ga ne želi nihče drug, zato ga dobi.

Drugi in četrti bober si oba najbolj želita drugo darilo, vendar si ga ne moreta deliti. Drugi bober si poleg drugega darila želi tudi tretjega. Četrti bober si poleg drugega darila želi še prvega. Ker si tretje darilo želi samo drugi bober (izmed bobrov, ki so še brez darila), bo drugi bober dobil tretje darilo. Preostali darili, prvo in drugo, pa bosta dobila prvi in četrti bober, ki si jih tudi najbolj želita.

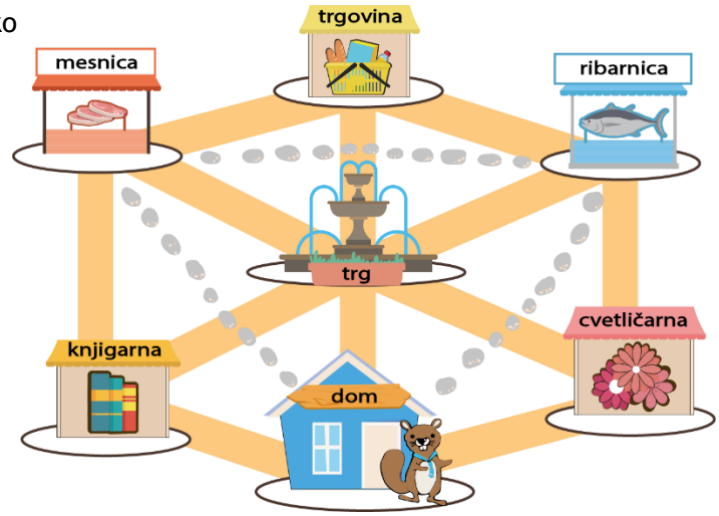
Računalniško ozadje

Gre za problem podoben Problemu stabilnih zakonov. Podobne naloge se pojavljajo še marsikje. Leta 2012 je bila Nagrada Švedske banke za ekonomske vede v spomin Alfreda Nobela (ki ji pravijo tudi Nobelova nagrada za ekonomijo) podeljena raziskovalcem, ki so uporabili podoben postopek za resnične probleme s področja ekonomije.

Nakupovanje

6. in 7. razred

Na sliki je zemljevid mesta. Bobri se lahko med stavbami sprehajajo po blatni poti ( ) ali kamniti poti ( ). Za sprehod od ene stavbe do druge po blatni poti potrebujejo 5 min in po kamniti 8 min.



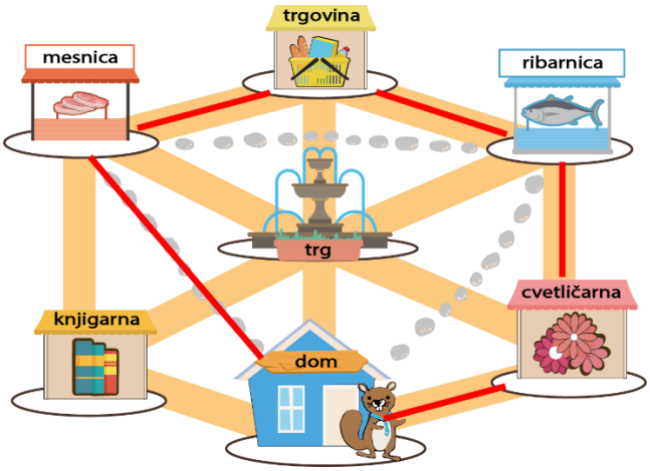
Bobrček tako potrebuje 5 min od trga do cvetličarne ali knjigarne in 8 minut od ribarnice do mesnice.

Mama je bobrčka prosila za pomoč pri nakupu. Kupiti mora:

- ribe (ribarnica)
- piščanca (mesnica)
- jajca (trgovina)
- rože (cvetličarna)

Najmanj koliko časa potrebuje Bobrček za celoten nakup, če svoje nakupovanje začne in zaključi doma?

Rešitev



Pravilni odgovor je 28 minut.

Bobrček se bo sprehodil po vsaj petih poteh, saj mora svojo pot začeti in končati doma ter vmes obiskati štiri trgovine. Vsaka pot traja najmanj 5 min, kar pomeni, da bo za celoten nakup potreboval vsaj $5 \cdot 5 = 25$ min.

Bobrčku bo obisk mesnice ali ribarnice vzel 10 min (preko knjigarne ali cvetličarne) ali 8 min (neposredno). Ker pa mora Bobrček nakup opraviti v trgovini in cvetličarni, je najhitrejši način, da gre v ribarnico na poti iz cvetličarne do trgovine. Ravno tako je bolje, če Bobrček izbere kamnito pot, ki povezuje mesnico in dom, kot pa da se iz mesnice odpravi domov preko knjigarne. Zato je smiselno, da Bobrček mesnico obišče prvo ali zadnjo.

Ker čas hoje med prodajalnami, ki jih mora obiskati, traja $5 \cdot 4 = 20$ min, je skupni čas hoje $8 + 20 = 28$ min. Izbrana pot je prikazana na spodnji sliki.

Računalniško ozadje

Kot pri več drugih nalogah gre tudi tu za neko različico iskanja optimalne poti.

Restavracija Fifo

6. in 7. razred

Zaradi pandemije prihajajo gostje v restavracijo po hrano za s seboj. Postrežba vsakega gosta traja 3 minute. Medtem ko strežejo enega, ostali čakajo v vrsti. Gostje niso v vrsto razvrščeni zgolj po tem, kdaj so prišli, ampak tudi po njihovi starosti. Najstarejši v čakalni vrsti je vedno naslednji postrežen. Preden postrežejo naslednjega gosta v vrsti, najprej končajo s trenutno postrežbo.

Trenutno je restavracija prazna. Bob (15 let) in Alex (40 let) prideta istočasno. Dve minuti kasneje prideta še Ben (70 let) in Dina (10 let). To so zadnji gostje danes.

V kakšnem vrstnem redu bodo postreženi, če upoštevamo pravila postrežbe v tej restavraciji?

Rešitev

Pravilna rešitev je Alex, Ben, Bob in Dina.

Ko prideta v restavracijo Alex in Bob, najprej postrežejo Alexa, Bob pa čaka v vrsti.

Po 2 minutah vstopita Ben in Dina. Boba že strežejo, zato se Ben in Dina postavita v vrsto, in sicer je Ben prvi v vrsti, saj je najstarejši, za njim je Bob in kot zadnja Dina.

Odgovor A ni pravilen, saj je upoštevano pravilo »kdor prej pride, je prej postrežen«, ne upošteva pa starosti.

Odgovor B ni pravilen, saj upošteva pravilo »najmlajši je najprej postrežen«, kar ni pravilo v tej restavraciji. Poleg tega ne upošteva časa prihoda.

Odgovor D ni pravilen, saj ne upošteva časa prihoda.

Bodite pozorni na to, da bi se vrstni red postrežbe lahko spremenil, če bi medtem v restavracijo vstopil še kdo drug.

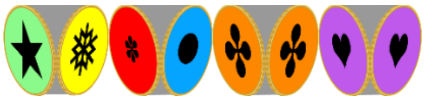
Računalniško ozadje

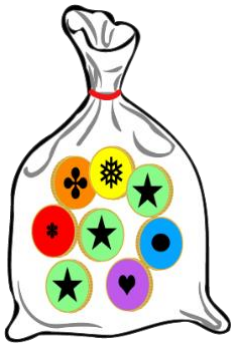
Operacijski sistem se mora odločati, kako razdeljevati vire (na primer procesorski čas, dostop do diska, omrežja...) različnim programom. Pri tem se mora odločati, kateremu programu dati prednost, da bo delovanje čimbolj gladko.

Vreča kovancev

6. in 7. razred

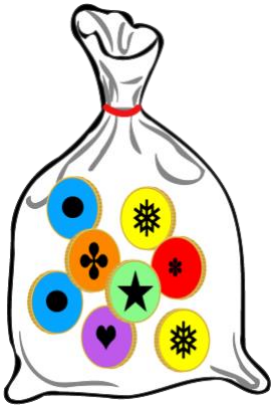
Bobri uporabljajo štiri vrste kovancev, ki so prikazani na spodnji sliki (obe strani posameznih kovancev):



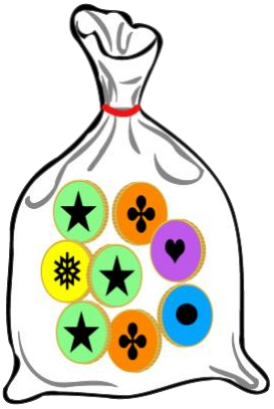


Bober Patrik je v vrečo naložil več kovancev (slika na desni) in jo nesel v banko. Na poti so se kovanci v vreči pošteno pretresli. Vrečo je postavil zraven treh drugih vreč s kovanci.

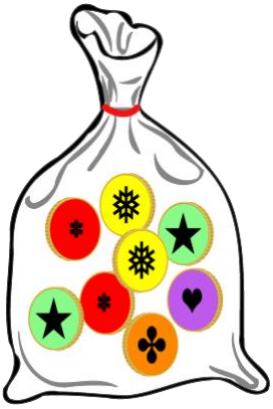
Katera vreča je Patrikova?



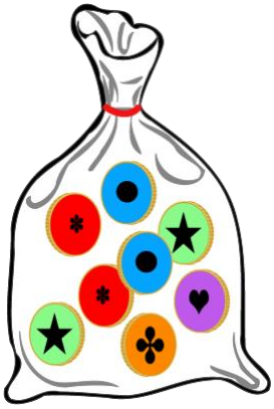
A)



B)



C)

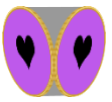


D)

Rešitev

Pravilni odgovor je C.

Patrikova vreča ima 4 kovance tipa 1 (zelen/rumen), 2 kovanca tipa 2 (rdeč/moder), 1 kovanec tipa 3 (oranžen) in 1 kovanec tipa 4 (vijola), kot to prikazuje spodnja tabela. V tabelo smo zapisali tudi tipe kovancev v vrečah A, B, C in D.

				
Patrikova vreča	4	2	1	1
Vreča A	3	3	1	1
Vreča B	4	1	2	1
Vreča C	**4**	**2**	**1**	**1**
Vreča D	2	4	1	1

Odgovor A ni pravilen, saj ima 3 zeleno-rumene kovance, kar ni dovolj, ker so v Patrikovi vreči 4-je taki kovanci.

Odgovor B je tudi napačen, saj vsebuje 2 oranžna kovanca, Patrikova vreča pa le enega.

Odgovor C je pravilen saj se število kovancev posameznega tipa ujema s številom kovancev v Patrikovi vreči.

Odgovor D je napačen, ker vreča vsebuje le 2 zeleno-rumena kovanca, Patrikova vreča pa 4 zeleno-rumene kovance.

Računalniško ozadje

Pri tej nalogi je bilo potrebno določene kovance, ki so sicer izgledali različno, obravnavati kot kovance iste vrste. V računalništvu moramo pri obdelavi podatkov včasih nekatere lastnosti prezreti in stvari, ki so sicer videti drugačne, obravnavati kot enakovredne.

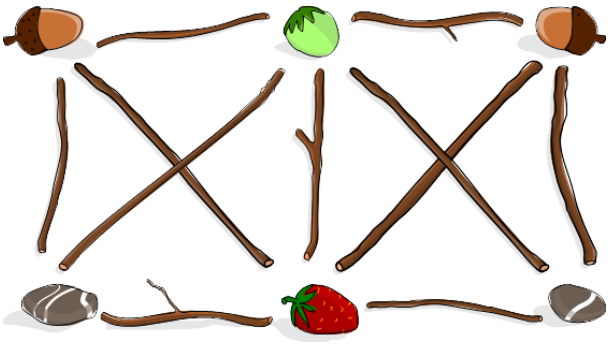
Jagodni tat

državno, 6. in 7. razred

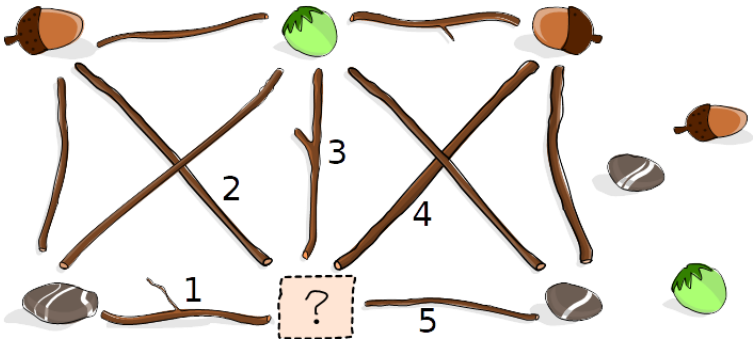
Anja je na tleh ustvarila vzorec iz štirih različnih vrst predmetov: želodov, lešnikov, kamnov in jagod. Nato je dodala palice v skladu z naslednjim pravilom:

Palica leži samo med dvema različnima vrstama predmetov.

Tako izgleda Anjin končan vzorec:



Anjina sestra Zoja vidi vzorec in poje jagodo. Da bi prikrila svoje dejanje, nadomesti jagodo z drugo vrsto predmeta. Pri tem mora odstraniti natančno eno palico, da vzorec ne krši Anjinega pravila.



S katerim predmetom je Zoja zamenjala jagodo in katero palico je odstranila?

Rešitev

Zoja je jagodo zamenja z lešnikom in odstrani palico, označeno s številko 3, saj je ta palica med dvema lešnikoma. Vse ostale palice ostanejo na svojih mestih, saj povezujejo različne predmete.

Če bi jagodo zamenjala z želodom, bi morala odstraniti palici 2 in 4. Če bi jagodo nadomestila s kamnom, pa bi morala odstraniti palici 1 in 5, da bi še vedno veljalo Anjino pravilo.

Računalniško ozadje

Anjin vzorec lahko imenujemo graf, predmete na sliki vozlišča, palice pa povezave med vozlišči. Vozlišči, ki sta povezani, imenujemo sosednji vozlišči.

Podmnožico vozlišč, znotraj katere so vsa vozlišča sosednja eno z drugim, imenujemo klika. Anjin graf vsebuje dve kliki: levo in desno polovico grafa.

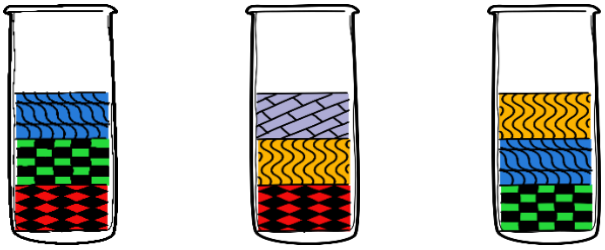
Če želimo pobarvati vozlišča tako, da nobena povezava ne bi povezovala dveh vozlišč iste barve, potrebujemo vsaj toliko barv, kot je velikost največje klike.

Odstranjevanje jagode je enako kot poskus barvanja Anjinega grafa z največ tremi barvami. To ni mogoče, saj je velikost največje klike štiri, zato je morala Zoja odstraniti tudi eno povezavo.

Problem barvanja grafov z minimalnim številom barv ima veliko uporabno vrednost. Uporablja se na primer pri razporejanju tekem na športnih tekmovanjih, pripravi sedežnih redov in tudi pri reševanju sudokuja.

Barvno presenečenje

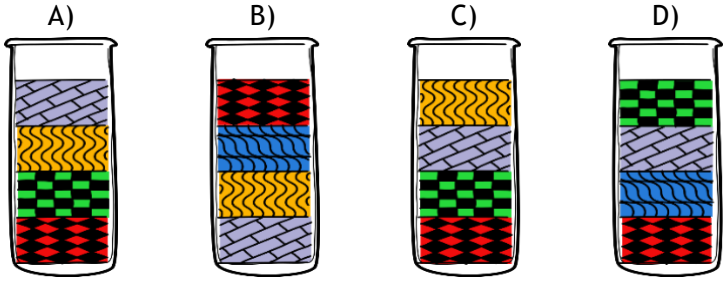
6. do 9. razred



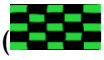
Prijatelj bobra Marka ima jutri rojstni dan. Mark mu za presenečenje želi pripraviti čašo, v kateri bodo plasti različnih tekočin.

Bober Znanko, njegov učitelj naravoslovja, mu je razložil, da bodo tekočine z manjšo gostoto plavale nad tekočinami z večjo gostoto. S tekočinami iz laboratorija mu je pripravil tri primere, kot kaže slika.

Mark bo za prijatelja pripravil čašo s 4 tekočinami. Katere od spodnjih možnosti se je lahko razveselil Markov prijatelj, če je Mark uporabil iste tekočine kot učitelj?



Rešitev

Mark bo prijatelja presenetil s čašo A. Ta je enaka kot drug učiteljev primer, le da ima dodano še zeleno () tekočino, za katero iz učiteljevih primerov vidimo, da je redkejša od rdeče () tekočine – iz 1. primera in hkrati gostejša od rumene () tekočine – iz 3. primera.

Čaše B ne more pripraviti, saj rdeča tekočina ne more plavati nad modro tekočino.

Čaše C ne more pripraviti, saj rumena tekočina ne more plavati nad vijolično.

Čaše D ne more pripraviti, saj zelena tekočina ne more biti nad vijolično tekočino.

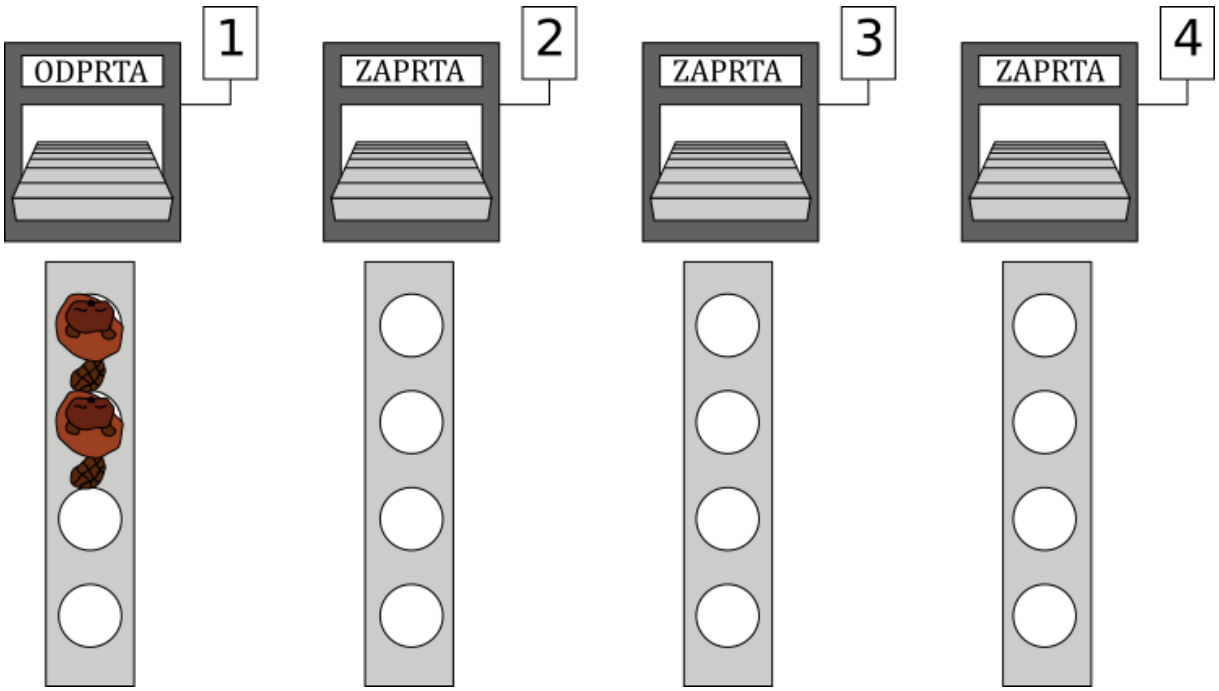
Računalniško ozadje

V matematiki – pa tudi v računalništvu – pogosto govorimo o relacijah in njihovih lastnostih. Relacija "biti gostejši" je primer tranzitivne relacije: če je A gostejši ob B in B gostejši od C, mora biti tudi A gostejši od C. S tem povezana naloga v računalništvu je topološko urejanje, kjer poznamo določeno relacijo med (nekaterimi) objekti, naša naloga pa je, da jih uredimo glede na to relacijo.

Blagajne

6. do 9. razred

V trgovini imajo štiri blagajne. Pred vsako lahko čakajo največ 4 stranke. Na vsaki lahko hkrati strežejo največ eni stranki, za vsako potrebujejo 2 minuti. Na začetku je odprta le blagajna 1.



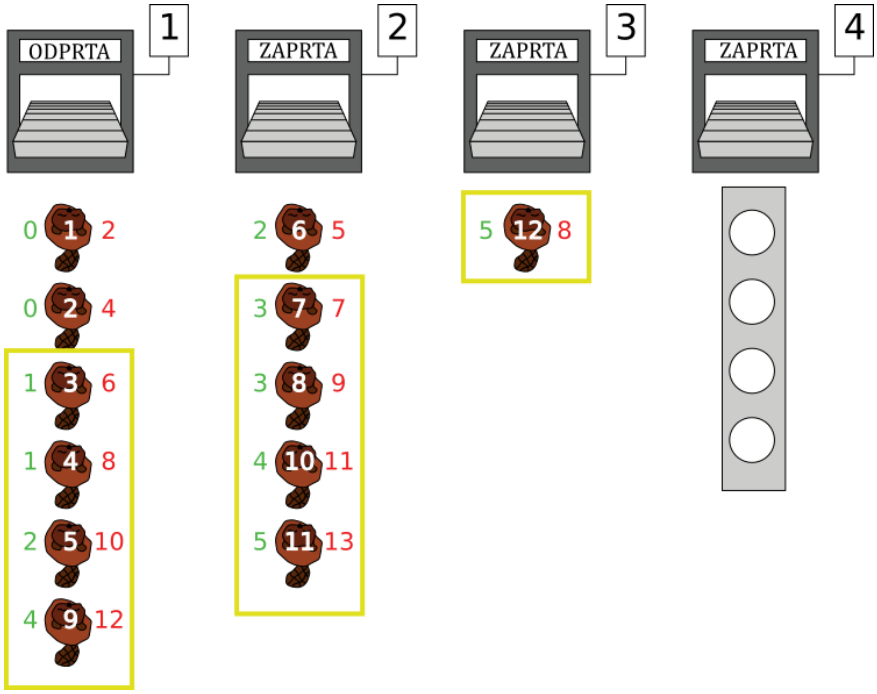
Ko stranka zaključi z nakupovanjem, se postavi na konec prve vrste, v kateri je še prostor. Najprej poskusi na blagajni 1, nato na blagajni 2 itn.

Če so vse vrste odprtih blagajn polne, se odpre nova blagajna in stranka se postavi v njeno vrsto. Toda za odprtje blagajne je potrebna dodatna minuta, zato so za postrežbo prve stranke novo-odprte blagajne potrebne skupaj 3 minute.

Na blagajno pride 12 strank, po dve stranki vsako minuto (prvi dve stranki prideta v času 0, minuto za njima drugi dve stranki itn.). Koliko časa traja, da vse stranke dokončajo svoj nakup?

Rešitev

Vse stranke bodo dokončale nakup v 13 minutah. Na spodnji sliki so z zeleno barvo prikazani časi, kdaj posamezna stranka pride na blagajno, z rdečo pa časi, kdaj stranka odide z blagajne. Druga blagajna se odpre po 2 minutah, ko prideta stranki 5 in 6; 5. stranka se lahko postavi v vrsto prve blagajne, saj je prva stranka ravno zaključila z nakupom, 6. pa se mora postaviti na novo blagajno. Blagajna 3 se odpre po 5 minutah.



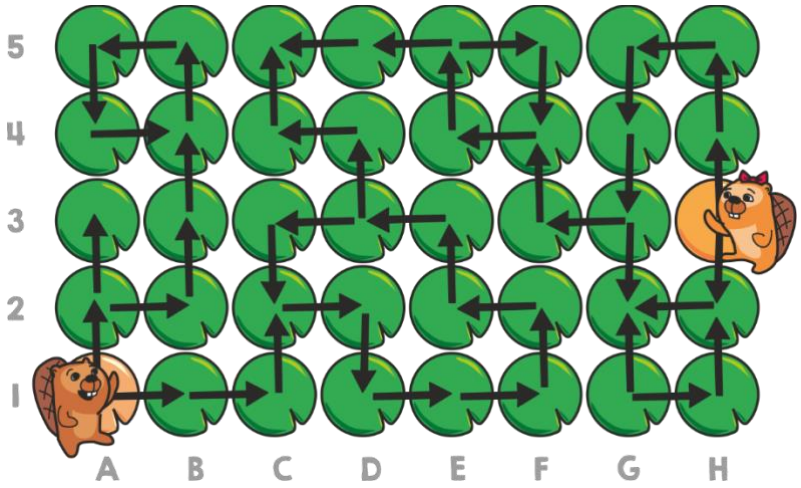
Računalniško ozadje

Naloga prihaja iz zanimivega področja računalništva: modeliranja strežniških sistemov. Namesto o blagajnah tam govorimo o strežnikih, ki dobivajo zahteve v določenih časovnih razmikih in za obravnavo potrebujejo različne čase. Drug zanimiv primer takšnega sistema je cestno omrežje, kjer imamo namesto blagajn križišča in semaforje, naloga pa je zagotoviti čim večjo pretočnost.

Se srečata?

6. do 9. razred

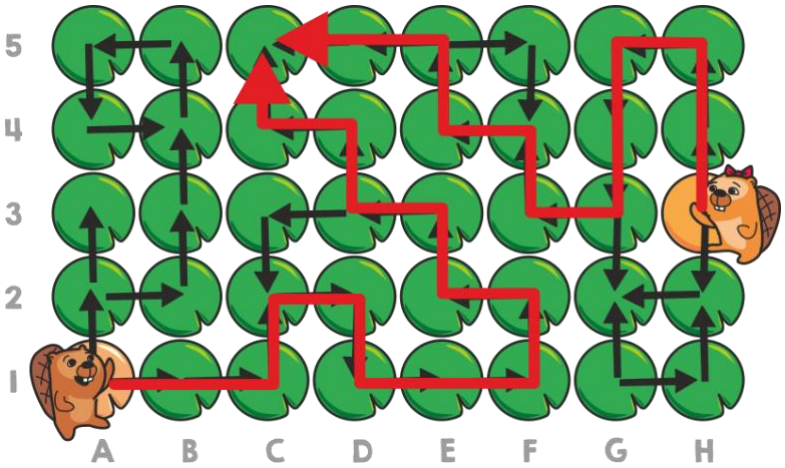
Bobri se po jezeru premikajo po listih lokvanja v smeri puščic. Bob svojo pot začne na listu A1 in Nina na listu H3.



Ali se bobra lahko srečata? In če, na katerem listu?

Rešitev

Bobra se lahko srečata na listu C5.

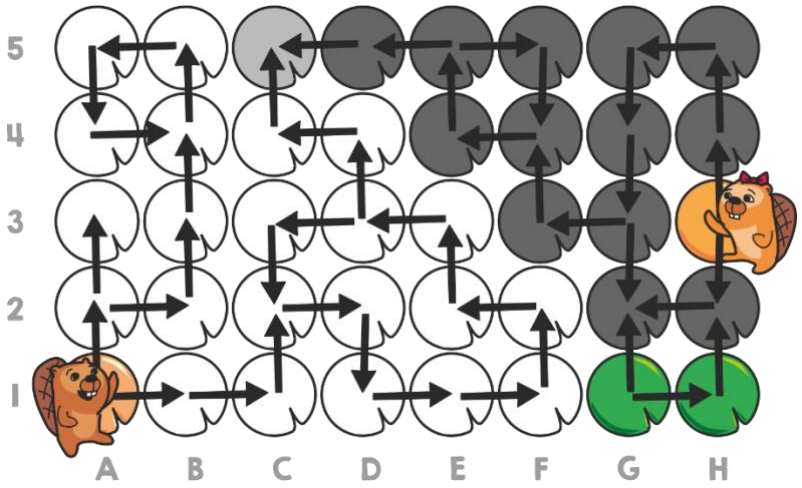


Bob ima na svojem začetnem položaju dve možnosti: lahko se premakne navzgor na list A2 ali pa desno na list B1. Če se premakne navzgor pristane v slepi ulici pri A3 ali pa se na listu B4 zacikla v zanki. Če se odloči premakniti desno (na B1) pot nadaljuje na D3. Na listu D3 se lahko premakne levo (na C3) in vstopi v zanko, ki ga zopet pripelje na list D3, ali se premakne navzgor (na D4), kjer konča v slepi ulici na listu C5.

Nina ima na začetnem polju ravno tako dve možnosti. Če se premakne navzdol (na H2), pristane v slepi ulici (na G2). Če se premakne navzgor (na H4), pa pristane v na listu G3. Od tam lahko zopet zaide v slepo ulico (na G2) ali pa do lista E5. Na listu E5 se lahko premakne desno (na F5) in vstopi v zanko, ki jo pripelje nazaj na list E5, ali se premakne levo (na D5), kjer konča v slepi ulici na listu C5.

Kot že vemo, lahko Bob tudi doseže polje C5, kar pomeni, da je edini možni list, na katerem se lahko srečata Bob in Nina, list C5. Spodnja slika prikazuje pot, po kateri lahko oba bobra dosežeta list C5.

K problemu lahko pristopimo tudi drugače. Naslednja slika prikazuje liste, na katerih lahko pristane Bob (obarvane belo) in liste na katerih lahko pristane Nina (obarvane temno sivo). Iz slike je razvidno, da je list C5 edini, ki je skupen obema.



Računalniško ozadje

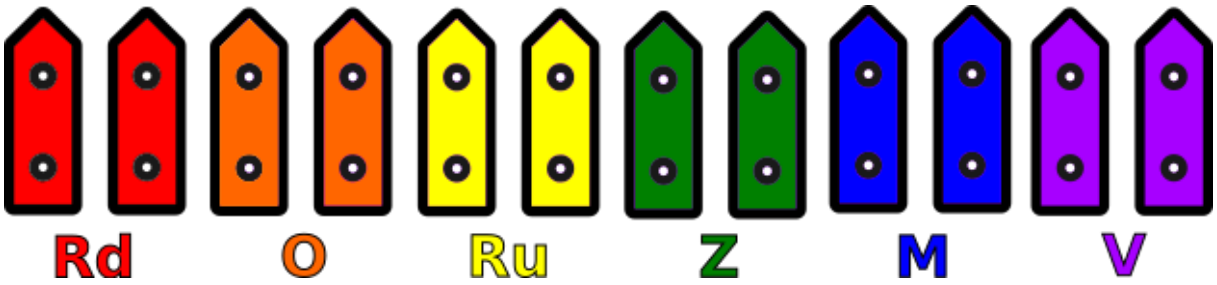
Pri ustvarjanju zadnje slike smo sledili premikom posameznega bobra. Ko smo naleteli na slepo ulico ali zanko, smo se vrnili na zadnji razcep, torej na list, kjer se je bilo potrebno odločiti, po kateri puščici nadaljevati pot. Z lista smo se nato premaknili na list v smeri druge, še neuporabljene puščice. Tako smo dobili množico listov, ki jih lahko obišče posamezen bober. Računalniki pri reševanju težkih problemov pogosto uporabljajo zelo podoben postopek.

Ta algoritem, pri katerem računalnik preverja različne možne korake, imenujemo *vračanje* (angl. *backtracking*).

Barvanje ograje

državno, 6. do 9. razred

Bober Janez želi pobarvati ograjo z 12 letvami. Ograjo želi pobarvati z rdečo (Rd), oranžno (O), rumeno (Ru), zeleno (Z), modro (M) in vijolično (V) barvo, pri čemer želi, da sta po dve letvi vsake barve.



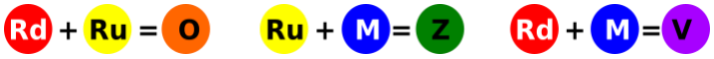
Ima tri vedra, polna rdeče, rumene in modre barve, ter tri prazna vedra za mešanje barv. Prazna vedra imajo tri oznake. Vsaka oznaka označuje četrtino vedra.

Z enim polnim vedrom barve lahko pobarva štiri letve. Oranžno, zeleno in vijolično barvo nameša po naslednjih pravilih:

rdeča + rumena = oranžna

rumena + modra = zelena

rdeča + modra = vijolična



Koliko letev lahko bober Janez pobarva po svojih željah?

Rešitev

Pobarva lahko vseh dvanajst letev.

Janez najprej pobarva dve letvi z rdečo barvo, pri čemer porabi pol vedra rdeče barve, dve letvi z rumeno barvo, pri čemer porabi pol vedra rumene barve, in dve letvi z modro, pri čemer porabi pol vedra modre barve. Od vsakega vedra barve mu tako ostane pol vedra barve.

Za oranžni letvi zmeša četrt vedra rumene barve s četrt vedra rdeče barve in tako dobi pol vedra oranžne barve, s katero pobarva dve letvi.

Za zeleni letvi zmeša četrt vedra rumene barve s četrt vedra modre barve in tako dobi pol vedra zelene barve, s katero pobarva dve letvi.

Za vijolični letvi zmeša četrt vedra modre barve s četrt vedra rdeče barve in tako dobi pol vedra vijolične barve, kar zadostuje za barvanje dveh letev.

Računalniško ozadje

V nalogi smo uporabljali barvni model, kakršnega ste spoznali pri likovnem pouku. Pri računalništvu pa uporabljamo drugačne barvne modele. Pri tisku uporabljamo barvni model CMY, kar pomeni, da model kot osnovne barve uporablja cian (svetlo modra), magento (vijolična) in rumeno barvo.

Pri kodiranju slik v računalniku pa pogosteje uporabljamo barvni model RGB, ki kot osnovne barve uporablja rdečo, zeleno in modro barvo. Ta model nam omogoča, da pravilno prikažemo barve na zaslonu.

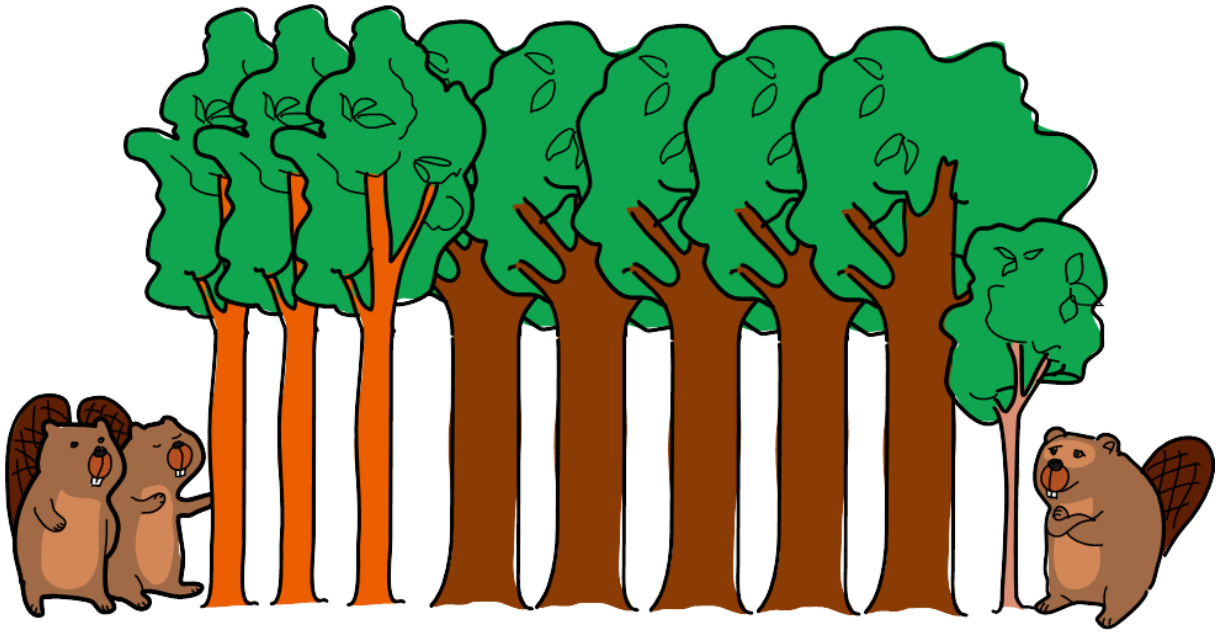
Trije bobri

državno, 6. do 9. razred

Trije bobri podirajo drevesa. Vsak bober dela sam. Drevesa imajo debla različnih debelin, zato jih bobri podirajo različno dolgo.

Število dreves iste debeline	Čas, potreben za podiranje enega drevesa
5	4 ure
3	3 ure
1	1 ura

Bobri lahko podrejo drevesa v poljubnem vrstnem redu, vendar morajo vedno drevo, ki ga začnejo podirati, podreti do konca, preden začnejo podirati naslednjega.



Najmanj koliko časa potrebujejo, da podrejo vsa drevesa?

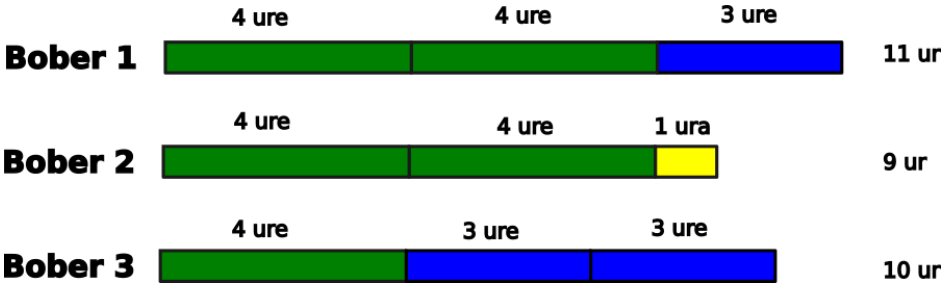
Rešitev

Bobri potrebujejo enajst ur, da podrejo vsa drevesa.

Lahko se zdi, da je pravilen odgovor 10 ur:

$$\frac{5 \cdot 4 + 3 \cdot 3 + 1 \cdot 1}{3} = 10$$

Vendar to ne drži, saj dreves ne moremo razdeliti v tri enakovredne skupine, bobri pa si med seboj ne pomagajo. Vsak bober mora podiranje drevesa, katerega se je lotil, dokončati sam. Zato bobri končajo ob različnih časih:



V tem primeru vsak bober vedno začne podirati največje drevo, ki je še na voljo (to je le ena od možnih rešitev).

Računalniško ozadje

Podobno nalogo, kot si jo pri reševanju zgornje naloge opravil ti, vsak trenutek delovanja računalnika izvaja razvrščevalnik. Za razvrščanje opravil, ki se bodo v vsakem trenutku izvajala na procesorju, uporablja posebne algoritme, ki omogočajo optimalno delovanje računalnika.

Pikčaste kukavice

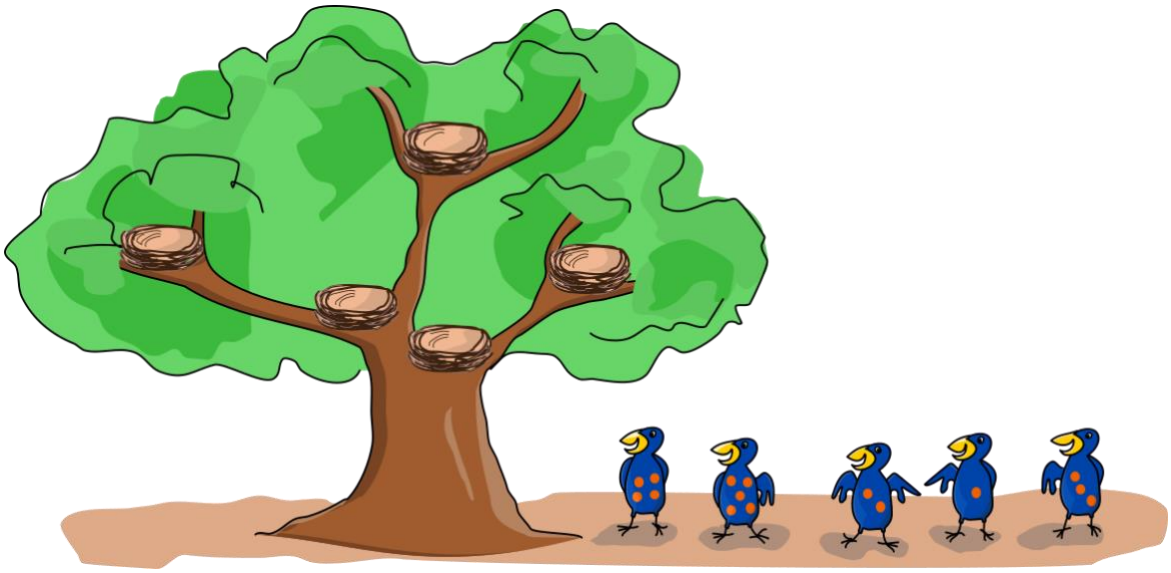
državno, 6. do 9. razred

Ob drevesu je pet pikčastih kukavic. Druga za drugo — po vrsti od leve proti desni — plezajo na drevo in se naselijo v prazna gnezda. Tista s štirimi pikami je prva. Vsaka pikčasta kukavica si izbere gnezdo po naslednjem pravilu:

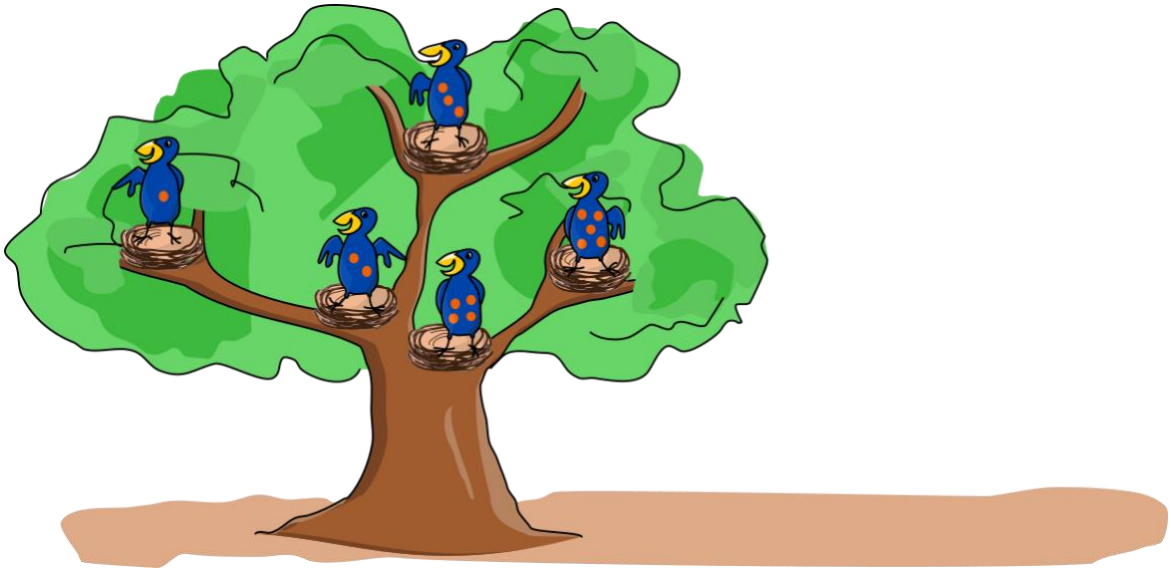
Začne na dnu drevesa in ponavlja naslednje korake, dokler ne najde praznega gnezda:

4. Vzpenja se po drevesu, dokler ne pride do gnezda.
5. Če je gnezdo prazno, se naseli vanj in tam ostane.
6. Če je gnezdo zasedeno, pogleda pikčasto kukavico, ki sedi v njem. Če ima ta:
    - več pik, nadaljuje na levo stran drevesa,
    - manj ali enako pik, nadaljuje na desno stran drevesa.

Katera pikčasta kukavica se bo naselila v gnezdo na najvišji veji?



Rešitev



Prva kukavica, ki ima 4 pike, se bo naselila v prvo gnezdo. Nato prileti kukavica s petimi pikami. Ko pride do prvega gnezda, tam že sedi kukavica s 4 pikami. Ker ima kukavica, ki že sedi v gnezdu, manj pik, odide druga kukavica na desno. Tam pride do gnezda, ki je prazno, zato se vanj naseli. Tretja prileti kukavica z dvema pikama. Ker je 2 manj kot 4, gre po levi strani drevesa do prvega gnezda, ki je prazno, zato se vanj naseli. Sledi kukavica z eno piko. Prvo gnezdo je zasedeno s kukavico s štirimi pikami in ker je 1 manj kot 4, gre na levo vejo. Tudi tu je gnezdo zasedeno, v njem je že naseljena kukavica z dvema pikama. Ker ima več pik, kot jih ima sama, nadaljuje na levo stran drevesa in se naseli v prostem gnezdu. Zadnja prileti kukavica s tremi pikami. Ker je 3 manj kot 4, gre po levi veji drevesa do gnezda, v katerem je kukavica z dvema pikama. Ker ima sama več pik, nadaljuje po desni veji drevesa. Ko pride do gnezda na najvišji veji, je to prazno, zato se naseli vanj.

Računalniško ozadje

V nalogi smo simulirali delovanje urejenih iskalnih dreves. To je eden od načinov, na katerega si računalniki organizirajo podatke glede na določen kriterij (kot je tu število pik), da jih kasneje hitreje najdejo.

Mačje slike

6. do 9. razred, srednja šola

Tinka je navdušena nad svojo mucko, zato jo pogosto fotografira in slike objavlja na Bobrogramu.

Iz slik želi izdelati video, ki bi kronološko prikazoval prva tri leta življenja njene mucke. Zato mora v svoj računalnik prenesti vse slike in jih ustrezno poimenovati.

Računalnik ureja slike po imenu, od a do ž in od 0 do 9. Tinka hoče vse slike urediti od najstarejše do najnovejše, zato imenu vedno doda tudi datum.

Kako naj Tinka poimenuje datoteke s slikami (primeri so za sliko, posneto 19. avgusta 2021)?

A) mucka_avgust_19_2021
B) mucka_19_avgust_2021
C) mucka_19_8_2021
D) mucka_19_08_2021
E) mucka_2021_avgust_19
F) mucka_2021_19_8
G) mucka_2021_08_19
H) mucka_2021_8_19

Rešitev

Tinka mora slike poimenovati na način, prikazan pod G (mucka_2021_08_19).

Poimenovanja pod A, B, C in D niso pravilna. Če imamo slike iz različnih let, želimo, da so vse slike istega leta skupaj, čeprav imajo različne dneve in mesece. Zato mora zapis leta biti pred mesecem in dnevom, saj imajo le tako vse slike iz istega leta enak začetni del imena in so v računalniku po urejanju skupaj. Če je dan ali mesec zapisan pred letom (kot je to v primerih A, B, C in D), potem bi bili dve sliki, ki sta posneti na isti dan v mesecu in isti mesec v letu, a eno leto narazen, po urejanju skupaj, ena za drugo. Na primer, fotografija z 19. avgusta 2020 bi

bila po urejanju postavljena tik pred fotografijo z 19. avgusta 2021, čeprav bi bilo med datotekami tudi več fotografij, posnetih v letu 2021 pred 19. avgustom. To pa ni način, ki ga želi Tinka.

Tudi poimenovanje pod E ni pravilno. Hitro lahko ugotovimo, da mora biti mesec zapisan s številko in ne z imenom, saj si imena mesecev ne sledijo po abecedi. Če mesec zapišemo z imenom, so slike iz aprila in avgusta po urejanju blizu skupaj, slike iz marca in maja pa morajo sicer biti blizu aprilu, ne pa tudi avgustu.

Tudi poimenovanje pod F ni pravilno. Podobno kot prej pri letu, mora biti iz enakega razloga tudi mesec zapisan pred dnevom. Če bi bila tako leto kot mesec enaka pri dveh slikah, bi morali biti obe sliki blizu skupaj. A način pod F (mucka_2021_19_8) ne deluje tako, saj bi na primer sliko z 19. septembra 2021 postavil med sliki z 18. avgusta 2021 in 20. avgusta 2021.

Tudi poimenovanje pod H ni pravilno. Ker je vseh mesecev dvanajst, pri zapisu meseca s številko potrebujemo za mesece od januarja do septembra eno številko, medtem ko za zapis mesecev oktober, november in december potrebujemo dve številki (prva številka je pri vseh treh mesecih enaka 1). Ker računalnik ureja datoteke po imenu (po črkah oz. številkah od leve proti desni), urejanje ne bo pravilno v primeru, ko bo primerjal mesec, zapisan z eno številko, z mesecem, zapisanim z dvema številkama. Poglejmo primer: če imamo dve sliki, prvo z 19. avgusta 2021 in drugo z 19. decembra 2021, in za zapis datuma uporabimo vzorec pod H, dobimo 2021_8_19 in 2021_12_19. Pri urejanju je prvih pet znakov enakih, primerjava šestega znaka (8 in 1) pa uredi 2021_12_19 pred 2021_8_19 (ker je 1 manjše od 8). To seveda ni pravilno in tako urejanje ne da želenega rezultata.

Pravilno primerjavo mesecev pri datumu bi dobili le v primeru, da mesec vedno zapišemo z dvema številkama. Tako januar zapišemo kot 01, februar 02 in tako dalje do 09 za september. Le v tem primeru bo abecedna primerjava dveh mesecev pravilna. To pa je način poimenovanja pod G, ki je tudi edini pravilen odgovor.

Računalniško ozadje

Računalniško ozadje je tu očitno – in nauk tudi. Pametno poimenuj datoteke!

Te zanima še kaj o zapisu datumov?

Na ta način – letnica, mesec, dan – pišejo datume Japonci. To je videti praktično. Malo manj praktično je, da Japonci (uradno) štejejo leta glede na vladavine cesarjev. Ko se zamenja cesar, začnejo ponovno šteti od 0, tako da morajo ob vsaki letnici povedati, za katerega cesarja gre. Sicer pa so ta sistem odkrili že Rimljani, ko so govorili o "tem in tem letu vladavine cesarja tega in tega".

Prvi računalniški sistemi so, predvsem zaradi varčevanja s prostorom, za predstavitev leta uporabljali le dve številki, na primer 81 za leto 1981. Tak način se je uporabljal do leta 2000, ko so ga zaradi dvoumnosti morali opustiti (ali leto 21 pomeni 1921 ali 2021?).

Marsovska koda

6. do 9. razred, srednja šola

Marsovci kodirajo sporočila tako, da ima vsaka zakodirana beseda dva dela: prvi del je številska vrednost besede, ki jo kodirajo, drugi del pa sestavljajo zaporedne številke po abecedi urejenih črk te besede.

Pri kodiranju si pomagajo z naslednjo tabelo:

A	B	M	N	O	R	S	T	U
1	2	4	10	50	180	300	650	960

Zveni zapleteno? Poglejmo primer kodiranja. Besedo MARS zakodirajo tako, da najprej izračunajo številsko vrednost besede, to je vsota vrednosti posameznih črk iz tabele: 4 + 1 + 180 + 300 = 485.

Nato vse črke v besedi uredijo po abecedi (za besedo MARS dobijo AMRS) in vsaki črki določijo zaporedno številko: A = 1, M = 2, R = 3 in S = 4. Besedo MARS bi s temi zaporednimi številkami zapisali kot 2134.

Na koncu sestavijo kodo besede MARS kot kombinacijo številske vrednosti in zaporednih številk, oboje ločeno s podpičjem: 485;2134.

Kako se v marsovski kodi zapiše besedo SATURN?

A) 1440;415632
B) 1440;514623
C) 2101;415632
D) 2101;514623

Rešitev

Pravilna rešitev je C: 2101;415632.

Do rešitve lahko pridemo tako, da besedo zakodiramo po podanih pravilih. Prvi del kode besede SATURN je vsota posameznih črk iz tabele, torej 300 + 1 + 650 + 960 + 180 + 10 = 2101. Za drugi del pa najprej črke uredimo po abecedi in jih oštevilčimo: A = 1, N = 2, R = 3, S = 4, T = 5, U = 6. Besedo SATURN zapišemo z zaporednimi številkami urejenih črk kot 415632. Če oba dela sestavimo, dobimo kodo za besedo SATURN: 2101;415632.

Lahko pa pravilno rešitev med podanimi odgovori poiščemo hitreje, ne da bi dejansko izračunali kodo besede SATURN. Poglejmo najprej prvi del, številsko vrednost besede. V odgovorih najdemo dve možnosti: 1440 in 2101. Beseda SATURN vsebuje črki T in U iz skrajno desnega dela tabele z velikimi vrednostmi 650 in 960. Takoj lahko vidimo, da je že vsota teh dveh vrednosti večja kot 1440. Torej sta odgovora A in B napačna, pravilna vsota pa mora biti 2101. Sedaj preverimo še drugi del kode. V odgovorih najdemo dve možnosti: 415632 in 514623. Obe

vsebujeta vsa števila od 1 do 6, torej obe opisujeta besedo iz šestih (različnih) črk. Če črke besede SATURN uredimo po abecedi, dobimo ANRSTU. Vidimo, da ima črka N zaporedno številko 2, črka R pa 3. Torej zaporedje 514623 predstavlja besedo, ki se konča na R (in ne na N), zato to ne more biti prava rešitev (pravzaprav zaporedje 514623 predstavlja besedo TASUNR) torej je odgovor D napačen. Ostane nam torej le še odgovor C, ki mora biti pravilen.

Računalniško ozadje

V nalogi smo za kodiranje uporabili transpozicijsko metodo šifriranja v kombinaciji s kontrolno vsoto (angl. checksum). Pri transpozicijski metodi šifriranja se spremeni vrstni red črk besede, ključ pa določa, kakšen je pravilni vrstni red. Kontrolna vsota sporočila se uporablja bolj pri kodiranju kot pri šifriranju. Omogoča nam, da zaznamo napake pri prenosu sporočila.

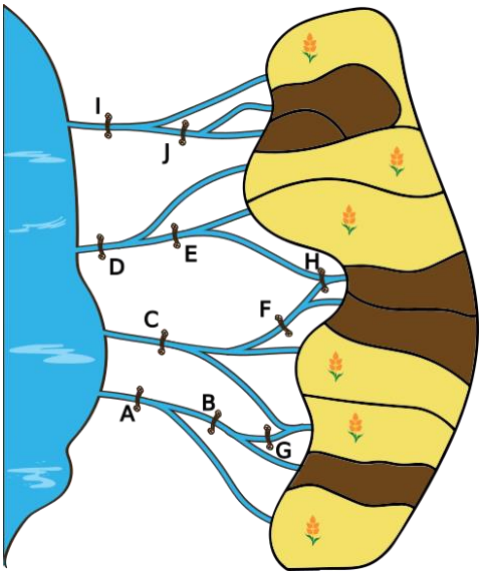
Namakanje polj

6. do 9. razred, srednja šola

Namakalni sistem, ki ga prikazuje slika, lahko namaka 11 polj, dotok vode pa določamo s sistemom desetih zapornic (označene s črkami od A do J), ki lahko odprejo ali zaprejo pretok vode iz jezera.

Na rumenih poljih na sliki raste pšenica (), rjava polja pa so zaraščena s plevelom. Trenutno so vse Zapornice dvignjene, a bi želeli namakati le polja s pšenico.

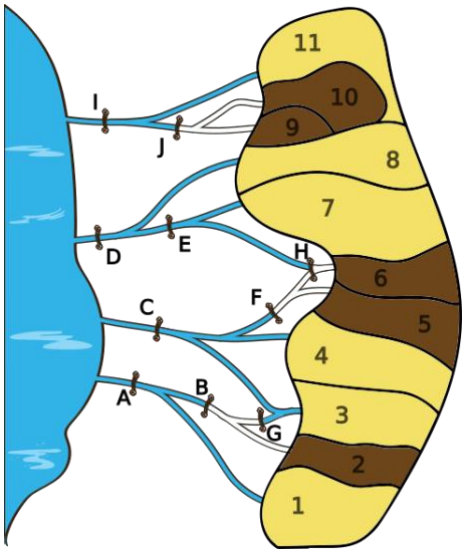
Katere zapornice moramo zapreti, da bo voda prišla do vseh polj s pšenico in do nobenega polja s plevelom?



Rešitev

Zapreti moramo zapornice B, F, G, H in J.

Če ne zapremo zapornic B, F, G, H in J, bo voda dosegla polja s plevelom. Če zapremo še kakšno dodatno zapornico, voda ne bo dosegla vseh polj s pšenico. Rešitev lahko dobimo tako, da preverimo vsako posamezno zapornico in določimo, ali mora biti odprta ali zaprta:



- A mora biti odprta, da se lahko namaka polje 1.
- B mora biti zaprta, da preprečimo namakanje polja 2. Ta zapornica sicer priskrbi vodo tudi polju 3, a to polje lahko namakamo tudi preko zapornice C.
- C mora biti odprta tako za polje 4 kot tudi za polje 3, saj slednjega ne moremo namakati preko A, saj je zapornica B zaprta.
- D mora biti odprta za namakanje polj 7 in 8.
- Tudi E mora biti odprta za polje 7.
- F mora biti zaprta, da preprečimo namakanje polja 5.
- G mora biti zaprta, saj namakamo polja na ravnini in bi ob odprti zapornici C voda iz kanala proti polju 3, pritekla k polju 2.
- Tudi H mora biti zaprta, četudi je zaprta že zapornica F, saj bi voda lahko pritekla preko odprtih zapornic D in E.
- I mora biti odprta za namakanje polja 11.
- Zaprta zapornica J pa prepreči namakanje polj 9 in 10.

Računalniško ozadje

V nalogi priteče voda do polja na podlagi več *pogojev*. Na primer, voda priteče do polja 7, če sta odprti obe pregradi D in E. Voda priteče do polja 3, če je odprta pregrada G in drži kateri koli od naslednjih dveh pogojev: (1) odprta je pregrada C ali (2) odprti sta pregradi A in B.

Take vrste kombiniranih pogojev oblikujemo z uporabo Boolovih operatorjev IN (kadar sta pregradi zaporedoma na istem kanalu) oziroma ALI (kadar lahko voda priteče do istega mesta preko dveh različnih kanalov). Rezultat takih pogojev je vedno ali *resnično* (angl. *true*) ali *neresnično* (angl. *false*), kar imenujemo *Boolova vrednost* ali *logična vrednost*.

Sneguljčica in prepirljivi palčki

6. do 9. razred, srednja šola

Sneguljčica je navdušena nad matematiko, zato je palčke poimenovala s številkami, kot je prikazano na sliki.

Palčki so se nekega večera skregali. Palček 12 prijateljuje s palčkoma 1 in 2, palček 13 s palčkoma 1 in 3 ter palček 23 s palčkoma 2 in 3. Le prijazni palček 123 je prijatelj z vsemi.

Sneguljčica si je zamislila pet ukazov, s katerimi prepirljive palčke kroti tako, da jih pošlje iz hiše ali pokliče nazaj. Ko zakliče določeno številko, palčki ravnajo po teh pravilih:

- ukaz »2« pokliče v hišo palčka 2 in njegove prijatelje (tj. 12, 23 in 123);
- ukaz »3« pokliče v hišo palčka 3 in njegove prijatelje (tj. 13, 23 in 123);
- ukaz »4« zahteva, da palček 1 in njegovi prijatelji zamenjajo položaj (tj. tisti, ki so v hiši, gredo ven, tisti, ki so zunaj, pa noter);
- ukaz »5« zahteva, da palček 2 in njegovi prijatelji zamenjajo položaj;
- ukaz »6« zahteva, da palček 3 in njegovi prijatelji zamenjajo položaj.

Poglejmo primer: če so palčki 1, 2 in 12 v hiši ter palčki 3, 13, 23 in 123 zunaj hiše, Sneguljčica pa zakliče: »5«, potem gresta palčka 2 in 12 takoj ven, palčka 23 in 123 pa prideta v hišo. Ostali ostanejo, kjer so.

Neko popoldne, ko so bili Sneguljčica in vsi palčki v hiši, pride na obisk princ. Sneguljčica bi želela ostati v hiši sama s princem. Katero je najkrajše zaporedje ukazov, s katerimi bo vse palčke poslala iz hiše?

Rešitev

Rešitvi sta dve, 42536 (na tekmovanju je bila med naštetimi rešitvami le ta) in 43625.

Ker nimamo neposrednega ukaza, s katerim bi katerega koli palčka poslali iz hiše, moramo za to uporabiti ukaze 4, 5 in 6. Problem pa je v tem, da pri uporabi teh ukazov nekateri palčki pridejo nazaj v hišo, zato moramo poskrbeti, da se ti palčki pred uporabo ukaza nahajajo na pravem mestu.

Ker nimamo ukaza, ki bi palčka 1 in njegove prijatelje priklical nazaj v hišo, lahko začnemo z ukazom »4« in pošljemo te palčke iz hiše. Tako v hiši ostanejo še 2, 3 in 23. V tem trenutku so nekateri prijatelji od 2 in 3 izven hiše, zato jih moramo poklicati v hišo, preden pošljemo vse ven z ukazom 5 oz. 6. Tako lahko uporabimo kombinacijo ukazov »25« in »36« (v katerem koli vrstnem redu), da skupino prijateljev pokličemo v hišo in jih potem pošljemo vse skupaj ven.

Ukazi Sneguljčice in posledične spremembe položaja palčkov prikazujeta tabeli.

Če Sneguljčica uporabi ukaze »42536«:

Ukaz	**4**	**2**	**5**	**3**	**6**
V hiši	2, 3, 23	12, 2, 3, 23, 123	3	3, 13, 23, 123	
Izven hiše	1, 12, 13, 123	1, 13	1, 2, 12, 13, 23, 123	1, 2, 12	1, 2, 3, 12, 13, 23, 123

Če Sneguljčica uporabi ukaze »43625«:

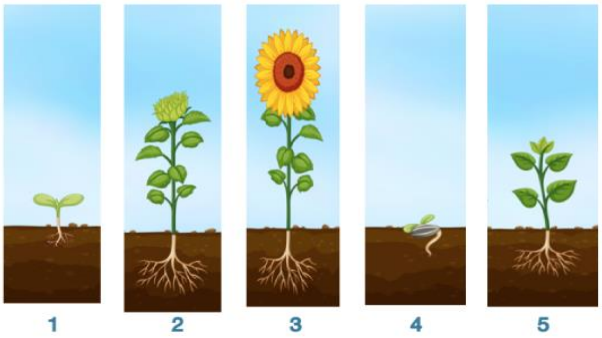
Ukaz	**4**	**3**	**6**	**2**	**5**
V hiši	2, 3, 23	2, 3, 13, 23, 123	2	2, 12, 23, 123	
Izven hiše	1, 12, 13, 123	1, 12	1, 3, 12, 13, 23, 123	1, 3, 13	1, 2, 3, 12, 13, 23, 123

Računalniško ozadje

Kako bi se lotili naloge z računalnikom? Zapisali bi vse možne razmestitve palčkov. Povezali bi tiste, za katere obstaja ukaz, ki nas pripelje iz ene v drugo in ob črto napisali ukaz. Potem bi poiskali najkrajšo pot po tej povezavah od trenutnega do želenega stanja in si zabeležili ukaze na povezavah na tej poti.

V resnici gre torej (spet) za nalogo iz iskanja najkrajših poti.

Borut ima pet slik, ki prikazujejo faze rasti sončnice. Ker slike niso v pravilnem vrstnem redu, jih želi preurediti tako, da bodo faze rasti prikazane po vrsti od leve proti desni. Naenkrat lahko zamenja le dve sliki.

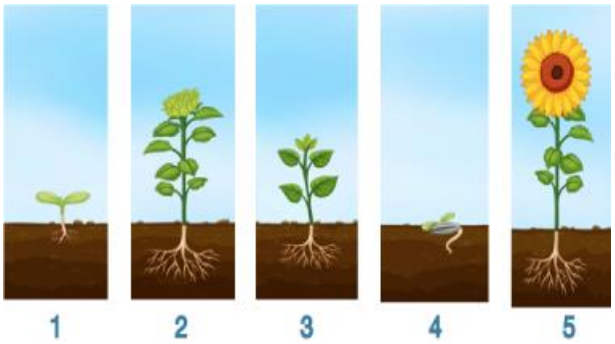


Najmanj koliko zamenjav mora narediti, da postavi slike v pravilni vrstni red?

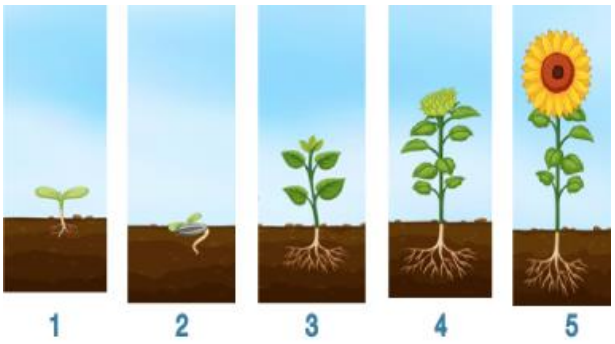
Rešitev

Narediti mora najmanj 3 zamenjave, da postavi slike v pravilni vrstni red.

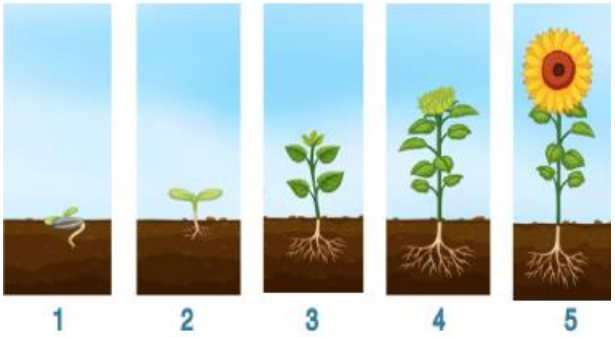
Spodnja tabela prikazuje eno možno rešitev.

Zamenjava	Rezultat
1) Zamenjaj sliki 3 in 5	

2) Zamenjaj sliki 2 in 4



3) Zamenjaj sliki 1 in 2



Slik ne moremo postaviti v pravilen vrstni red z manj kot tremi zamenjavami, ker:

a) mora biti slika zadnje faze (cvetoča sončnica) na zadnjem mestu, zato moramo premakniti sliko 3 na mesto 5 (najmanj ena zamenjava);
b) mora biti slika prve faze (kaljenje) na prvem mestu, zato moramo premakniti sliko 4 na mesto 1 (najmanj ena zamenjava);
c) sliko 2 moramo postaviti na mesto 4, kar zahteva najmanj še eno zamenjavo.

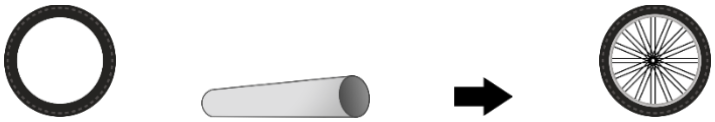
Ker se mesta 5, 1, 4 ne prekrivajo, potrebujemo najmanj te tri zamenjave.

Računalniško ozadje

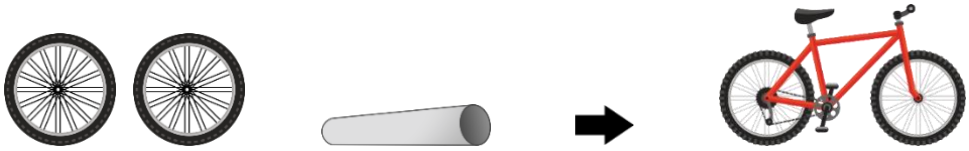
Kot pri prejšnji nalogi gre tudi tu v bistvu za iskanje najkrajše poti. Omejitev na menjavo parov pa je smiselna zato, ker tudi pri računalniškem preurejanju podatkov pogosto zamenjujemo pare podatkov.

Stara roba, nova raba 6. do 9. razred, srednja šola

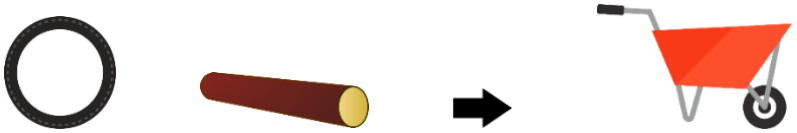
Luka zbira odpadni material in ga uporabi za izdelavo novih uporabnih stvari. Zna izdelati kolo, bicikel, tricikel in samokolnico. Slike kažejo, katere materiale potrebuje za vsakega od njih.



guma + kovinska palica → kolo



2 kolesi + kovinska palica → bicikel



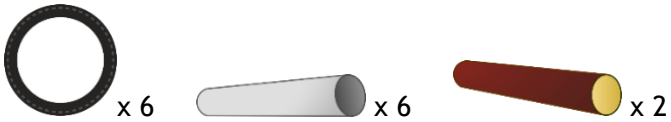
guma + lesena palica → samokolnica

kolo + bicikel → tricikel

Luka svoje izdelke prodaja po naslednjih cenah:

kolo: 10 € bicikel: 100 € samokolnica: 50 € tricikel: 150 €

Luka ima na razpolago naslednji material: 6 gum, 6 kovinskih palic in 2 leseni palici.



Koliko lahko največ zasluži s prodajo svojih izdelkov?

Rešitev

Zasluži lahko največ 300 €.

Verjetno najprej pomislimo, da mora Luka narediti tricikel, saj je ta največ vreden. Za izdelavo tricikla potrebuje 2 kolesi in 1 kovinsko palico (da naredi bicikel) ter še 1 kolo. Za izdelavo 3 koles potrebuje 3 gume in 3 kovinske palice. Torej mu ostanejo še 3 gume, 2 kovinski palici in 2 leseni palici.

Ta material ne zadošča, da bi naredil še en bicikel (drug najdražji izdelek), saj ima eno kovinsko palico premalo. Torej lahko izdela le kolesa in samokolnice. Ker so samokolnice dražje od koles, bo izdelal 2 samokolnici, za kar bo porabil 2 gumi in 2 leseni palici. Ostane mu še 1 guma in 2 kovinski palici. Izdela lahko še 1 kolo, ostane pa mu 1 kovinska palica, ki je ne more porabiti.

Skupna vrednost vseh izdelkov (1 tricikel, 2 samokolnici, 1 kolo) bi tako bila 150 + 2 x 50 + 10 = 260 €. Torej, če Luka izdela tricikel, lahko zasluži največ 260 €.

Kaj pa, če se Luka ne odloči izdelati tricikla? Potem bi lahko izdelal 2 bicikla, za kar bi porabil 4 gume in 6 kovinskih palic, ostali pa bi mu še 2 gumi in 2 leseni palici. To bi zadostovalo za izdelavo 2 samokolnic. V tem primeru bi tudi porabil ves razpoložljiv material.

Skupna vrednost vseh izdelkov (2 bicikla in 2 samokolnici) bi tako bila 2 x 100 + 2 x 50 = 300 €. To je tudi največ, kar lahko zasluži z razpoložljivim materialom.

Računalniško ozadje

Učinkovita uporaba virov je pogost problem, katerega poskušajo računalničarji reševati s programi za optimizacijo. V ta namen se uporablja več različnih algoritmov. Prvi opisan v rešitvi je požrešni algoritem, ki vedno najprej poskuša z elementom največje vrednosti. Čeprav to dobro deluje v nekaterih primerih, so pogoste tudi situacije, kjer začetna izbira elementa z največjo vrednostjo omeji število možnih naslednjih elementov in posledično ne pripelje do optimalne rešitve. Tak primer smo imeli tudi v naši nalogi.

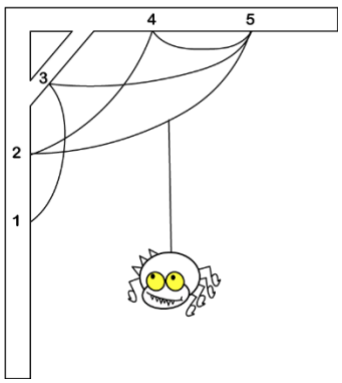
Vzorec pajkove mreže 2

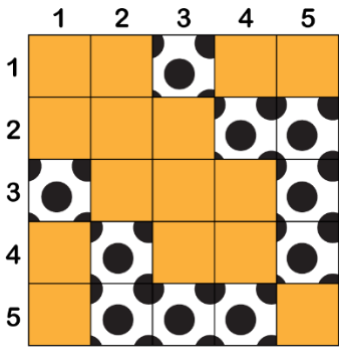
6. do 9. razred, srednja šola

Modna oblikovalka Vanda se zgleduje po naravi. Njen zadnji hit so prešite odeje, ki nosijo vzorec pajkove mreže. Vanda vzorec pripravi tako, da prešteje, na koliko mestih je pajkova mreža pritrjena (N pritrditvenih mest), in nato sestavi odejo kvadratne oblike iz N x N kosov tkanine.

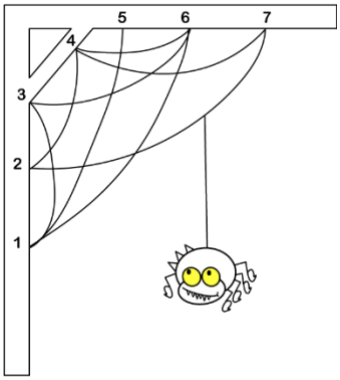
1) Za vsako nit pajkove mreže, ki teče med točkama X in Y, postavi dva kosa pikčaste tkanine:
   a) en kos postavi na mesto, kjer se stikata vrsta X in stolpec Y,
   b) drugi kos postavi na mesto, kjer se stikata vrsta Y in stolpec X.
2) Vsi preostali kosi tkanine so enobarvni.

Tako bi na primer za pajkovo mrežo na levi (s petimi pritrditvenimi mesti) sestavila prešito odejo na desni.



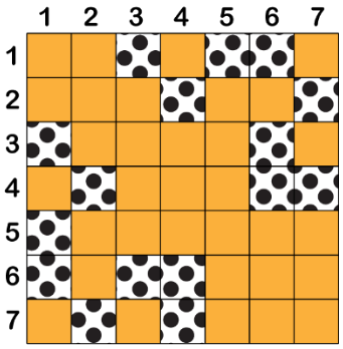


Vanda je našla še eno zanimivo pajkovo mrežo s sedmimi pritrditvenimi mesti (spodnja slika).

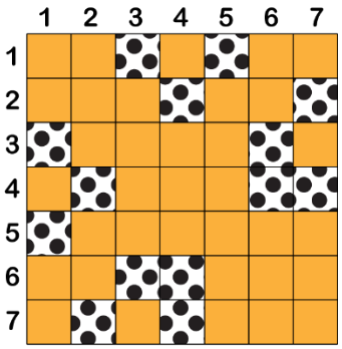


Kako bo izgledala prešita odeja, ki jo bo navdihnila ta pajkova mreža?

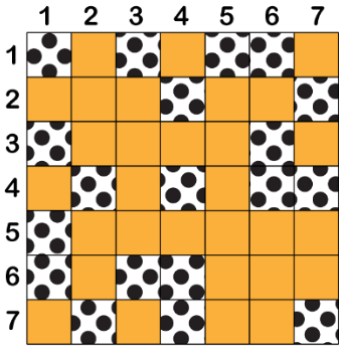
A.



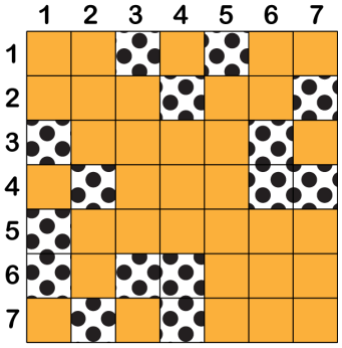
B.



C.



D.



Rešitev

Pravilen odgovor je A. V pajkovi mreži vidimo, da je mesto 1 z nitmi povezano z mesti 3, 5 in 6. Torej morajo biti v prešiti odeji v prvi vrsti trije kosi pikčaste tkanine, in sicer v stolpcih 3, 5 in 6. Vsi ostali kosi tkanine pa so enobarvni (torej v stolpcih 1, 2, 4 in 7). Tak vzorec ima le prešita odeja pod odgovorom A.

Preverimo še ostale dele, če so ustrezni. V pajkovi mreži povezuje nit tudi mesto 2 z mestoma 4 in 7. Torej ima prešita odeja v drugi vrsti pikčaste kose tkanine v stolpcih 4 in 7. Mesto 3 v pajčevini je z nitjo povezano z mestoma 1 in 6, zato so v prešiti odeji v tretji vrstici pikčasti kosi tkanine v stolpcih 1 in 6. Četrta vrsta odeje ima pikčaste kose tkanine v stolpcih 2, 6 in 7, saj je mesto 4 v pajčevini povezano z mesti 2, 6 in 7. Mesto 5 je v pajčevini povezano le z mestom 1, zato je v peti vrsti na odeji pikčast le prvi kos tkanine. Sledi šesta vrsta s tremi pikčastimi kosi tkanine, v stolpcih 1, 3 in 4, ker v pajkovi mreži najdemo mesto 6 povezano z nitjo z mesti 1, 3 in 4. Zadnja vrsta v prešiti odeji odraža mesto 7, ki je z nitjo povezano z mestoma 2 in 4, zato so pikčasti kosi tkanine v stolpcih 2 in 4. Odgovor A je dejansko pravilen.

Odgovor B ni pravilen, ker tej prešiti odeji manjka pikčasti kos tkanine v šestem stolpcu prve vrste in tudi v prvem stolpcu šeste vrste.

Odgovor C ni pravilen, ker so pikčasti kosi tkanine napačno postavljeni v prvi stolpec prve vrste, četrti stolpec četrte vrste ter sedmi stolpec sedme vrste.

Ob upoštevanju prvega pravila (postavitev pikčastega kosa tkanine) dobimo vedno odejo, ki ima simetričen vzorec glede na diagonalo (en pikčasti kos postavimo na mesto XY, drugega pa na mesto YX). Pri odgovoru D pa vzorec ni simetričen, saj je v šesti vrstici prvega stolpca pikčast kos tkanine, v šestem stolpcu prve vrstice pa najdemo enobarvnega.

Presedanje učencev

državno, 6. do 9. razred, srednja šola

V enem od razredov v bobrovi šoli sedi sedem učencev. Vsak od njih v roki drži zastavico s številko. Bobri sedijo v vrsti, eden za drugim.

Na začetku so se usedli, kot kaže slika.

Učitelj jih želi presesti tako, da bodo številke na njihovih zastavicah urejene po velikosti od 1 povsem spredaj do 7 v zadnji vrsti. Preseda jih lahko le tako, da se po dva hkrati zamenjata.

Na primer: ko se bober 3 in 1 zamenjata, to pomeni, da se bober 1 presede za mizo bobra 3 in bober 3 za mizo, kjer je prej sedel bober 1.

Najmanj koliko zamenjav je potrebnih, da učitelj doseže željeni vrstni red?

Rešitev

Potrebnih je pet zamenjav.

Za rešitev nalogo lahko uporabimo algoritem razvrščanje z izbiranjem. Ta algoritem razdeli vhodni seznam na dva dela: že urejen del seznama in neurejeni del seznama. Na začetku je urejen del seznama prazen, neurejeni del seznama pa je kar cel vhodni seznam. Algoritem najprej poišče najmanjši element v neurejenem delu seznama in ga zamenja z elementom, ki je na prvem mestu neurejenega dela seznama. S tem dobimo prvi element urejenega dela seznama, neurejen del seznama pa je za en element krajši. Po istem postopku v preostalem neurejenem delu seznama algoritem poišče najmanjši element in ga doda na konec urejenega dela seznama tako, da z njim zamenja prvi element v preostalem neurejenem delu seznama.

V našem primeru je postopek naslednji:

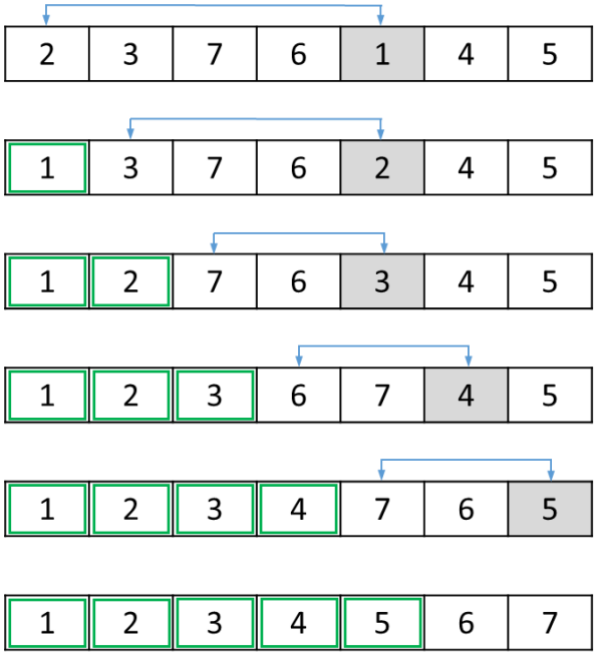
Najmanjši element v seznamu je 1, ki ga zamenja s prvim elementom seznama, to je 2. Po zamenjavi urejeni del seznama vsebuje število 1, preostali del seznama je še neurejen.

V preostalem delu seznama je najmanjši element 2, ki ga zato zamenjamo s prvim elementom v neurejenem delu seznama, to je 3. Urejeni del seznama zdaj vsebuje števili 1 in 2 (obkroženo z zeleno).

Najmanjši element v še neurejenem delu seznama je 3, zato ga zamenjamo s prvim elementom neurejenega dela seznama, to je 7.

Najmanjši element v še neurejenem delu seznama je 4, zato ga zamenjamo s prvim elementom neurejenega dela seznama, to je 6. Urejen del seznama zdaj vsebuje števila 1, 2, 3 in 4.

Najmanjši element v še neurejenem delu seznama je 5, zato ga zamenjamo s prvim elementom neurejenega dela seznama, to je 7. Urejeni del seznama zdaj vsebuje števila od 1 do 5, neurejeni pa števili 6 in 7. Ker sta že na svojih mestih, zamenjava ni več potrebna in ju lahko dodamo k urejenemu delu seznama v tem vrstnem redu.



Računalniško ozadje

Algoritmi za razvrščanje so zelo raznoliki in različno hitri. V skladu s pravili naloge bi lahko uporabili tudi algoritem urejanje z mehurčki, a bi ta za urejanje seznama potreboval več zamenjav in zato ni optimalen za rešitev naše naloge.

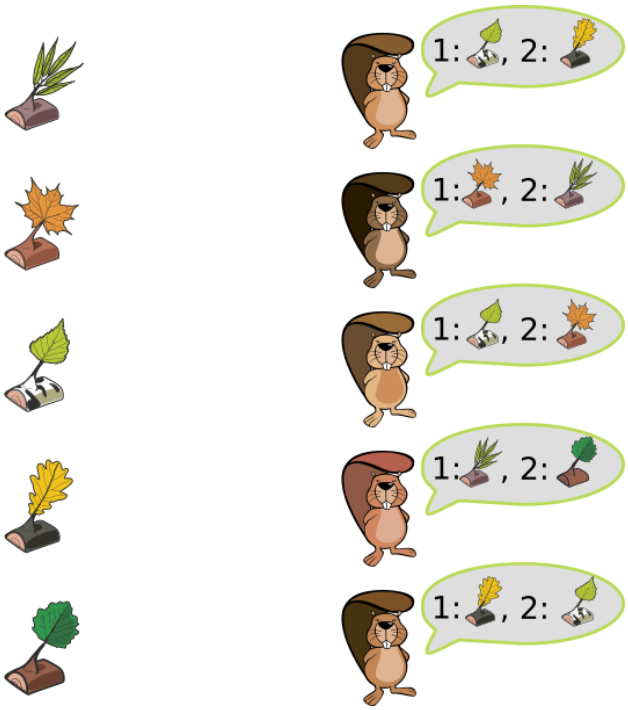
Želje 3

državno, 6. do 9. razred, srednja šola

Družina bobrov ima pet daril za svojih pet otrok. Vsak bober najprej pove, katero darilo si najbolj želi in katero je njegovo drugo najljubše. Darila morajo biti razdeljena tako, da:

- Čim več bobrov dobi najljubše darilo.
- Ostali dobijo drugo najljubše darilo.

Koliko bobrom se bo izpolnila prva želja?



Rešitev

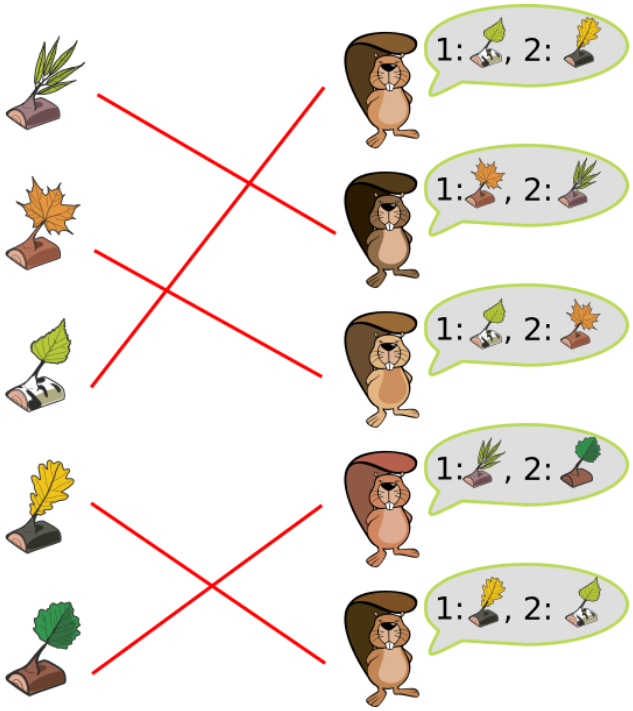
Dva bobra bosta dobila darili, ki sta ju navedla kot prvi želji, preostali trije bobri pa darila, ki so jih navedli kot svojo drugo željo (glej sliko).

Edini bober, ki si želi peto darilo, je četrti bober, zato ga dobi on.

Med preostalimi bobri si prvo darilo želi le drugi bober, zato dobi to darilo.

Med preostalimi tremi bobri si drugo darilo želi le tretji bober, zato dobi to darilo.

Ostaneta tretje in četrto darilo ter bobra, ki si v različnem prioritetnem redu želita ti dve darili. Tretje darilo (breza) dobi prvi bober, ki si ga najbolj želi, četrto darilo pa peti bober, saj je to njegovo najljubše. Prvemu in petemu bobru se tako izpolni prva želja.



Računalniško ozadje

Problem, ki ga rešujemo, je podoben problemu stabilnih zakonov. Takšne naloge se ne pojavljajo le v računalništvu temveč še marsikje. Leta 2012 je bila Nagrada Švedske banke za ekonomske vede v spomin Alfreda Nobela (ki ji pravijo tudi Nobelova nagrada za ekonomijo) podeljena raziskovalcem, ki so uporabili podoben postopek za resnične probleme s področja ekonomije.

Pretvarjanje

državno, 6. do 9. razred, srednja šola

Podatke lahko kodiramo in predstavimo v različnih oblikah. Katerokoli obliko lahko pretvorimo v neko drugo obliko. Bober Elif je izumil pretvorbeni stroj (→), ki uporablja 5 procesov za pretvorbo določenih oblik v druge.

Proces	Začetni simboli		Rezultat
!	*	→	@@
?	#	→	**
&	*@	→	#
^	##	→	*@@
$	@@@	→	

Vsak proces vpliva le na prvo ujemajočo skupino simbolov v zapisu: tako je na primer za vhod ## in proces ? rezultat **# (## → ? → **#) in ne ****. Če stroj v zapisu ne najde ustrezne skupine simbolov, na katere deluje proces, se ne spremeni nič in stroj začne izvajati naslednji proces. Zato lahko bober Elif doseže želen rezultat s pretvarjanjem simbolov, kolikorkrat želi.

Kakšen vrstni red procesov naj izbere bober Elif, če želi zapis simbolov ## pretvoriti v @@@?

A) ## → & → ^ → & → ? → ! → ? → $
B) ## → & → ? → & → ^ → ? → ! → $
C) ## → ^ → & → ? → & → ? → ! → ? → $
D) ## → ^ → & → ? → & → ? → ! → ! → ! → $

Rešitev

Pravilen odgovor je D. Spodaj so prikazani vsi koraki pretvorbe simbolov (po izvedbi posameznega procesa):

D) ## → *@@ → #@ → **@ → *# → *** → @@** → @@@@* → @@@@@@ → @@@

Ostali odgovori so napačni, saj po izvedbi posameznih procesov izgledajo zapisi simbolov tako:

A) ## → ## → *@@ → #@ → **@ → @@*@@ → @@*@@ → @@*@@

B) ## → ## → **# → **# → **# → **** → @@*** → @@***

C) ## → *@@ → #@ → **@ → *# → *** → @@** → @@** → @@**

Računalniško ozadje

Pogosta naloga računalnikov je tudi obdelava vhodnih podatkov po določenih pravilih. Ta pravila so v naši nalogi izvajali procesi. Procese lahko tudi verižimo, to je izvajamo enega za drugim, kar pomeni, da izhod prvega procesa pride na vhod naslednjega procesa. Takemu zaporedju procesov rečemo cevovod.

Keglji

državno, 6. do 9. razred, srednja šola

Ana in Borut sta v vrsto postavila pet kegljev. V eni potezi lahko podreta bodisi en kegelj bodisi dva sosedna keglja. V vsaki potezi lahko izbereta, katere keglje bosta podrla. Zmaga tisti, ki podre zadnji kegelj. Tako Ana kot Borut si lahko zagotovita zmago, če pametno izbereta keglje, ki jih bosta podrla v vsaki potezi.

Prva je na vrsti Ana. Kateri kegelj ali keglja naj podre v prvi potezi, da si zagotovi zmago?

A) B) C) D) E)

Rešitev

Odgovor A ni pravilen: če Ana podre prvi kegelj, Borut lahko zmaga, če v naslednji potezi podre tretji in četrti kegelj. Ana v naslednji potezi lahko podre bodisi drugi bodisi peti kegelj in tako Borutu ostane zadnji kegelj. Podobno se zgodi, če Ana najprej podre desni kegelj in nato Borut podre drugi in tretji kegelj …

Odgovor B ni pravilen: če Ana najprej podre prvi in drugi kegelj, v naslednji potezi pa Borut podre četrti kegelj, ponovno Ana lahko izbira le med bodisi tretjim bodisi petim kegljem, Borut pa podre zadnjega.

Odgovor C ni pravilen: če Ana najprej podre drugi kegelj, Borut pa v naslednji potezi podre tretjega in četrtega, lahko Ana v tretji potezi podre le en kegelj, prvi ali peti, in zadnjega podre Borut.

Odgovor D ni pravilen: če Ana podre drugi in tretji kegelj, Borut v naslednji potezi podre četrtega in ponovno ostane Ani na izbiro, da podre bodisi prvega ali petega, Borut pa podre zadnji kegelj.

Pravilni odgovor je E. Če Ana podre sredinski kegelj, razdeli keglje na dve ločeni skupini. Če Borut nato podre en kegelj (ali v levi ali v desni skupini), lahko Ana podre en kegelj v drugi skupini in tako ostaneta še dva keglja, ki nista sosedna, tako da Borut ne more podreti zadnjega. Če pa se Borut odloči v svoji prvi potezi podreti dva sosedna keglja, lahko Ana v zadnji potezi podre preostala dva keglja.

Računalniško ozadje

Kayles je kombinatorična igra, ki jo je navdihnil Henry E. Dudeney (*The Canterbury Puzzles and Other Curious Problems*, Heinemann, London 1907, Puzzle No. 73). Ime "kayles" izhaja iz francoskega "quilles", kar pomeni "kegljanje". Tako kot pri tej igri tudi v računalništvu pogosto za reševanje problemov uporabljamo znanje o kombinatoriki.

Premikanje krogel

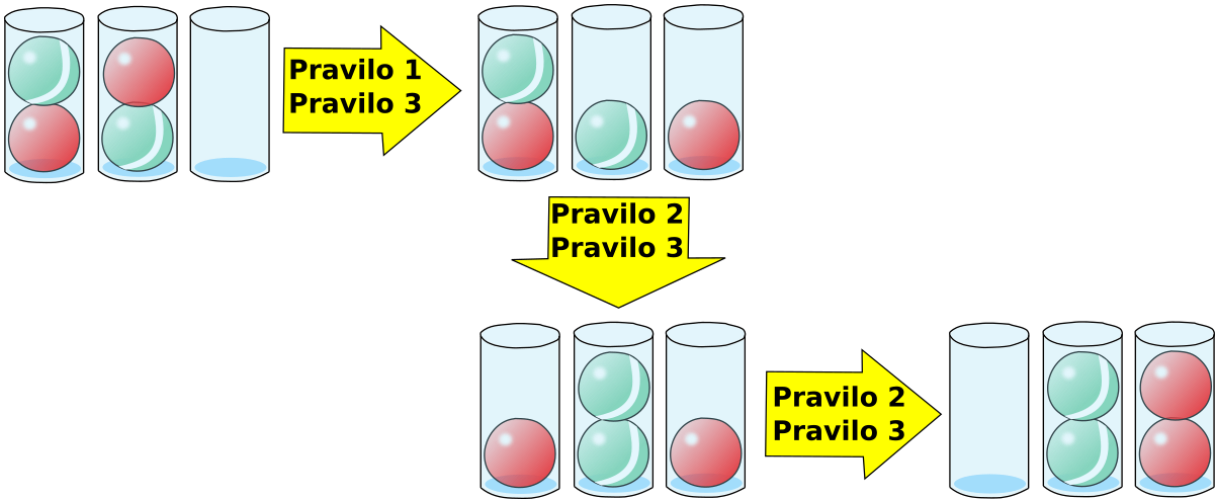
državno, 6. do 9. razred, srednja šola

Bobri se igrajo igro premikanja krogel iz valja v valj. Cilj igre je, da razvrstijo krogle iste barve v isti valj. Pravila igre so naslednja:

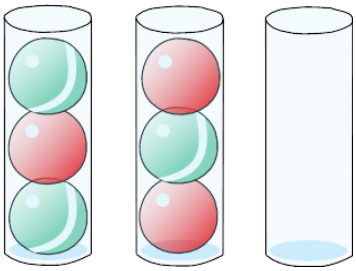
Pravilo 1: Kroglo lahko premakneš v prazen valj.

Pravilo 2: Če je v valju prostor, lahko vanj premakneš kroglo, ki je iste barve kot zgornja krogla v valju.

Pravilo 3: V eni potezi lahko premakneš le zgornjo kroglo v valju.



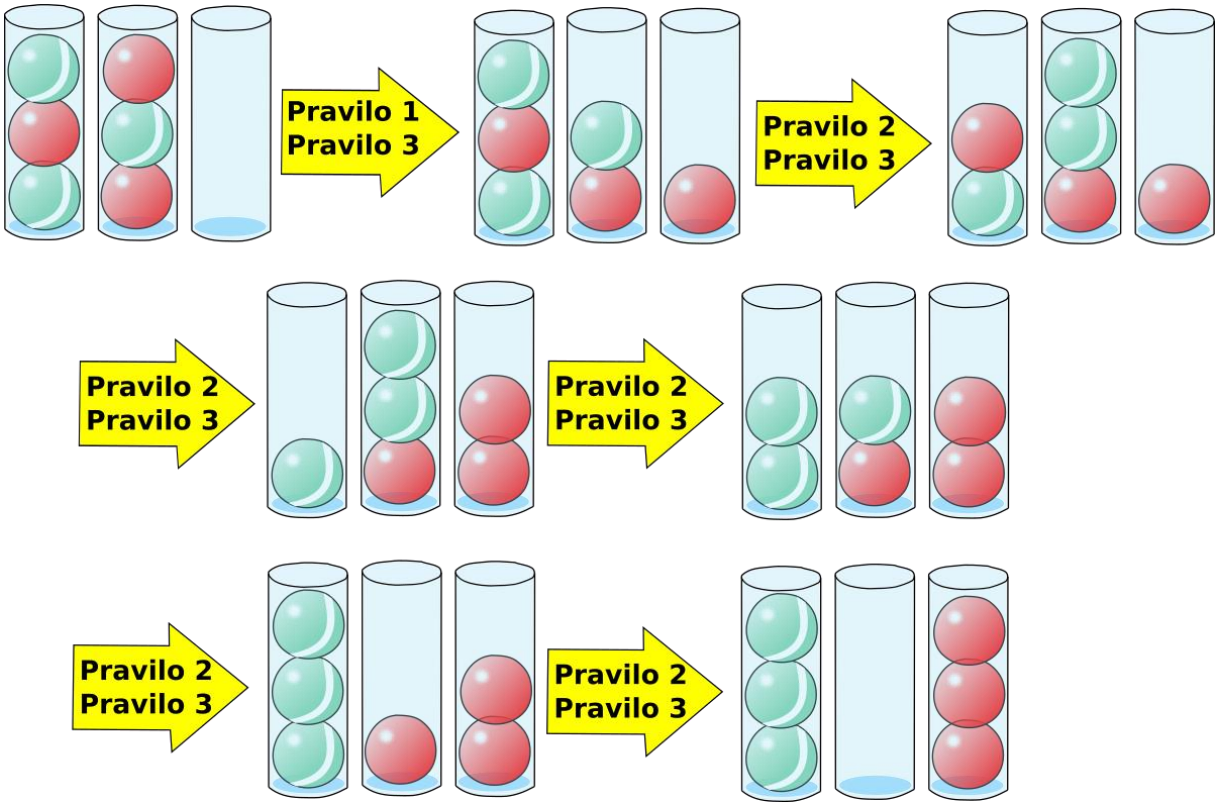
Najmanj koliko potez potrebuješ za zmago v spodnji igri?



Rešitev

Igro lahko zmagaš v šestih potezah.

Primer rešitve je prikazan spodaj. Možne so še druge rešitve, vendar vsaka od njih zahteva vsaj 6 potez.



Računalniško ozadje

Pri tej igri krogle v valje zlagamo kot na sklad. Na sklad se podatki vedno shranjujejo v vrstnem redu od prvega do zadnjega, s sklada pa jih jemljemo od zadnjega proti prvemu. V zgornji igri lahko tako kot pri skladu do krogle, ki smo jo prvo položili v valj dostopamo šele, ko smo z nje dvignili že vse krogle, ki smo jih zložili nanjo.

Kosilnica

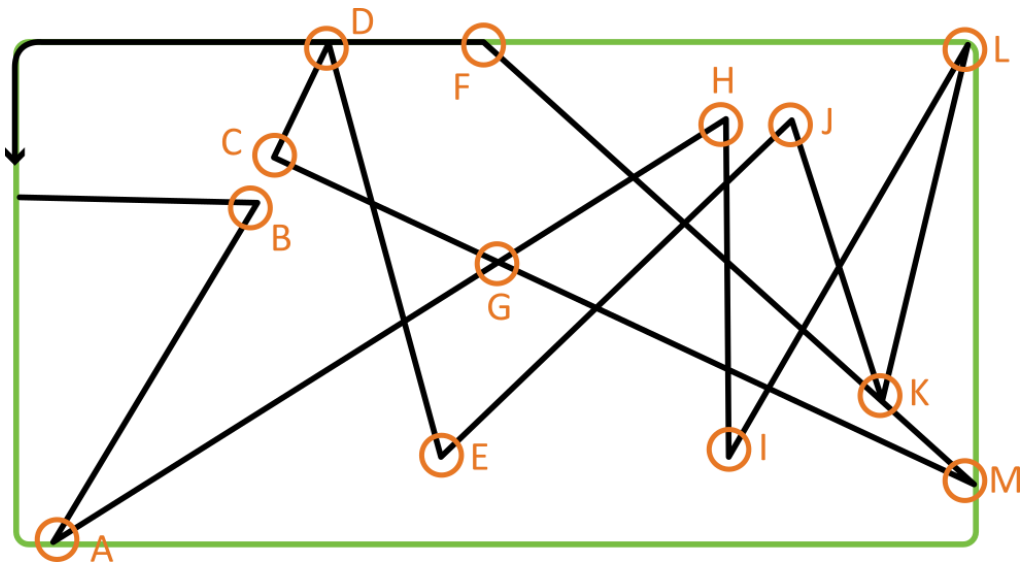
državno, 6. do 9. razred, srednja šola

Angela je na zemljevidu, ki beleži gibanje kosilnice, ugotovila, da je **med nočno košnjo** iz parka izginil kip.

Če si ogledaš zemljevid premikov kosilnice, lahko ugotoviš, kje v parku so postavljeni predmeti — drevo, klop, gredica rož in kip.

Robotska kosilnica se premika po naslednjih pravilih.

1. Na začetku košenja izbere naključno smer in košnjo nadaljuje naravnost.
2. Ko se zaleti v oviro ali rob parka, se obrne v novo naključno izbrano smer in nato pot nadaljuje naravnost.
3. Ko je baterija kosilnice skoraj prazna, se njeno obnašanje spremeni: ko doseže rob parka, se pomika po robu, dokler ne pride do polnilne postaje, kjer se ustavi.

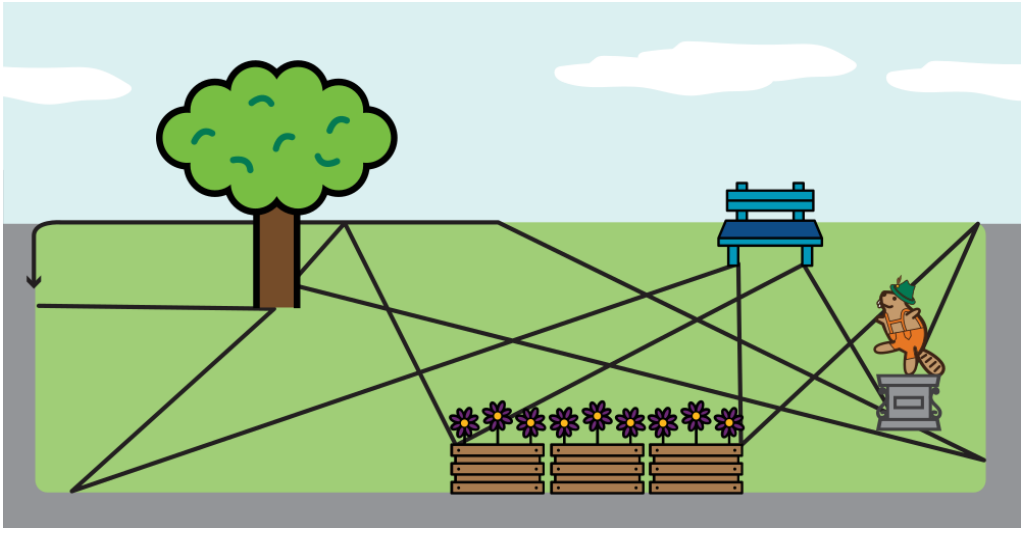


Pri kateri oznaki na zemljevidu bi moral biti postavljen pogrešani kip?

Rešitev

Pogrešani kip bi se moral nahajati v spodnjem desnem delu parka (oznaka K) , saj se je tam vedenje robotske kosilnice čez noč spremenilo. Spodnja slika prikazuje, kako je park zgledal pred krajo kipa.

Kmalu po začetku košnje je robotska kosilnica sredi trate spremenila smer. Kar seveda pomeni, da je naletela na oviro, drevo. Pot je nato nadaljevala v novi smeri, dokler ni dosegla roba parka, kjer je spremenila smer in nadaljevala košnjo vse do klopce. Robotska kosilnica se je med košenjem nato zaletela v cvetlično gredo, rob parka, kip, klop (ponovno), cvetlično gredo (ponovno), rob parka, drevo (ponovno) in nato košnjo nadaljevala skozi isto območje, na katerem je bil prej postavljen kip.



Ko je nato robotska kosilnica ob dotiku roba parka spremenila smer, je zopet svojo pot nadaljevala skozi območje, kjer je bila prej postavlja ovira (kip). Ko je kosilnica dosegla rob parka, je svojo košnjo nadaljevala ob meji parka vse do polnilne postaje.

Zemljevid nakazuje, da je robotska kosilnica šla skozi spodnji desni del parka dvakrat, ne da bi spremenila smer, medtem ko je v začetku košnje tam zadela v oviro in smer spremenila. Kar seveda pomeni, da je ovira med košnjo izginila.

Računalniško ozadje

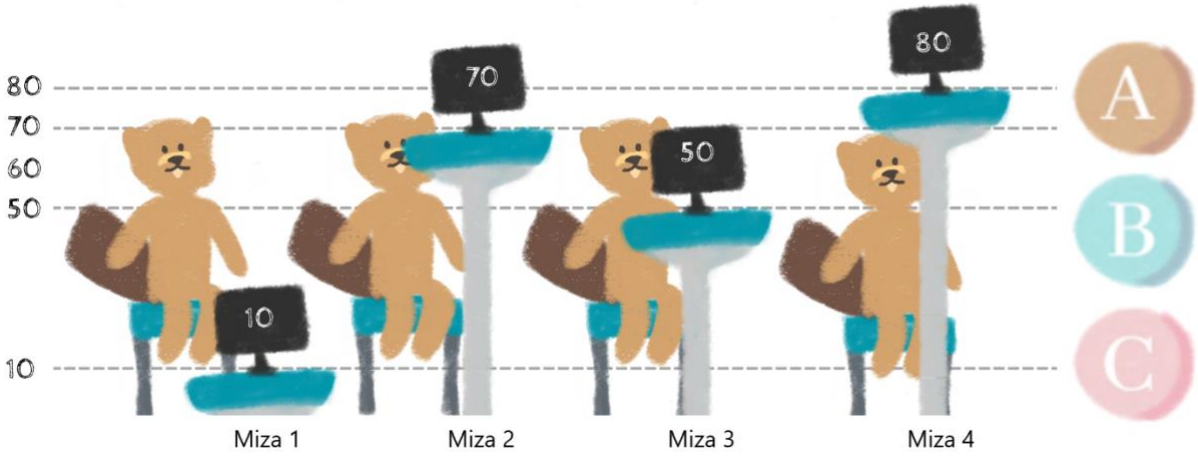
Izziv te naloge je najti povezavo med ovirami v parku in obnašanjem robotske kosilnice. Kosilnica se premika po določenih pravilih, na katera vpliva okolje. Tovrstno sklepanje je del računalniškega mišljenja in vključuje prepoznavanje vzorcev, predstavitev in interpretacijo podatkov.

Premikanje robotske kosilnice v nalogi upravlja zelo preprost program. Programska oprema komercialnih izdelkov je veliko bolj pametna — nekateri roboti se lahko premikajo okoli predmetov na travniku, namesto da bi le naključno spreminjali smer; nekateri celo ustvarijo digitalni zemljevid trate in se nato premikajo sistematično namesto naključno.

Težave z mizo

državno, 6. do 9. razred, srednja šola

Učenci prilagajajo višino svojih miz z uporabo električnega sistema. Priporočena višina učenčeve mize je 60 bobrovih enot. Žal je nekdo po gumbih polil vodo, kar je povzročilo spremembo višin vseh miz, kot je prikazano na spodnji sliki.



Pokvarila so se tudi stikala in sedaj kontrolni gumbi delujejo tako:

- Gumb A ob vsakem pritisku dvigne mize 1, 2 in 3 za 10 bobrovih enot.
- Gumb B ob vsakem pritisku spusti mize 2, 3 in 4 za 10 bobrovih enot.
- Gumb C ob vsakem pritisku dvigne mize 1, 3 in 4 za 10 bobrovih enot.

Kolikokrat bo morala gospodična Bober pritisniti gumbe A, B in C, da bo vse mize postavila na priporočeno višino 60 bobrovih enot.

A) 3-krat gumb A, 4-krat gumb B in 2-krat gumb C.
B) 4-krat gumb A, 5-krat gumb B in 1-krat gumb C.
C) 5-krat gumb A, 1-krat gumb B in 0-krat gumb C.
D) 2-krat gumb A, 4-krat gumb B in 6-krat gumb C.

Rešitev

Pravilen odgovor je A: 3-krat gumb A, 4-krat gumb B in 2-krat gumb C.

Če gospodična Bober trikrat pritisne gumb A, bodo mize 1, 2 in 3 visoke 40, 100 in 80 bobrovih enot. Če nato pritisne štirikrat gumb B, bodo mize 2, 3 in 4 visoke 60, 40 in 40 bobrovih enot. In ko na koncu pritisne dvakrat gumb C, se bodo še preostale tri mize — 1, 3 in 4 — nastavile na 60 bobrovih enot, kar pomeni, da bo cilj dosežen.

Možnost B je napačna, saj bi gospodična Bober s temi ukazi nastavila ustrezno višino le mizama 1 in 2, ne pa tudi 3 (50 bobrovih enot) in 4 (40 bobrovih enot).

Možnost C je tudi napačna, ker gospodična Bober s temi ukazi zagotovi ustrezno višino le mizi 1. Miza 2 je ob koncu visoka 110, miza 3 90 in miza 4 70 bobrovih enot.

Tudi možnost D je napačna, saj so končne višine miz 90 (miza 1), 50 (miza 2), 90 (miza 3) in 100 (miza 4) bobrovih enot. Kar pomeni, da ni nobena od miz nastavljena na priporočeno višino.

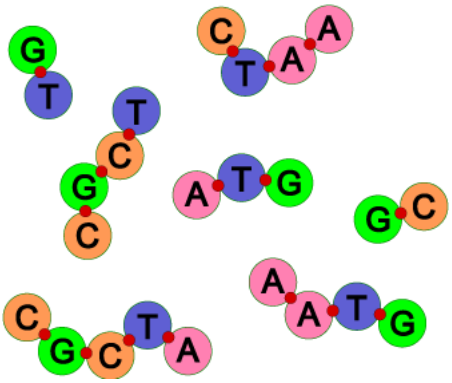
Računalniško ozadje

V nekaterih primerih pri podajanju ukazov računalniku določen ukaz vpliva na več podatkov kot smo želeli. V tem primeru je potrebno najti ustrezen nabor ukazov, da bo program podatke spremenil natanko tako, kot želimo.

Dekodiranje genoma

državno, 6. do 9. razred, srednja šola

Molekula DNK je veriga nukleotidov, vsak nukleotid pa vsebuje eno od štirih dušikovih baz: adenin (A), citozin (C), gvanin (G) ali timin (T). Zaporedje lahko beremo v eno ali drugo smer: na levi sliki je torej lahko ATG ali GTA.



Znanstvenik je naredil več kopij nekega zaporedja. Te kopije so razpadle na koščke in nato so se nekateri koščki izgubili. Ostali so koščki na desni sliki.

Zdaj želi rekonstruirati prvotno molekulo. Uporabiti mora vse koščke, pri čemer lahko dva stakne v enega; tako lahko, na primer, koščka TGAACT in GGGTCA stakne v TGACTGGG, saj se ACT pojavi v obeh.

Koliko nukleotidov ima torej najkrajša možna veriga, sestavljena iz koščkov na tej sliki?

Rešitev

Najkrajša možna molekula vsebuje 8 nukleotidov.

To je molekula CGCTAATG oziroma GTAATCGC.

Hitro lahko preverimo, da ta molekula vsebuje vse navedene delčke. Ker se delčka CGCTA (dolžine 5) in AATG (dolžine 4) ujemata le v enem nukleotidu (A), je 8 nukleotidov tudi najmanjše možno število.

Računalniško ozadje

Problem, ki smo ga reševali v tej nalogi, se pojavi ob branju zaporedja, ki je zapisano v genih neke celice. Zato s polimerazno verižno reakcijo (*polymerase chain reaction, PCR* — ti je ta kratica od nekod znana?) najprej namnožijo genski material, ki bi ga radi sekvencirali. Nato z določenim postopkom preberejo dobljene delčke. Te je potem potrebno sestaviti v celotno zaporedje, kot si to počel v tej nalogi.

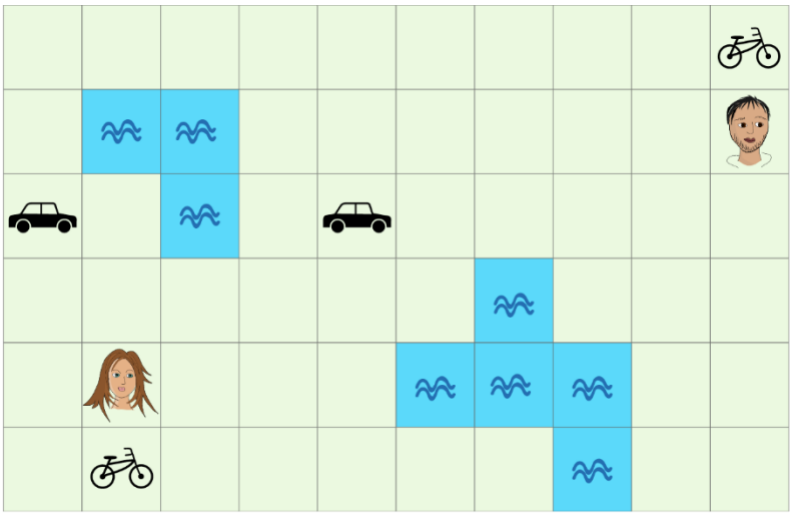
Ko so se pred tridesetimi leti lotili sekvenciranja človeškega genoma, je bilo sestavljanje teh koščkov eden največjih računskih podvigov, dandanes pa je sekvenciranje genomov rutinsko.

Ob izbruhu epidemije covid-19 so zato brž prebrali genom virusa, ugotovili, kateri del opisuje konico, ki jo prepoznavajo naša protitelesa, in s pomočjo tega znanja izjemno hitro pripravili tudi učinkovita cepiva. Ta pokažejo našemu telesu, kako je videti ta virus, in ga pripravijo na spopad z njim. Zasluge za to imamo torej tudi računalnikarji!

Nujno srečanje

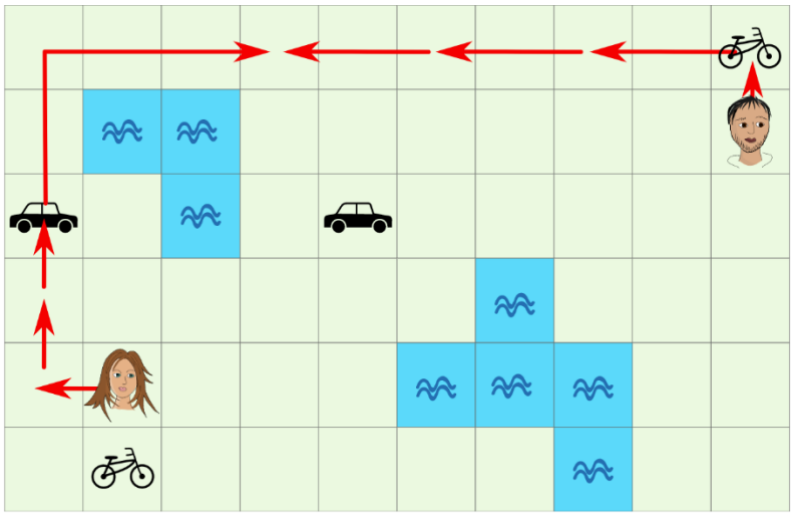
državno, 6. do 9. razred, srednja šola

Ana in Bojan se morata nujno čimprej srečati. Njuni trenutni lokaciji sta prikazani na zemljevidu. Lahko gresta peš z enega kvadrata vodoravno ali navpično na sosednji kvadrat, za kar potrebujeta 1 minuto. Če prideta do kolesa ali avta, ju lahko uporabita za hitrejše potovanje — s kolesom v 1 minuti premagata 2 kvadrata, z avtom pa 5 kvadratov. Preko vode pa ne moreta potovati.



Najmanj koliko minut potrebujeta, da se oba dobita na istem kvadratu?

Rešitev



Za srečanje potrebujeta najmanj 4 minute.

Ena od možnih optimalnih poti za vsakega od njiju je prikazana na sliki.

Druga možnost bi bila, da gre Ana najprej do kolesa na sosednjem kvadratu in odkolesari do avta (in nadaljuje, kot je prikazano na sliki).

Pa je to res najkrajši možni čas? Bi se lahko srečala v 3 minutah? Razmislimo takole: čeprav lahko Ana v treh minutah doseže avto na levi, nato nima več časa, da bi se z njim kamorkoli odpeljala. Bojan pa tudi ne more v 3 minutah priti na isto mesto. Avta na desni pa nobeden od njiju ne more doseči v treh

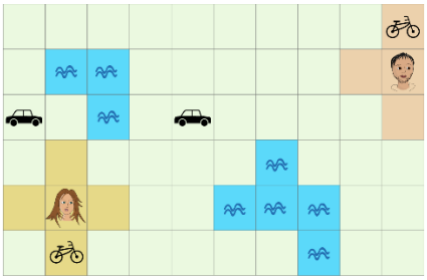
minutah. Torej avta v tem primeru nista uporabna in ju lahko odstranimo iz zemljevida. Ana in Bojan sta med seboj oddaljena za več kot 6 kvadratov, torej več kot 6 minut peš. Torej potrebujeta kolo. Ker sta oddaljena za več kot 9 kvadratov, morata oba uporabiti kolo. Vendar oba porabita 1 minuto, da prideta do kolesa, tako da jima ostaneta le 2 minuti za kolesarjenje. V 2 minutah lahko vsak od njiju prekolesari 4 kvadrate. Ker sta obe kolesi na zemljevidu oddaljeni za več kot 8 kvadratov, se niti z uporabo kolesa ne moreta srečati v 3 minutah.

Računalniško ozadje

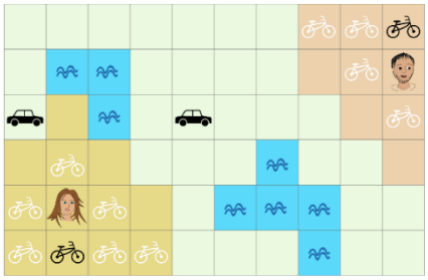
Pri reševanju naloge smo najprej poiskali rešitev, ki je videti najboljša, nato pa smo z razmislekom še preverili, če je ta rešitev res najkrajša.

Lahko pa bi se iskanja rešitve lotili tudi na bolj sistematičen način, kot to delajo računalniki. Navadno se pri podobnih problemih uporabi *algoritem iskanja v širino*, s katerim sistematično pregledujemo vse možne rešitve. V naši nalogi bi to naredili na naslednji način:

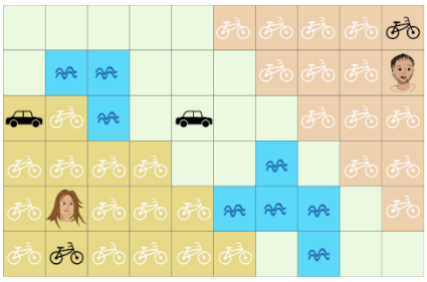
1. Najprej na zemljevidu označimo vse kvadrate, ki jih lahko Ana in Bojan dosežeta v eni minuti.



2. V naslednjem koraku označimo vse kvadrate, ki jih lahko dosežeta v eni minuti z vsake pozicije, ki smo jo označili v predhodnem koraku. Pri tem zabeležimo tudi podatek, katero (najhitrejše) transportno sredstvo je bilo uporabljeno. Tako so označeni vsi kvadrati, ki jih lahko dosežeta v (največ) 2 minutah.

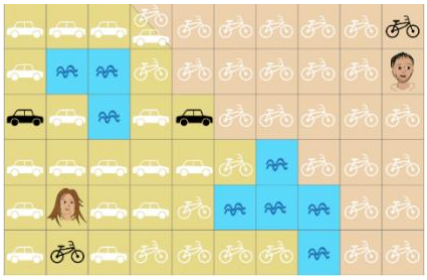


3. V naslednjem koraku spet označimo kvadrate, ki jih lahko dosežeta s kvadratov iz predhodnega koraka. S tem so označeni vsi kvadrati, ki jih lahko dosežeta v (največ) 3 minutah.



Ker se Anino označeno območje nikjer ne prekriva z Bojanovim označenim območjem, to pomeni, da se prijatelja ne moreta srečati v 3 minutah.

4. Naredimo še en korak in označimo vse kvadrate, ki jih lahko dosežeta v 1 minuti s kvadratov, označenih v predhodnem koraku.

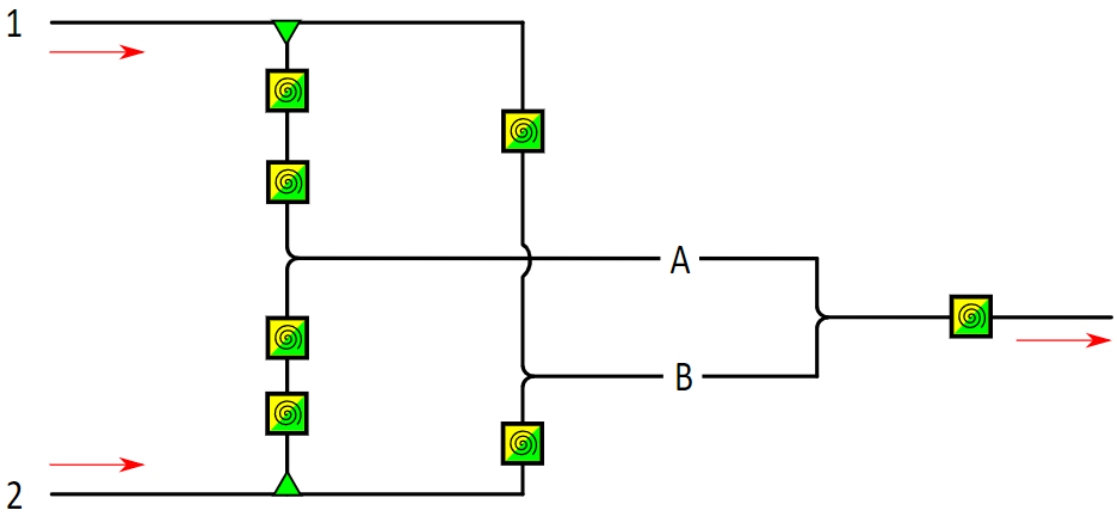


Vidimo, da se sedaj obe označeni območji prekrivata v enem kvadratu (četrti kvadrat v prvi vrsti), kar pomeni, da se na tem kvadratu lahko srečata oba prijatelja po 4 minutah.

Na podoben način dela tudi program za navigacijo, ki išče najhitrejšo pot do cilja, pri tem pa pazi, da za pot izbira le ceste in se izogne rekam in hribom. Tako sistematično iskanje najboljše rešitve nam včasih vrne rešitev, ki na prvi pogled ni očitno najhitrejša. Nekoliko daljša pot naokoli, ki pa ima manj semaforjev, je lahko hitrejša kot direktna (krajša) pot — na drugi konec mesta, kot je Ljubljana, hitreje pridemo po obvoznici okrog njega kot skozi središče mesta.

Hočem pistacijo! državno, 6. do 9. razred, srednja šola

V slaščičarni imajo nov stroj za izdelavo sladoleda. V stroj damo dve vrsti sestavin za okus sladoleda: pistacija (zelena) in vanilja (rumena). Sestavine z vhoda (označena 1 in 2) potujejo skozi cevi stroja od leve proti desni, dokler na izhodu ne dobimo sladoleda.



Stroj sestavljajo naslednji deli:

Ta naprava spremeni okus sladoleda — vaniljo spremeni v pistacijo ali pistacijo v vaniljo.

Če v to napravo pride sladoled z okusom pistacije, spremeni smer in gre naprej v tisti smeri, v katero kaže naprava (v primeru na sliki je to gor). Sladoled, ki ni pistacija, ne spremeni smeri.

Na tem mestu se cevi ne stikajo, ampak gre ena cev preko druge.

Danes lahko uporabimo le eno dvojno vrečko sestavin za okus sladoleda, zato moramo na oba vhoda (1 in 2) stresti isto sestavino. Torej na obeh vhodih uporabimo ali vaniljo ali pa pistacijo.

Kam moramo dodati napravo  , da bomo na izhodu dobili pistacijo, ne glede na to, kateri okus uporabimo na vhodu?

A) A
B) B
C) A in B
D) niti A niti B

Rešitev

Pravilen odgovor je C: dodatno napravo moramo namestiti na mestih A in B.

Ker naprava spremeni okus sladoleda, bosta dve zaporedni napravi spremenili okus spet nazaj v začetni okus na vhodu.

Če so vhodne sestavine okusa pistacije, potem mora biti v stroju sodo število naprav za spremembo okusa, da dobimo na izhod spet pistacijo. Če pa na vhod damo vaniljo, mora ta preko lihega števila naprav za spremembo okusa, da na izhod dobimo pistacijo.

Če na vhoda 1 in 2 damo pistacijo, sladoled v stroju spremeni smer in gre po levi navpični cevi ter skozi A do izhoda. Na poti gre skozi tri naprave za spremembo okusa, zato moramo na mesto A dodati še eno napravo za spremembo okusa, da bodo skupaj štiri (sodo število) in s tem dobimo pistacijo tudi na izhodu.

Če pa na vhoda 1 in 2 damo vaniljo, sladoled potuje naravnost skozi napravo za spremembo smeri (po vodoravni cevi) in naprej po desni navpični cevi ter skoti B do izhoda. Na poti gre skozi dve napravi za spremembo okusa, zato moramo na mesto B dodati še eno napravo za spremembo okusa, da bodo skupaj tri (liho število) in s tem dobimo pistacijo tudi na izhodu.

Računalniško ozadje

Shema, ki jo moramo v tej nalogi dopolniti, je sinhronizacijski avtomat, ki ne glede na podan okus na vhodu vedno pripravi okus pistacije na izhodu.

Na shemo lahko pogledamo tudi kot na logični diagram, a smo tu namesto klasične binarne ali Boolove logike uporabili trovrednostno ali ternarno logiko. Slednja pozna poleg vrednosti *resnično* (*true*, T ali 1) in *neresnično* (*false*, F ali 0) tudi stanje nedoločenosti, to je vrednost *nedoločeno* (*unknown* ali U). V cevi imamo namreč lahko sladoled z okusom vanilje, sladoled z okusom pistacije ali pa je cev prazna. Zato je tudi naprava za spreminjanje okusa drugačna kot navadni logični NE, saj spremeni okus sladoleda, ki je v cevi, medtem ko ne naredi nič, če v cevi ni sladoleda: NE(F) = T, NE(T) = F, NE(U) = U.

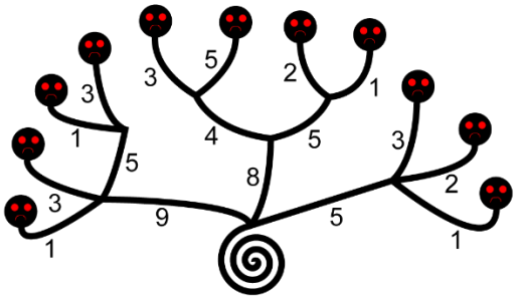
Druga naprava za spremembo smeri pistacije, ki je prikazana kot zelen trikotnik, ima en vhod in dva izhoda. Torej lahko njeno delovanje opišemo z dvema ternarnima logičnima operacijama (označimo ju s P in R): P(F) = U, P(T) = T, P(U) = U ter R(F) = F, R(T) = U, R(U) = U.

Na podlagi ternarne logike lahko sestavimo tudi računalnike (pravzaprav so jih nekaj že naredili), ki so v nekaterih pogledih, predvsem v povezavi s teorijo informacij, boljši od računalnikov, ki temeljijo na binarni logiki.

Herkul in Hidra

8. in 9. razred, srednja šola

Herkul se bori proti Lernajski Hidri, večglavi kači podobni vodni pošasti. Herkul uporablja meč, s katerim lahko odseka Hidri glavo. Vendar pa za nekatere dele Hidrinega vratu potrebuje tudi več zamahov, da jih preseka. Herkul pozna skrivnost, koliko zamahov je potrebnih, da preseka posamezni Hidrin vrat. Število je zapisano na sliki.

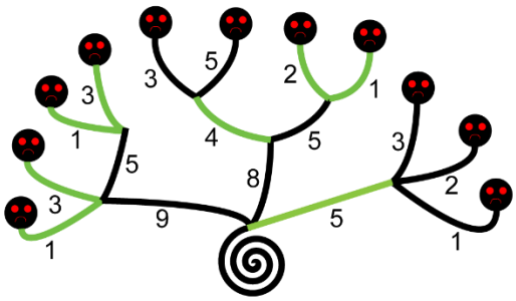


Kakšno je najmanjše število zamahov, ki jih potrebuje Herkul, da odseka vse Hidrine glave?

Rešitev

Herkul potrebuje najmanj 20 zamahov, da lahko odseka vse Hidrine glave.

Slika prikazuje, kako dobimo optimalen rezultat (zelena krivulja ponazarja vrat, ki naj ga Herkul preseka).



Hidro si lahko predstavljamo kot drevo, ki »raste« iz korena. Poiskati moramo najmanjšo vsoto uteži povezav, ki ločijo vse liste od korena. To je standardni problem najmanjšega prereza grafa in ga lahko rešimo s pomočjo različnih algoritmov (npr. Ford-Fulkesronov algoritem).

Obstaja pa tudi bolj preprost algoritem za iskanje najmanjšega prereza grafa med listi in korenom drevesa. Začnemo pri listih in se premikamo proti korenu ter za vsako povezavo preračunamo, ali je optimalno prerezati to povezavo ali ne.

Poglejmo delovanje tega algoritma na našem primeru. Začnemo pri listih. Če bi vsak list odrezali od njegovega predhodnika, bi dobili 1 + 3 + 1 + 3 + 3 + 5 + 2 + 1 + 3 + 2 + 1.

Nato se pomaknemo eno povezavo proti korenu. Pri vsakem premiku lahko ohranimo predhodno izračunano vrednost prereza ali pa jo nadomestimo s prerezom nove povezave (vzamemo, kar ima manjšo vsoto uteži). Pri drugi iteraciji imamo tako: 1 + 3 + min(5, 1 + 3) + min(4, 3 + 5) + min(5, 2 + 1) + min(5, 3 + 2 + 1), oziroma 1 + 3 + 4 + 4 + 3 + 5.

Tretja iteracija nam da: min(9, 1 + 3 + 4) + min(8, 4 + 3) + 5, oziroma 8 + 7 + 5. Ker smo sedaj že prišli do korena, je torej končna rešitev 8 + 7 + 5 = 20.

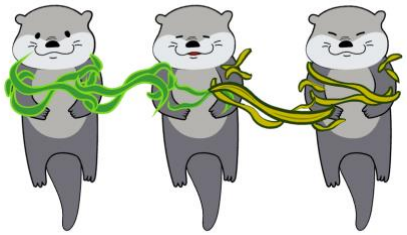
Računalniško ozadje

Naloga predstavlja primer vzvratnega rezanja dreves. To je pomemben postopek, ki ga uporabljamo pri optimizaciji podatkov ali modelov, predstavljenih z drevesi.

Popoldanski dremež

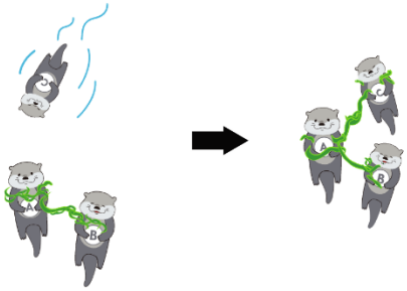
8. in 9. razred, srednja šola

Ko se srečata dve vidri, se povežeta z ovijanjem morske trave, zato da bosta ostali skupaj tudi med popoldanskim dremežem. Vidri, ki sta že (posredno) povezani prek drugih vider, pa se ne ovijeta s travo, da ne bi prišlo do vozlov.

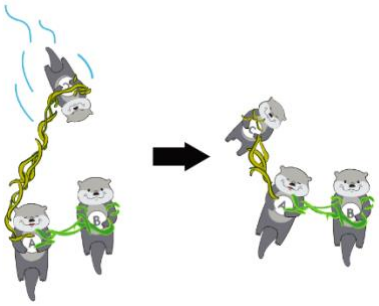


Če se na primer srečajo vidre A - B, A - C ter B - C, se zgodi naslednje:

Vidra B sreča vidro A. Obe se povežeta z ovijanjem morske trave.



Vidra C sreča vidro A. Obe se ovijeta z morsko travo.



Vidra C sreča vidro B. Ker sta že povezani preko vidre A, se ne ovijeta z morsko travo.

Vidre se srečajo v naslednjem vrstnem redu: A - B, A - C, B - C, D - E, A - E, D - F, A – F.

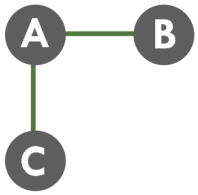
Koliko morskih trav je na koncu ovitih okoli vidre A?

Rešitev

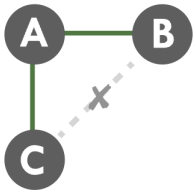
Okoli vidre A so ovite tri morske trave.

Rešitev poiščemo tako, da pogledamo, kako se vidre srečujejo in povezujejo.

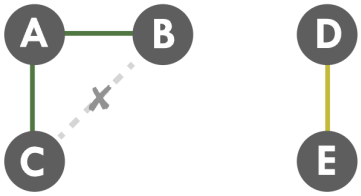
A sreča B. Povežeta se z morsko travo. Torej ima v tem trenutku vidra A ovito **eno** morsko travo.



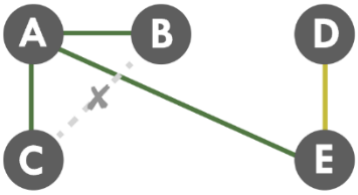
A sreča C in povežeta se z morsko travo. V tem trenutku sta okoli vidre A oviti **dve** morski travi.



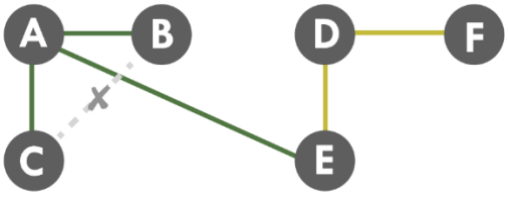
B sreča C. Ker sta B in C že povezani preko A, se ne povežeta z morsko travo.



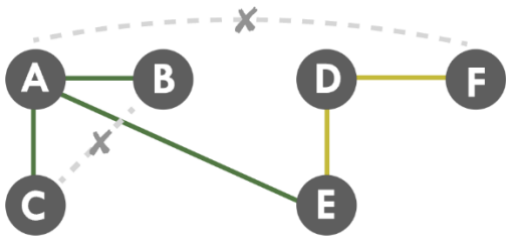
D sreča E. Povežeta se z morsko travo.



A sreča E. Povežeta se z morsko travo. V tem trenutku so okoli vidre A ovite **tri** morske trave.



D sreča F. Povežeta se z morsko travo.



A sreča F. Ker sta A in F že povezani preko E in D, se ne povežeta z morsko travo.

Torej, okoli vidre A so ovite le tri morske trave.

Računalniško ozadje

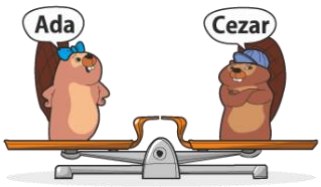
Disjunktna množica je podatkovna struktura, ki hrani zbirko disjunktnih (neprekrivajočih se) množic. Nudi operacije dodajanja novih množic in združevanja množic. V tej nalogi smo združili dve množici tako, da smo okoli dveh vider ovili morsko travo.

Disjunktna množica je pomembna podatkovna struktura pri implementaciji Kruskalovega algoritma za iskanje minimalnega vpetega drevesa v grafu. Disjunktne množice se uporabi, da se izogne ciklom v izbranih povezavah, kar je enako kot izogibanje vozlom v tej nalogi.

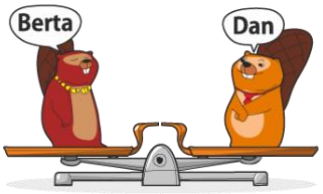
Štirje bobri se igrajo na tehtnici. Posneli so veliko fotografij. Tri so spodaj.

Katera slika bi tudi lahko bila med posnetimi fotografijami?

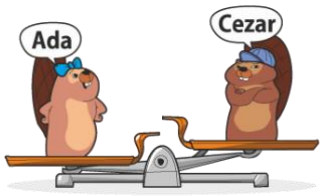
A)



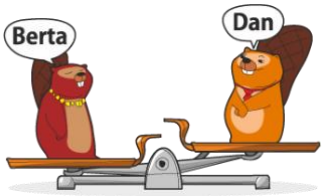
B)



C)



D)



Rešitev

Možna je le slika C, na kateri je Cezar lažji od Ade. Ostalih treh slik niso mogli posneti, ker ne morejo biti resnične.

Nalogo lahko rešimo z deduktivnim sklepanjem.

Če bi bila resnična slika A, bi to pomenilo, da Ada tehta toliko kot Cezar. Potem mora tudi Berta tehtati toliko kot Dan (kar je razvidno iz zgornje desne slike). V tem primeru bi morala biti tehtnica na spodnji sliki v ravnovesju (Ada + Dan = Berta + Cezar). Ker slika ne kaže ravnovesja, je naša predpostavka napačna, torej slika A ne more biti resnična.

Če bi bila resnična slika B, bi to pomenilo, da Berta tehta toliko kot Dan. Tudi tu je podobno kot v prejšnjem primeru: glede na zgornjo desno sliko bi morala Ada tehtati enako kot Cezar in na spodnji sliki bi morala biti tehtnica v ravnovesju. Torej je tudi ta predpostavka napačna in slika B ni resnična.

Če bi bila resnična slika D, bi morala Berta tehtati več kot Dan (Berta > Dan). Če ponovno pogledamo zgornjo desno sliko, bi potem moral Cezar tehtati več kot Ada (Cezar > Ada), da bi bila tehtnica v ravnovesju. Vendar pa v tem primeru spodnja slika ne bi mogla prikazovati, da Berta in Cezar skupaj tehtata manj kot Ada in Dan. Pravzaprav bi moralo biti ravno obratno. Torej je tudi ta predpostavka napačna in slika D ni resnična.

Če bi bila resnična slika C, bi Cezar tehtal manj kot Ada (Cezar < Ada), kar se ujema s stanji na vseh treh fotografijah. Torej je pravilen odgovor C.

Računalniško ozadje

V matematiki – pa tudi v računalništvu – pogosto govorimo o relacijah in njihovih lastnostih. Relacija "biti gostejši" je primer tranzitivne relacije: če je A težji ob B in B težji od C, mora biti tudi A težji od C. S tem povezana naloga v računalništvu je topološko urejanje, kjer poznamo določeno relacijo med (nekaterimi) objekti, naša naloga pa je, da jih uredimo glede na to relacijo.

Rokoborba

8. in 9. razred, srednja šola

Janko Bobr trenira za profesionalnega rokoborca. Osvojil je šest različnih **položajev**:

- Lying (P1)
- Standing (P2)
- Running (P3)
- Against the ropes (P4)
- In the corner (P5)
- Top rope (P6)

Njegov trener ga bo naučil še nekaj **premikov**, vsak premik pa ima tudi seznam položajev, iz katerih ga lahko izvedemo. Janko bi se želel naučiti premik za vsak položaj, a bi se hkrati želel naučiti čim manj različnih premikov, saj bo le tako imel dovolj časa za vajo vsakega od njih. Premiki in položaji, iz katerih jih lahko izvede, so naslednji:

- M1: Crossbody – Running, Top Rope (P3, P6)
- M2: Suplex – Standing, Top Rope (P2, P6)
- M3: Clothesline – Standing, Running, Top Rope, Against the ropes (P2, P3, P4, P6)
- M4: Back Elbow – Standing, In the corner, Against the ropes (P2, P4, P5)
- M5: Armbar – Standing, Lying (P1, P2)
- M6: Running Splash – Running (P3)

Kakšno je najmanjše število premikov, ki se jih mora naučiti Janko, da bo lahko izvedel premik iz katerega koli položaja?

Rešitev

Janko se mora naučiti 3 premike. Če podrobneje pogledamo vse premike, vidimo, da iz položaja *Lying* lahko izvedemo le premik *Armbar* (M5), iz položaja *In the corner* pa le premik *Back Elbow* (M4). Oba premika (M4 in M5) pa pokrijeta le štiri različne položaje, zato se bo Janko moral naučiti vsaj še en dodaten premik. Manjkajoča dva položaja, *Running* in *Top rope*, pokrije premik *Crossbody* (M1) ali pa premik *Clothesline* (M3). Torej se mora Janko naučiti vsaj 3 premike, to pa so lahko premiki M1, M4 in M5 ali pa M3, M4 in M5.

Računalniško ozadje

Prikazan problem se imenuje problem iskanja najmanjše dominantne množice. Imamo šest množic, vsaka predstavlja enega od premikov in vsebuje položaje, iz katerih lahko izvedemo dani premik. Naloga je, da izberemo najmanjše število množic (premikov) tako, da bo unija izbranih množic (vsi položaji iz katerih lahko izvedemo te premike) vsebovala vse elemente (položaje). Povedano drugače, naloga je pokriti vse elemente z najmanjšim številom množic. V našem primeru smo bili pri izbiri množic (premikov) omejeni z dvema podanima elementoma (položajema), kar je olajšalo izbiro množic, ki pokrivajo določen seznam elementov.

Tatovi umetnin

8. in 9. razred, srednja šola

Bevo Ladrone je šef mafijske tolpe, specializirane za krajo umetniških slik. Vsako noč kadi cigare v sobi, v katero kurirji prenašajo ukradene slike. Tam jih popišejo, preden jih odnesejo naprej v skladišče. Vsak kurir, ki prinese sliko, jo odloži na vrh kupa in vsak kurir, ki odnese sliko, odnese sliko z vrha kupa.

Preden se je Bevo izpridil v mafijca, je bil računalnikar, zato so njegove evidence skrbno zbrane v dveh tabelicah.

Prinašanje slik

Čas	Slika
23:40	Bobri na travi
00:15	Srečni bober
00:55	Sonce in luna
01:30	Čarobni gozd
02:18	Breza in hrast
03:10	Močvirna romanca

Odnešanje slik

Čas	Kurir
00:25	A
01:35	C
02:35	A
02:40	B
03:20	C
03:35	D

Zjutraj je Bevo odkril, da je slika "Sonce in luna" izginila. Kurir, ki bi jo moral po popisu odnesti v Bevovo skladišče, jo je ukradel!

Bevo bo poslal svojega najzvestejšega bodybuilderja na prijateljski pogovor ob čaju ali kavi z enim od kurirjev. S katerim? Kateri kurir ima pogum krasti bossu?

Rešitev

Sliko Sonce in luna je odnesel kurir B.

Pri transportu slik sta dva pomembna dogodka: nekdo postavi sliko na kup slik in nekdo vzame sliko s kupa slik. Oba dogodka sta zabeležena v evidencah. Iz obeh tabel v nalogi lahko sestavimo novo tabelo, ki oba dogodka prikazuje časovno urejena, poleg tega pa še trenutno stanje kupa slik.

Čas	Dogodek	Kup slik
23.40	prispe slika Bobri na travi	Bobri na travi
00.15	prispe slika Srečni Bober	Srečni bober Bobri na travi
00.25	A odnese Srečni bober	Bobri na travi
00.55	prispe slika Sonce in luna	Sonce in luna Bobri na travi
01.30	prispe slika Čarobni gozd	Čarobni gozd Sonce in luna Bobri na travi
01.35	C odnese Čarobni gozd	Sonce in luna Bobri na travi
02.18	prispe slika Breza in hrast	Breza in hrast Sonce in luna Bobri na travi
02.35	A odnese Breza in hrast	Sonce in luna Bobri na travi
14.40	B odnese Sonce in luna	Bobri na travi

... in tu se lahko ustavimo ter obiščemo kurirja B.

Računalniško ozadje

V nalogi se igramo s podatkovno strukturo sklad oziroma vrsta LIFO – last in, first out. Kot pove njeno angleško ime, je značilnost te vrste, da je prvi na vrsti tisti, ki je prišel zadnji. Pri reševanju je bilo potrebno simulirati obnašanje te vrste na zbranih podatkih.

Skrivni dnevnik

državno, 8. in 9. razred, srednja šola

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	R	S	T	U
V	Z	Č	Š	Ž

Petra in Jana sta našli skrivni dnevnik sošolke Lucije. A na njuno žalost je Lucija zapise v dnevnik zakodirala z uporabo vodoravnih in navpičnih črt. Dekleti sta odkrili, da si je pri kodiranju pomagala s tabelo črk, ki je bila zapisana na prvi strani dnevnika:

Hitro sta tudi opazili, da je v dnevniku veliko več kot le 25 različnih simbolov. Vendar sta nekako uspeli dekodirati besedo PAVEL, to je ime Lucijinega brata, ki je v dnevniku zapisana takole:

Dekodirajte ime Lucijinega fanta, ki je v dnevniku zapisano z naslednjimi simboli:

A) JOŽEF
B) PETER
C) JANEZ
D) DENIS

Rešitev

Lucijinemu fantu je ime JOŽEF.

Ker je v dnevniku več kot 25 različnih simbolov, je posamična črka lahko zakodirana na različne načine. To lahko preverimo s črko E, saj se četrti simbol v zakodirani besedi PAVEL ne pojavi v zakodiranem imenu Lucijinega fanta, čeprav črka E nastopa v vseh štirih možnih imenih.

V zakodiranih simbolih imajo poseben pomen vodoravne in navpične črte. Število vodoravnih črt ustreza številki vrstice, v kateri se črka nahaja v tabeli. Podobno število navpičnih črt ustreza številki stolpca, v katerem se črka nahaja v tabeli. Torej lahko črko, ki je zakodirana s tem simbolom, najdemo na presečišču ustrezne vrstice in ustreznega stolpca.

Tako ima na primer prvi znak  2 vodoravni in 5 navpičnih črti. V tabeli torej poiščemo črko, ki se nahaja v 5. stolpcu 2. vrstice, to je črka J. Na tak način lahko dekodiramo vse znake in dobimo ime JOŽEF.

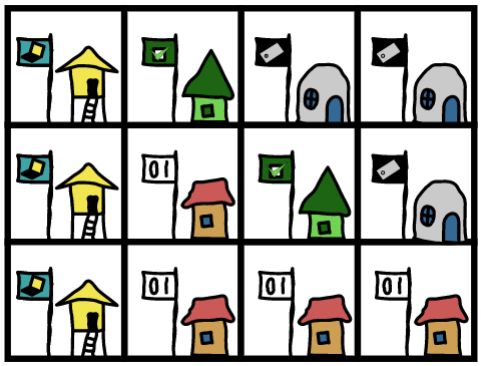
Računalniško ozadje

Kodiranje besedila je v računalništvu zelo pomembno. Uporabljamo ga pri komunikaciji preko interneta, saj je le tako lahko komunikacija varna in preneseni podatki ostanejo skriti.

Združene države Digitalije

državno, 8. in 9. razred, srednja šola

V daljni deželi Digitaliji so štiri države, prebivalci vsake pa živijo v več mestih. Na sliki je poenostavljen zemljevid dežele z 12 mesti, v katerih živijo Binarci, Mobitelci, Čekboksi in Laptopi.



Nekega dne se odločijo, da se bodo združili. Ker pa je to precej zapleten proces, so se ga lotili po korakih: v vsakem koraku se združita le dve državi. Zaradi priprave obsežne birokracije ob združevanju traja združevanje dveh držav toliko mesecev, kolikor imata ti dve državi skupaj vseh mest. Ko se dve državi združita, postaneta ena država in postopek združevanja se nadaljuje, dokler na koncu ne dobimo ene same države.

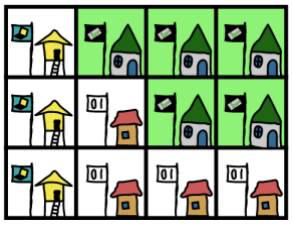
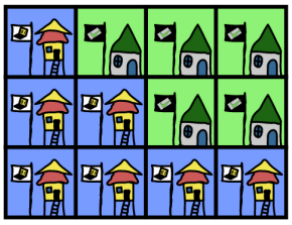
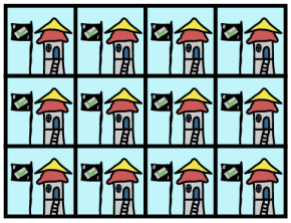
Najmanj koliko mesecev potrebujejo vse štiri države za združitev?

Rešitev

Za združitev potrebujejo najmanj 24 mesecev.

Najhitrejša strategija združevanja vseh štirih držav je, da minimiziramo število združevanj, v katera je vključeno vsako od mest. Tako vidimo, da moramo državo z največ mesti združiti zadnjo, saj največje število mest tako dodamo le enkrat. Torej moramo v vsakem koraku združevanja izbrati dve državi z najmanjšim številom mest.

Postopek je prikazan v spodnji tabeli:

1. korak: Ker imajo Čekboksi najmanjše število mest, jih najprej vključimo v združevanje. Drugo najmanjše število mest (to je 3) pa imata dve državi (Mobitelci in Laptopi), zato lahko izberemo katero koli izmed njiju. Recimo, da za prvi korak združevanja izberemo Mobitelce. Torej najprej združimo Čekbokse in Mobitelce in dobimo novo državo, recimo Mobičeke.  To združevanje traja **5 mesecev**, na koncu pa imamo Laptope (3 mesta), Binarce (4 mesta) in Mobičeke (5 mest).	**2. korak:** Sedaj sta dve državi z najmanjšim številom mest Laptopi (3 mesta) in Binarci (4 mesta). Torej se ti dve združita in dobimo novo državo, recimo Binlape.  To združevanje traja **7 mesecev**, na koncu pa imamo Mobičeke (5 mest) in Binlape (7 mest).	**3. korak:** Ostane nam še 5 mest Mobičekov in 7 mest Binlapov, ki jih združimo zadnje.  To združevanje traja **12 mesecev**.

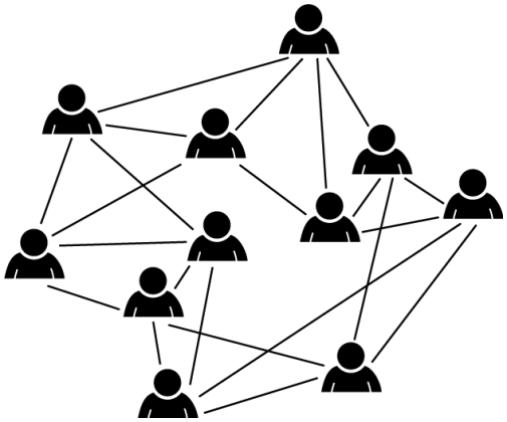
Torej je najmanjše število mesecev, ki jih potrebujemo za združitev vseh štirih držav, enako 5 + 7 + 12 = 24 mesecev.

Računalniško ozadje

Tudi v tej nalogi smo reševali problem optimizacije: poiskati moramo strategijo za rešitev, ki maksimizira ali minimizira določeno vrednost, upoštevajoč postavljene omejitve. S problemi optimizacije se dnevno srečujemo v našem življenju, ko iščemo najkrajšo pot do šole, najhitrejšo pot do blagajne, urnik aktivnosti s kar najmanj prekrivanja in podobno. Za reševanje optimizacijskih problemov lahko uporabimo različne algoritme, med njimi je tudi *požrešni algoritem* (angl. *greedy algorithm*).

Pri reševanju te naloge smo uporabili požrešni algoritem: na vsakem koraku združevanja smo se odločili za najboljšo možnost (najkrajši čas združevanja dveh držav) in to nas je privedlo tudi do najboljšega končnega rezultata (najkrajši čas za združitev vseh držav). Vendar v splošnem požrešni algoritmi ne privedejo vedno do optimalne končne rešitve (odvisno od samega optimizacijskega problema), a nam navadno v primernem času dajo spodoben približek optimalne rešitve.

Revizijski odbor državno, 8. in 9. razred, srednja šola



Člani Mestnega sveta so povezani z drugimi člani Mestnega sveta kot sodelavci v službi, družinski člani, poslovni partnerji ali pa so v isti politični stranki.

Mestni svet ima 11 članov. Če sta dva člana povezana, to označimo na diagramu s črto med njima.

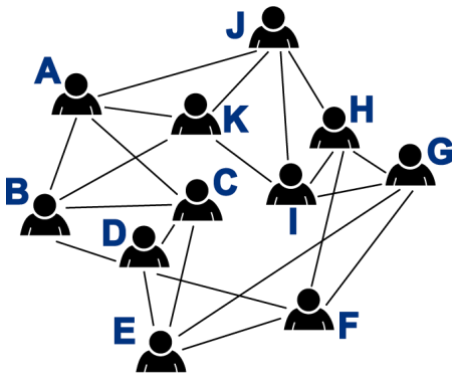
Izmed članov Mestnega sveta moramo izbrati Revizijski odbor, v katerem pa člani ne smejo biti medsebojno povezani.

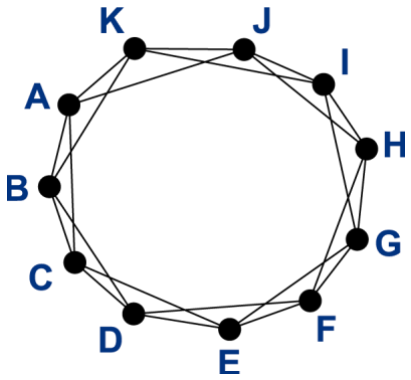
Največ koliko članov ima lahko Revizijski odbor?

Rešitev

Revizijski odbor ima lahko največ 3 člane.

Za lažjo razlago, kako pridemo do rešitve, člane Mestnega sveta na grafu označimo s črkami (slika spodaj levo). Hitro lahko preštejemo, da je vsak član Mestnega sveta povezan z natanko štirimi drugimi člani (iz vsake točke grafa imamo 4 povezave). Če graf preuredimo tako, kot ga prikazuje slika na desni, vidimo, da je vsaka točka povezana le s po dvema sosednjima točkama na levi in na desni.





Tako je na primer točka C povezana le s točkami A, B, D in E.

Torej lahko člane Revizijskega odbora poiščemo tako, da začnemo pri katerem koli članu Mestnega sveta (ker graf nima nobenih posebnih točk glede povezav), recimo pri članu A. Nato gremo, recimo, v nasprotni smeri urinega kazalca in izberemo naslednjega člana, ki ni povezan z A, to je član D. Nadaljujemo z naslednjim, ki ni povezan z D, to je G. In naslednji, ki ni povezan z G, je potem J. Vendar pa je J povezan z A, zato ne more biti član Revizijskega

odbora. Torej imamo v Revizijskem odboru lahko le tri medsebojno nepovezane člane, to so A, D in G.

Seveda lahko iskanje članov Revizijskega odbora začnemo pri katerem koli članu Mestnega sveta in iščemo v kateri koli smeri, a bomo na koncu vedno dobili največ tri medsebojno nepovezane člane.

Računalniško ozadje

Povezave med člani Mestnega sveta smo prikazali z grafom. Vozlišča grafa so bile osebe (člani), povezave pa njihove medsebojne relacije (sodelavci, družinske vezi …). V računalništvu grafe veliko uporabljamo, saj je mogoče z njimi opisati veliko različnih problemov.

Krajši zapis

srednja šola

Peter v svojih sporočilih zakodira črke z binarnimi števkami `1` in `0`. Ker se črki M in A pojavljata pogosteje, zanju uporabi krajši zapis. Določil je naslednje kode:

Črka	**M**	**A**	**E**	**L**	**N**	**J**
Koda	`1`	`00`	`0010`	`0110`	`1010`	`1110`

Peter je Evi poslal zakodirano sporočilo z nenavadnim imenom njegovega novega hišnega ljubljenčka:

`1 0 0 1 0 0 1 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 0 0 0`

Koliko črk ima ime Petrovega hišnega ljubljenčka?

Rešitev

Ime ima 8 črk.

Rešitev dobimo tako, da najprej dekodiramo Petrovo sporočilo. Torej moramo celotno sporočilo, zaporedje enic in ničel, razbiti na take dele, da se ujemajo s kodami posameznih črk. Če se razbijanja lotimo z leve strani, bomo kmalu naleteli na dvoumnost. Prvo črko enostavno določimo kot črko M (ustreza prvi `1`), a že pri drugi črki se zalomi: tu je lahko črka A, ki ji ustreza koda `00`, ali pa črka E, ki ima kodo `0010`. Na tem mestu ne moremo vedeti, katera je prava rešitev. Ob napačni izbiri, da je na drugem mestu črka A, bomo v naslednjih korakih prišli do mesta, od katerega ne bomo znali nadaljevati. Le tako bomo spoznali, da je bila naša izbira napačna (in postopek ponovili z drugo izbiro, črko E).

Problem je v tem, da Petrova koda ni prefiksna, kar pomeni, da se nekatere kode črk začnejo s kodo druge črke. Vendar če Petrovo kodo pogledamo bolj natančno, opazimo, da nobena črka nima kode, ki se konča s kodo druge črke. Torej je ta koda postfiksna. Če se torej lotimo dekodiranja sporočila od desne proti levi, ne bomo imeli težav z dvoumnostjo.

Tako lahko enostavno dekodiramo sporočilo, če z dekodiranjem začnemo na desni strani: zadnja črka je A, saj je to edina koda, ki se konča z `00`. Sledi črka J (`1110`), nato E (`0010`), N (`1010`), A (`00`), L (`0110`), E (`0010`) in M (`1`). Poslano sporočilo je torej MELANEJA, beseda pa ima 8 črk.

Računalniško ozadje

V računalniku je vse predstavljeno z biti, ki imajo lahko le dve stanji. Pri pretvorbi besedila v zaporedje bitov moramo paziti, da lahko to zaporedje bitov vedno pretvorimo nazaj v

originalno besedilo. To pa je možno le, če v nobenem primeru dva ali več različnih besedil nima enake binarne predstavitve. Torej moramo uporabiti kodiranje, ki ni dvoumno.

Če želimo pri kodiranju tudi stisniti besedilo (da dobimo čim krajšo binarno predstavitev besedila), se kot zelo dobra izkaže naslednja strategija: črkam, ki se bolj pogosto pojavljajo, priredimo krajšo kodo, za manj pogoste črke pa uporabimo daljšo kodo. Seveda moramo pri tem paziti, da izbrane kode zagotavljajo tudi učinkovito nedvoumno dekodiranje (torej rekonstrukcijo originalnega besedila). V ta namen se uporabljajo prefiksne in postfiksne kode, ki smo jih spoznali v tej nalogi.

Prigrizki

srednja šola

Podjetje Bober Snack pripravlja pakirane prigrizke, ki vsebujejo želode () in gobice (). Vsaka vrečka ima natanko 8 prigrizkov, zadnjega pa v vrečko vedno doda gospod Bober. Pri tem se drži naslednjih pravil:

- Če je v vrečki sodo število želodov, doda gobo.
- Če je v vrečki liho število želodov, doda želod.

Neke noči se je v skladišče podjetja prikradla nagajiva podgana in pustila naslednje sporočilo:

ZAMENJALA SEM EN ŽELOD IZ NEKE VREČKE Z ENO GOBO IZ NEKE DRUGE VREČKE.

Naslednje jutro je gospod Bober takoj opazil naslednje vrečke sumljivega izgleda:

- Vrečka 1: 
- Vrečka 2: 
- Vrečka 3: 
- Vrečka 4: 

Kateri sta tisti dve vrečki, v katerih je nagajiva podgana zamenjala prigrizke?

Rešitev

Prigrizki so bili zamenjani v vrečkah 2 in 3.

Pri iskanju rešitve se osredotočimo na učinek, ki ga ima zadnji dodan prigrizek gospoda Bobra. Opazimo, da je pravilo za dodajanje odvisno od števila želodov v vrečki. Če je med prvimi sedmimi prigrizki v vrečki sodo število želodov, dodajanje gobice ne spremeni števila želodov. Torej v vrečki ostane sodo število želodov. Če pa je med prvimi sedmimi prigrizki v vrečki liho število želodov, dodajanje še enega želoda poveča število želodov v vrečki za 1. Zato tudi število želodov v vrečki postane sodo.

Torej ima vsaka dopolnjena vrečka vedno sodo število želodov. Če je v neki vrečki liho število želodov, to pomeni, da je imela tačke vmes nagajiva podgana.

Vrečka 1 vsebuje 4 želode, vrečka 2 vsebuje 3 želode, vrečka 3 vsebuje 7 želodov in vrečka 4 vsebuje 2 želoda. Ker sta bili v zamenjavi uporabljeni le dve vrečki, je bila zamenjava narejena med vrečkama 2 in 3.

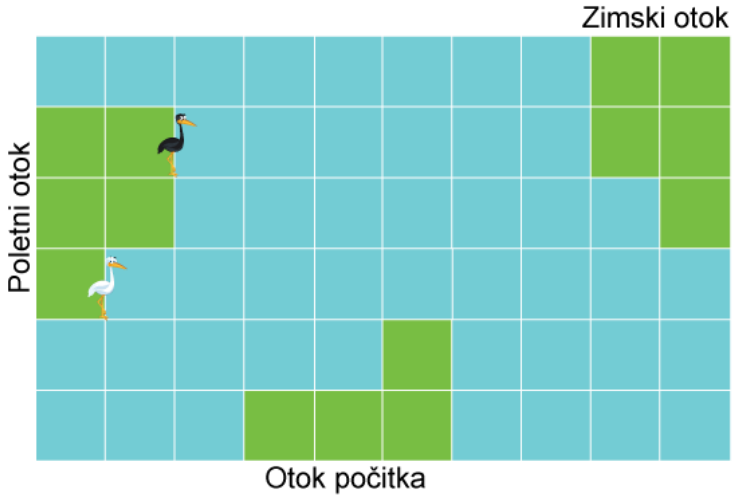
Računalniško ozadje

Dodajanje osmega prigrizka v vrečko glede na število želodov v vrečki je bilo ključno za identifikacijo vrečk, pri katerih je prišlo do zamenjave. V računalništvu to ustreza detekciji napak, ob prenašanju po komunikacijskih kanalih in preko omrežij. Seveda v računalništvu namesto želodov in gobic uporabljamo ničle in enke oziroma bite podatkov.

Selitev ptic

srednja šola

V morju so trije otoki. Na Poletnem otoku sta dve ptici, ki se preko zime preselita na toplejši Zimski otok.



Črna čaplja lahko leti preko morja s hitrostjo 2 kvadrata na uro in mora po preletenih 4 kvadratih 1 uro počivati na kopnem.

Bela čaplja lahko leti preko morja s hitrostjo 4 kvadratov na uro, a mora po preletenih 4 kvadratih na kopnem počivati 2 uri.

Obe ptici lahko hodita po kopnem s hitrostjo 1 kvadrata na uro. Ptici se lahko premikata le po mreži levo, desno, gor in dol, ne moreta se premikati diagonalno.

Katera čaplja se lahko hitreje preseli s Poletnega na Zimski otok in koliko je hitrejša od druge ptice, če obe izbereta najhitrejšo pot selitve?

Rešitev

Bela čaplja je 2 uri hitrejša od črne čaplje.

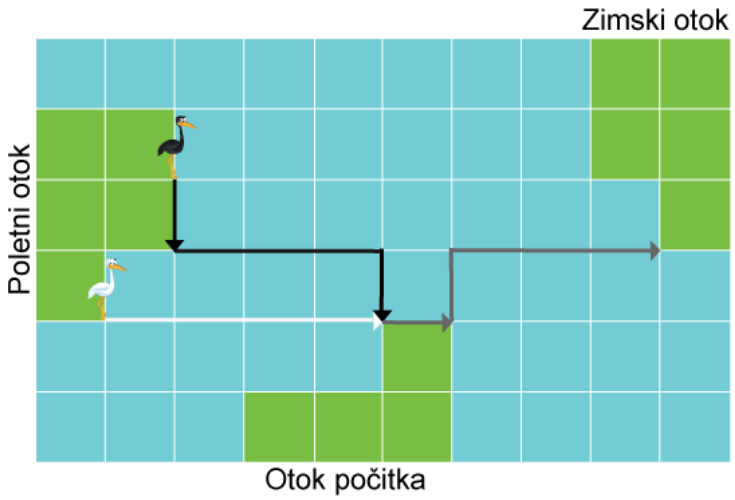
Najprej moramo ugotoviti najhitrejšo pot selitve za vsako ptico.

Bela čaplja lahko leti 4 kvadrate v desno s Poletnega otoka na Otok počitka, za kar porabi 1 uro. Po 2 urah počitka po kopnem prehodi en kvadrat na desno, za kar potrebuje 1 uro. Na koncu lahko preleti še 4 kvadrate od Otoka počitka do Zimskega otoka, za kar potrebuje 1 uro. Skupen čas selitve bele čaplje je tako 1 + 2 + 1 + 1 = 5 ur.

Črna čaplja najprej prehodi 1 kvadrat navzdol po otoku, za kar potrebuje 1 uro. Od tam lahko preleti 4 kvadrate do Otoka počitka, za kar potrebuje 2 uri. Po enournem počitku prehodi po otoku en kvadrat v desno, kar ji vzame še 1 uro. Nato pa lahko poleti še 4 kvadrate preko

morja do Zimskega otoka, za kar potrebuje 2 uri. Skupen čas selitve črne čaplje je tako 1 + 2 + 1 + 1 + 2 = 7 ur.

Torej je bela čaplja pri selitvi dve uri hitrejša od črne čaplje.



Obe ptici preletita preko morja najdaljšo razdaljo, ki jo zmoreta brez počitka. Zato ni možno, da bi kateri koli del (počasnejši) poti, ki jo prehodita, zamenjali s (hitrejšim) letenjem, saj njun doseg letenja ni dovolj velik, da bi dosegli naslednji otok. To pomeni, da sta zgoraj izračunana časa selitve optimalna.

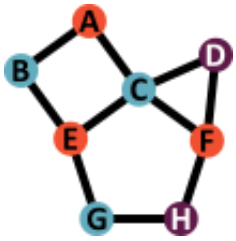
Računalniško ozadje

Naloga zahteva *optimizacijo*, to je iskanje najbolj učinkovite rešitve za podani problem. Poiskati smo morali najbolj direktno pot od enega do drugega otoka za vsako ptico. Pri rešitvi smo morali upoštevati tudi *omejitve*, saj je morala ptica počivati po preleteni določeni razdalji, kar omejuje razdaljo, ki jo ptica lahko preleti neprekinjeno. S kombiniranjem optimizacije in omejitev lahko kar najbolje uporabimo razpoložljive omejene *vire*, kot je ohranjanje energije ptice pri selitvi.

Skupinsko učenje

srednja šola

Osem učencev skupaj dela domačo nalogo. Ker posamezni učenci pri nalogi ne želijo sodelovati z določenimi drugimi učenci, se razdelijo v tri skupine.



To lahko predstavimo tudi z grafom, kjer vozlišče predstavlja učenca, barva vozlišča pa določa skupino. Povezava med dvema vozliščema pomeni, da ta dva učenca naloge ne želita delati skupaj.

Danes sta v učilnici le dve mizi, zato lahko učenci sestavijo le dve skupini. Zato bo potrebno dva izmed njih prepričati, da naj danes sodelujeta v isti skupini. Katera dva? Katero povezavo naj odstranimo iz zgornjega grafa, da dobimo le dve skupini namesto treh?

Rešitev

Odstraniti moramo povezavo C-F.

Prepričati dva učenca, da sodelujeta v isti skupini, pomeni odstraniti eno povezavo v grafu. Odstraniti moramo tako povezavo, da bomo lahko le z dvema barvama pobarvali vsa vozlišča, pri tem pa nobeni dve povezani vozlišči ne bosta iste barve.

Edina možna rešitev je, da odstranimo povezavo, ki je označena oranžno.

Tako dobljeni graf lahko pobarvamo z dvema barvama, kot prikazuje spodnja slika:

In kako vemo, da je odstranitev te povezave edina možna rešitev? Najprej razmislimo o trikotniku vozlišč desno zgoraj (označen oranžno):

Če odstranimo katerokoli povezavo izven tega trikotnika, bomo še vedno potrebovali tri različne barve, da pobarvamo vozlišča v tem trikotniku.

Razmislimo še o spodnjem peterokotniku (označen oranžno):

Če odstranimo katerokoli povezavo izven tega peterokotnika, se ta cikel ohrani in tudi zanj potrebujemo najmanj tri različne barve, da pobarvamo vsa vozlišča. Vozlišča lahko poskusimo pobarvati z dvema barvama: barvamo izmenično z dvema barvama v smeri urinega kazalca, a ko pridemo do zadnjega vozlišča, bo to imelo enako barvo kot prvo vozlišče, saj je število vozlišč v tem ciklu liho.

Torej moramo odstraniti tisto povezavo, ki hkrati prekine tako cikel trikotnika kot tudi cikel peterokotnika, in to je tudi edina možna rešitev.

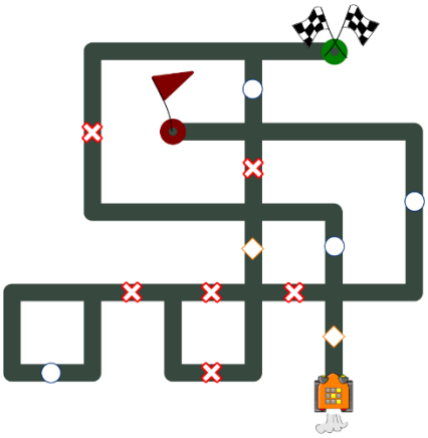
Računalniško ozadje

Veliko problemov iz realnega sveta lahko predstavimo kot problem barvanja grafa. Primer je graf v tej nalogi, kjer so vozlišča učenci in povezave pomenijo, da povezana učenca ne želita sodelovati v isti skupini. Če vozlišča pobarvamo s k barvami, to pomeni, da smo vsakega učenca dodelili v eno od k skupin. Katerikoli dve vozlišči, ki sta v grafu neposredno povezani, morata biti različnih barv.

Povezavi v grafu rečemo kritična povezava, če ob odstranitvi te povezave lahko graf pobarvamo z manj barvami. V nalogi smo poiskali kritično povezavo v grafu.

Usmerjanje robota

srednja šola



Robot začne na prikazani poziciji in se premika po poti. Na poti so trije znaki (, in), ki določajo smer robota v naslednjem križišču. Robot ne sme priti do rdeče zastavice ( ).

Znaki imajo naslednje pomene:

- **zavij levo** v naslednjem križišču,
- **zavij desno** v naslednjem križišču in
- **pojdi naravnost** v naslednjem križišču,

vendar ni znano, kakšen pomen ima kateri od treh znakov.

Pomen znaka je relativen glede na smer prihoda v križišče.

V primeru na levi sliki zelen trikotnik pomeni zavij levo v naslednjem križišču. Slika prikazuje gibanje robota, če pride v križišče iz različnih smeri: vedno zavije levo.

Kakšni morajo biti pomeni teh treh znakov, da bo robot lahko dosegel cilj v točki ?

Rešitev

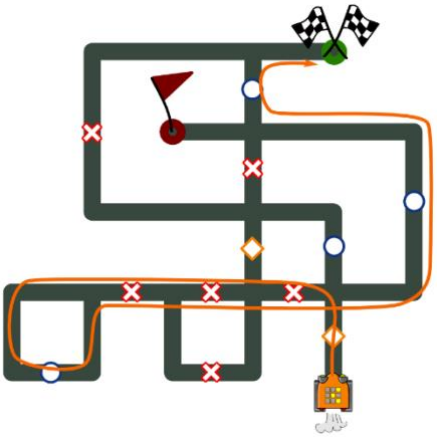
Odgovor je:

= pojdi naravnost

= zavij desno

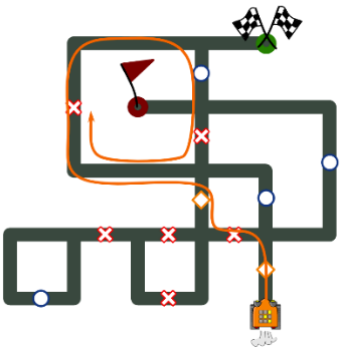
= zavij levo

Robotova pot je narisana na sliki.



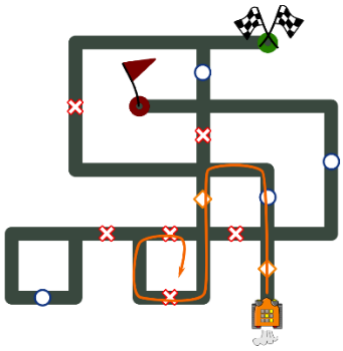
In kako pridemo do te rešitve?

Za rešitev problema lahko uporabimo različne strategije. Ena od njih je, da pregledamo vse različne možnosti. Ker je le 6 možnih načinov, kako lahko interpretiramo posamezne znake, lahko pregledamo vsako od možnosti (spodnje slike). Vidimo, da le ena vodi do cilja ( ).



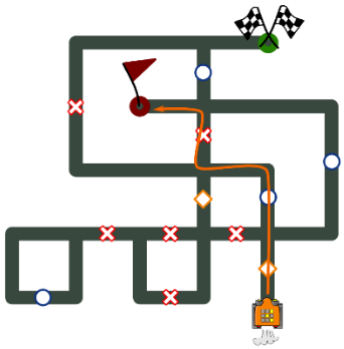
= zavij desno,
= pojdi naravnost,
= zavij levo

= zavij levo,
= pojdi naravnost,
= zavij desno



= zavij desno,
= zavij levo,
= pojdi naravnost

= pojdi naravnost,
= zavij desno,
= zavij levo



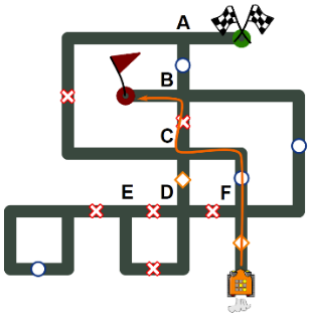
= zavij levo,
= zavij desno,
= pojdi naravnost

= pojdi naravnost,
= zavij levo,
= zavij desno

Druga strategija pa je sledenje poti, ki bi jo lahko naredil robot, in sprotno določanje pomena znakov. Če robot na svoji poti doseže rdečo zastavico, začne krožiti ali pa pomena znakov ne moremo konsistentno uporabiti, se pomaknemo nazaj za en korak (do zadnjega križišča) in poskusimo drugo možno pot.

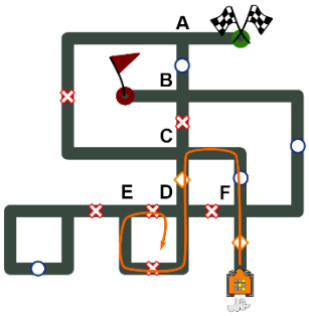
Ko pride robot do prvega križišča (F), ima tri možnosti: lahko zavije levo, gre naravnost ali pa zavije desno.

Možnost 1: Recimo, da gre robot naravnost. To pomeni, da znak ◇, ki ga je robot videl na poti do križišča, pomeni *pojdi naravnost*. Robot nadaljuje do križišča C. Ker *pojdi naravnost* določa znak ◇, robot pa je na poti videl znak ○, lahko zavije le levo ali desno.



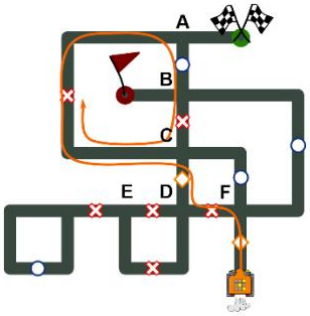
Možnost 1.1: Recimo, da robot na križišču C zavije desno. To pomeni, da znak ○ pomeni *zavij desno*. Potem znak ✕ pomeni *zavij levo*. Tako robot na križišču B zavije levo in pride do ⚑. Ker to očitno ni prava pot, se vrnemo nazaj na križišče C in poskusimo z drugo možnostjo.

Možnost 1.2: Recimo, da robot na križišču C zavije levo. To pomeni, da znak ○ pomeni *zavij levo*, znak ✕ pa *zavij desno*. Na križišču D gre robot naravnost, ker ◇ pomeni *pojdi naravnost*, in pride do križišča E. Tu zavije desno, daj je pred križiščem videl znak ✕. Na križišču D ponovno zavije desno in se tako ujame v zanko. Ker tudi to ni prava rešitev, se ponovno vrnemo na križišče C. Vendar pa smo na tem križišču preverili že vse možne poti (in nobena ni bila prava), zato lahko sklepamo, da naša prva predpostavka ni bila pravilna. Vrnemo se na križišče F in poskusimo druge možne poti.



Možnost 2: Predpostavimo, da robot na križišču F zavije levo, torej znak ◇ pomeni *zavij levo*.

Možnost 2.1: Ko pride do križišča D, ima ponovno tri možne poti. Ker je pred križiščem videl znak ✕, lahko zavije desno ali gre naravnost (zavijanje levo zapoveduje znak ◇). Recimo, da zavije desno. To pomeni, da znak ✕ pomeni *zavij desno*, znak ○ pa *pojdi naravnost*. Od križišča D robot nadaljuje do C, kjer zavije levo, ter do A, kjer zavije desno. Na križišču D gre naravnost in na križišču C ponovno zavije desno. Tako bi se ujel v zanko. Ker tudi to ni prava rešitev, moramo ponovno korak nazaj, do križišča D, in raziskati druge poti.



Možnost 2.2: Edina druga možna pot v križišču D je, da nadaljuje naravnost. Torej znak ✕ pomeni *pojdi naravnost* in znak ○ *zavij desno*. Sedaj lahko sledimo poti robota, ki ga uspešno pripelje do cilja. S tem smo odkrili tudi pravi pomen vseh treh znakov.

Računalniško ozadje

Prva strategija, ki smo jo opisali v rešitvi, se imenuje *groba sila* (angl. *brute force*). To je način reševanja problema, pri katerem sistematično preverjamo vse možnosti, dokler ne najdemo rešitve. V primerih, ko je različnih možnosti veliko, je preverjanje vseh možnosti zelo počasno in neučinkovito. Zato je bolje uporabiti kakšno drugo strategijo.

Druga strategija, ki smo jo opisali v rešitvi, pa se imenuje *strategija vračanja* (angl. *backtracking*). Pri tem sproti iščemo možnosti za rešitev in opustimo trenutno možnost rešitve takoj, ko se izkaže, da ne vodi v pravilno rešitev. Prednost te strategije pred strategijo grobe sile je v tem, da nam nove možne rešitve ni potrebno graditi od začetka, ampak se vrnemo le korak nazaj, v katerem smo naredili zadnjo (napačno) izbiro in nadaljujemo z drugo izbiro.

V naši nalogi imamo le malo spremenljivk (znakov), zato je le malo (samo 6) različnih poti, ki jih lahko ubere robot. Zato v tem primeru dobro deluje tudi strategija grobe sile. Pri večjih problemih, ko imamo več spremenljivk in moramo raziskati več možnih poti, pa strategija vračanja nudi boljšo in bolj elegantno rešitev.

Uganke, kot je na primer Sudoku, lahko elegantno rešimo z uporabo vračanja.

Izdelava pručk

državno, srednja šola

Bober Albert izdeluje pručke. Vsaka pručka ima štiri enako velike noge, ki določajo velikost pručke. Ker si stranke želijo večje ali manjše pručke, Albert izdela pručke različnih velikosti.

Ko Albert v gozdu nabira les za pručke, ne uspe vedno najti enako velikih kosov za noge. Vendar pa lahko vsako nogo poljubno skrajša, a ob skrajšanju noge preostanka materiala ne more več uporabiti.

Trenutno ima Albert na zalogi naslednjih 32 nog za pručke:

Dolžina:	10	9	8	7	6	5	4	3	2
Število:	3	6	3	3	5	3	3	2	4

Ker je krajšanje nog precej mukotrpno delo, želi Albert skrajšati kar najmanj nog. Najmanj koliko nog mora skrajšati, da lahko izdela 8 pručk?

Rešitev

Skrajšati mora najmanj 6 nog.

Ker za 8 pručk potrebuje 32 nog, bo moral uporabiti vse noge iz zaloge. Najbolj učinkovito jih lahko uporabi, če upošteva naslednji dve pravili:

Pravilo 1: Če je na zalogi število nog določene dolžine večje od 4, uporabi kar največ kompletov štirih nog te dolžine.

Ker za pručko potrebujemo 4 enako velike noge, ni potrebno, da noge krajšamo, če že imamo 4 noge enake dolžine. Tako bi lahko uporabili vse komplete 4 nog enake dolžine (9, 6 in 2), ostanejo pa nam še naslednje noge:

Dolžina:	10	9	8	7	6	5	4	3
Število:	3	2	3	3	1	3	3	2

Pravilo 2: Vse noge enake dolžine uporabi za pručko ali pa jih skrajšaj.

To pravilo je očitno za najdaljše in najkrajše noge. Najdaljše noge (dolžine 10) moramo skrajšati, saj jih ne moremo uporabiti za izdelavo pručke (manjka ena noga, ki pa je ne moremo narediti s krajšanjem). Najkrajše noge (dolžine 3) pa moramo uporabiti za pručko, saj jih ne moremo skrajšati in uporabiti za manjšo pručko (oz. tako krajšanje bi bilo nepotrebno).

V tabelo dopišimo še manjkajoče število nog vsake dolžine za izdelavo pručke:

Dolžina:	10	9	8	7	6	5	4	3
Število:	3	2	3	3	1	3	3	2
Manjkajoče:	0	2	1	1	3	1	1	2

Če bi za krajšanje uporabili noge le dveh različnih dolžin (dolžine 10 in dolžine 8), ne bi mogli dobiti dovolj nog za pručke, saj nam manjka več kot 6 nog ($3 + 3 < 2 + 1 + 3 + 1 + 1 + 2$). Torej moramo za krajšanje uporabiti noge treh različnih dolžin.

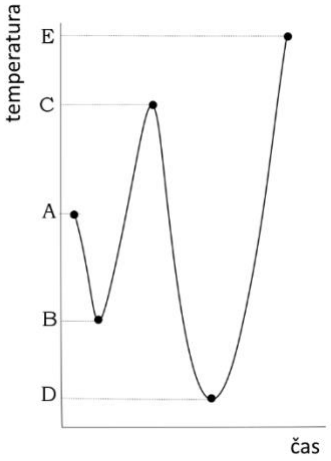
Najmanjše število nog treh različnih dolžin je 6 (dolžine 10, 9 in 6). Nogo dolžine 6 lahko skrajšamo na dolžino 5 in tako dopolnimo komplet štirih nog dolžine 5. Preostalih 5 najdaljših nog pa lahko skrajšamo tako, da s po eno nogo dopolnimo komplete nog dolžine 8, 7 in 4, dve nogi pa dodamo kompletu dolžine 3.

Računalniško ozadje

Za reševanje te naloge si je bilo potrebno izmisliti preprost optimizacijski postopek. V računalništvu optimizaciji posvečamo veliko pozornosti, saj s tem pospešimo delovanje računalnikov.

Prijetna temperatura

državno, srednja šola



Bober Teo večkrat dnevno meri temperaturo vode in meritve beleži v tabeli. Prvo meritev naredi takoj, ko vstane, zadnjo pa, preden gre spat. Ker se temperatura vode neprestano spreminja, tekom dneva zabeleži le ekstremne temperature; to so tiste, do katerih je temperatura naraščala in nato začne padati ali pa obratno, po padanju temperature je ta začela naraščati. Če se je temperatura spreminjala, kot prikazuje desna slika, je Teo zabeležil v tabelo vrednosti temperature v točkah A, B, C, D in E.

Teo je zelo občutljiv na temperaturo vode in le ena vrednost temperature je tista, ki je zanj prijetna in pri kateri se najbolje počuti.

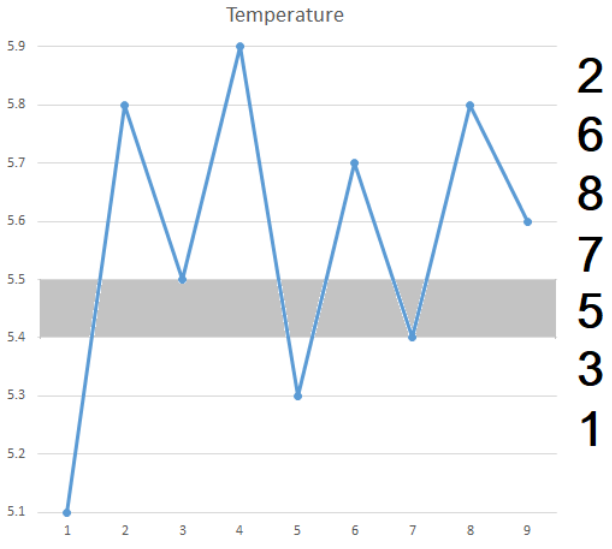
Včeraj je bila temperatura vode za Tea prijetna natanko petkrat. V tabelo je Teo zapisal naslednje temperature: 5,1, 5,8, 5,5, 5,9, 5,3, 5,7, 5,4, 5,8 in 5,6.

Med katerima vrednostma se nahaja za Tea prijetna temperatura vode?

Rešitev

Med 5,4 in 5,5.

Iz zabeleženih temperatur lahko sestavimo graf, podoben grafu iz primera v nalogi.



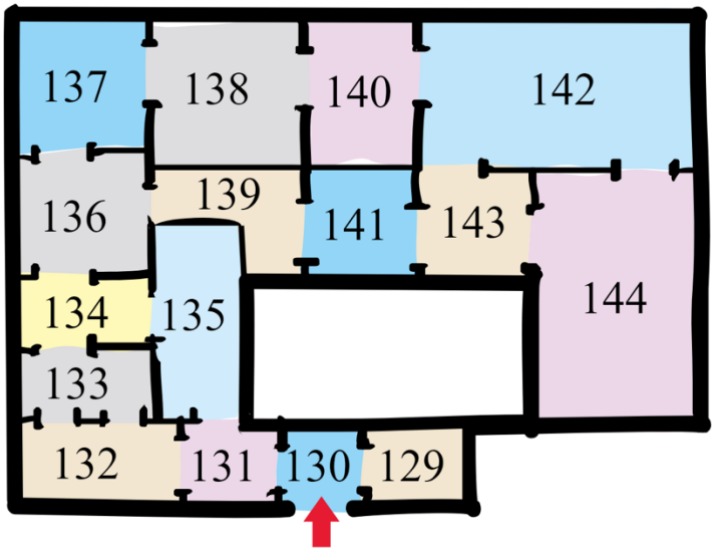
Nato za vsak interval preštejemo, koliko črt ga seka (številke so zapisane na desni). Edino vrednost temperature na intervalu med 5,4 in 5,5 (meje intervala niso vključene) je bila v tem dnevu dosežena natanko petkrat.

Računalniško ozadje

Pri reševanju te naloge ste uporabili osnovne prijeme podatkovne analitike. Nalogo veliko lažje rešimo, če podatke predstavimo vizualno, z grafom.

Ogled muzeja

državno, srednja šola



Slika kaže načrt dela muzeja Ermitaž v Sankt Peterburgu. Viktorija je v muzej vstopila pri vratih, označenih z rdečo puščico. Želi si ogledati vse sobe tega dela muzeja, pri ogledu pa se bo ravnala po naslednjih pravilih:

- Ko pride v sobo, najprej pregleda, če je zraven kakšna soba, ki je še ni obiskala.
  - Če taka soba obstaja, bo obiskala neobiskano sosednjo sobo z najmanjšo številko.
  - Če take sobe ni, si ogleda eksponate v trenutni sobi.
- Ko zaključi z ogledom eksponatov v sobi, se vrne v sobo, **iz katere je prvič vstopila v trenutno sobo**.

Katera bo peta soba po vrsti, v kateri si bo Viktorija ogledala eksponate?

Opomba: v knjižici smo – po pritožbi enega od tekmovalcev – spremenili navodilo iz "se vrne v zadnjo sobo, iz katere je prišla v trenutno sobo" v nedvoumnejše " se vrne v sobo, **iz katere je prvič vstopila v trenutno sobo**". Več o tem v razlagi rešitve računalniškega ozadja.

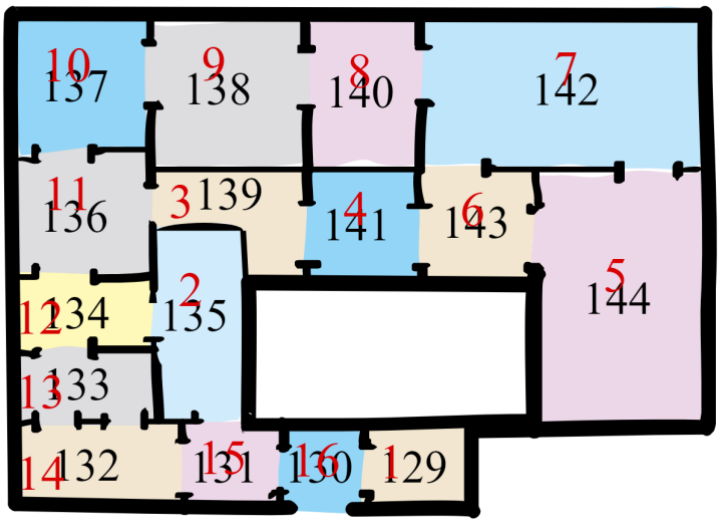
Rešitev

Soba 144.

Na načrtu sob lahko označimo zaporedje sob, v katerih si bo Viktorija ogledala eksponate ob upoštevanju postavljenih pravil.

Najprej Viktorija vstopi v sobo 130 in poleg te sobe sta dve neobiskani sobi, to sta 129 in 131. Naslednjo torej obišče sobo 129, ker je to neobiskana soba z najmanjšo številko. Ker ta soba nima nobene neobiskane sosednje sobe, si Viktorija ogleda eksponate v sobi (na načrtu označeno z rdečo številko 1) in se vrne v sobo 130. Nato obišče sobo 131 (sosednja neobiskana soba z najmanjšo številko) in gre naprej v 132, 133, 134 ter prispe v sobo 135. Ta soba nima nobene sosednje sobe, ki je Viktorija še ne bi obiskala (bila je že v sobah 131 in 134), zato si ogleda eksponate (torej je soba 135 druga, ki si jo ogleda) in se vrne v sobo 134. Od tu

nadaljuje v sobe 136, 137, 138, 140, 142, 143, 141 ter prispe v sobo 139, v kateri spet nima nobene še neobiskane sosednje sobe, zato si v njej ogleda eksponate (tretja soba, v kateri si ogleda eksponate) in se vrne v sobo 141. Tudi ta soba nima nobene še neobiskane sosednje sobe, zato si ogleda eksponate v njej (četrta soba za ogled eksponatov) in se vrne v sobo 143. Nadaljuje v naslednjo še neobiskano sosednjo sobo, 144. To je tudi njena zadnja še neobiskana soba. Ker soba 144 nima še neobiskanih sosednjih sob, si ogleda eksponate v njej, torej je to peta soba, v kateri si je ogledala eksponate.



Če sledimo Viktorijinim ogledom do konca, vidimo, da se je po ogledu sobe 144 vrnila v sobo 143, si jo ogledala in se vrnila v sobo 142, ki si jo je tudi ogledala. Tako si je ogledovala sobe in se vračala nazaj preko sob 140, 138, 137, 136, 134, 133, 132, 131 do sobe 130, ki si jo je ogledala zadnjo.

Računalniško ozadje

Viktorija je za ogled muzeja izbrala *algoritem iskanja v globino*. Ta algoritem je poleg *algoritma za iskanje v širino* najpogostejši način sistematičnega pregledovanja dreves in grafov, pri katerem obiščemo čisto vsak element (nobenega elementa ne izpustimo).

V opisu naloge smo povedali za kasnejši popravek. Ta zgodba kaže, zakaj naravni jeziki niso primerni za pisanje programov. Sestavljalec naloge je imel, predpostavljamo, v mislih razliko med "priti" (v smislu priti prvič) in "vrniti se" v sobo. V tem smislu je bilo prvotno navodilo pravilno. Po drugi strani pa tudi ob vračanju prideš, torej je bilo prvotno navodilo vseeno dvoumno. V tem primeru Viktorija iz 141 – po tem, ko si ogleda eksponate v njej – ne bi šla v 143, od koder je prvič stopila v 141, temveč v 139 od ondod "nazadnje prišla". Kot pravilna smo zato priznali oba možna odgovora.

Zato računalnike programiramo v računalniških jezikih, ne slovenščini. Tam takšnih dvoumnosti ni.

Resničnostna tabela

državno, srednja šola

Resničnostna tabela opisuje izhod (○ ali ●) za eno ali več vhodnih spremenljivk. Spodnja tabela tako opisuje izhod za vse možne kombinacije spremenljivk x, y in z.

x	y	z	izhod
○	○	○	○
○	○	●	●
○	●	○	○
○	●	●	○
●	○	○	●
●	○	●	●
●	●	○	●
●	●	●	●

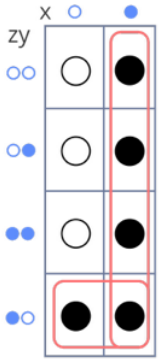
To tabelo bi lahko opisali tudi s formulo tako, da bi navedli vse kombinacije x, y in z, pri katerih je izhod enak ●:

```
(x=○ in y=○ in z=●) ali
(x=● in y=○ in z=○) ali
(x=● in y=○ in z=●) ali
(x=● in y=● in z=○) ali
(x=● in y=● in z=●)
```

V tej formuli smo zapisali 15 vhodnih simbolov.

Če bolj natančno pogledamo tabelo, vidimo, da je pri x=● izhod vedno ●. Zato lahko formulo skrajšamo in uporabimo le 4 simbole:

```
(x=○ in y=○ in z=●) ali
(x=●)
```



Vendar lahko formulo zapišemo še krajše! Tabelo lahko predstavimo z le 3 simboli:

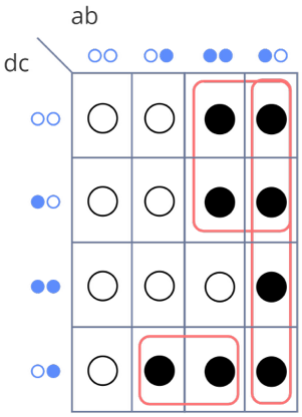
```
(y=○ in z=●) ali
(x=●)
```

V desnem diagramu sta obe skupini v formuli označeni rdeče.

Poglejmo sedaj nekoliko večjo resničnostno tabelo s štirimi vhodnimi spremenljivkami. Pripadajoči diagram je prikazan desno ob tabeli (bodite pozorni na vrstni red spremenljivk c in d).

Katero je najmanjše število simbolov, ki jih potrebujemo za zapis te resničnostne tabele?

a	b	c	d	izhod
○	○	○	○	○
○	○	○	●	○
○	○	●	○	○
○	○	●	●	○
○	●	○	○	○
○	●	○	●	○
○	●	●	○	●
○	●	●	●	○
●	○	○	○	●
●	○	○	●	●
●	○	●	○	●
●	○	●	●	●
●	●	○	○	●
●	●	○	●	●
●	●	●	○	●
●	●	●	●	○



Rešitev

Odgovor je: 7 simbolov.

Rešitev najdemo v diagramu, ki pripada resničnostni tabeli. V njem poiščemo vse skupine, ki označujejo izhod ● (v diagramu so že označene z rdečo). Velikost označene skupine pravzaprav določa, koliko simbolov potrebujemo za njen opis.

Za opis enega izhodnega simbola (skupina z enim samim ●) potrebujemo štiri vhodne simbole (a, b, c in d). Za opis dveh sosednjih izhodnih simbolov (skupina dveh ●) potrebujemo tri vhodne simbole. Za opis štirih sosednjih izhodnih simbolov (skupina štirih ●, 1x4 ali 2x2) potrebujemo dva vhodna simbola. Za opis osmih izhodnih simbolov (skupina osmih ●, 2x4) pa potrebujemo en sam vhodni simbol.

Na diagramu so prikazane tri skupine, dve zajemata štiri izhodne simbole, ena pa dva izhodna simbola. Torej za njihov opis potrebujemo 2 + 2 + 3 = 7 vhodnih simbolov.

Vsako od treh skupin (skupina 2x2, skupina 1x4 in skupina 2x1) opišemo s formulo ter jih združimo z *ali*. Rešitev za podano resničnostno tabelo je tako formula s skupno 7 vhodnimi simboli:

```
(a=● in c=○) ali
(a=● in b=○) ali
(b=● in c=● in d=○)
```

Računalniško ozadje

V nalogi smo si pri reševanju pomagali s Karnaughjevimi diagrami. Z njimi lahko poenostavimo logične tabele in jih izrazimo z uporabo najmanjšega števila logičnih izrazov. To je zelo pomembno pri izdelavi logičnih vezij, saj je cena vezja odvisna tudi od števila različnih logičnih elementov, ki jih uporabimo pri izdelavi. Zato poskušamo izdelati tako vezje, da uporabimo kar najmanj logičnih elementov. Pri iskanju najmanjšega potrebnega nabora elementov si pomagamo s Karnaughjevim diagramom.

Karnaughjevi diagrami so zasnovani tako, da pri premiku iz enega na drugo sosednje polje naredimo spremembo le pri enem bitu. To velja tudi za prehod z zadnjega na prvo polje (npr. s spremembo enega bita pridemo s prvega polja v stolpcu na zadnje polje v istem stolpcu). Ravno te lastnosti Karnaughjevih diagramov omogočajo, da v primeru, ko dve sosednji polji vsebujeta izhod 1 (v našem primeru ●), lahko v formuli izpustimo bit, ki ga spremenimo pri prehodu med tema dvema poljema, saj ga ne potrebujemo v končnem izrazu. Na ta način združimo dve vrstici resničnostne tabele.

Karnaughjevi diagrami so uporabni, če imamo do štiri vhodne spremenljivke. Čeprav se to ne sliši veliko, imajo v praksi manjša vezja štiri ali manj vhodov ali pa lahko večje vezje razdelimo na več manjših s štirimi ali manj vhodi.

Pregled nalog

Šolsko tekmovanje

			2.	3.	4.	5.	6.	7.	8.	9.	SŠ
Gujeolpan 1	Južna Koreja		●								
Zrcalna slika	Nemčija		●								
Igrače	Pakistan		●	●							
Od pike do pike	Islandija		●	●							
Želje 1	Nemčija		●	●							
Gujeolpan 2	Južna Koreja			●							
Ribe	Kanada			●							
Darila	Irska			●	●						
Smešni filtri	Tajvan			●	●						
Kolački	Irska			●	●	●	●	●			
Kocke	Romunija				●	●					
Pobiranje korenja	Kitajska				●	●					
Poplavljanje	Portugalska				●	●					
Ugani kdo	Južna Koreja				●	●					
Chez Connie	Švica				●	●					
Vzorec pajkove mreže 1	Kanada				●	●					
Marko skače	Slovenija				●	●	●	●	●	●	
Iskanje zaklada	Nemčija					●					
Kengurujki	Uzbekistan					●					
Pikčasete kukavice	Kanada					●					
Želje 2	Turčija					●					
Nakupovanje	Tajvan						●	●			
Restavracija Fifo	Saudova Arabija						●	●			
Vreča kovancev	Irska						●	●			
Barvno presenečenje	Indonezija						●	●	●	●	

			2.	3.	4.	5.	6.	7.	8.	9.	SŠ
Blagajne	Pakistan						●	●	●	●	
Se srečata?	Litva						●	●	●	●	
Mačje slike	Švica						●	●	●	●	●
Marsovska koda	Turčija						●	●	●	●	●
Namakanje polj	Turčija						●	●	●	●	●
Sneguljčica in prepirljivi palčki	Italija						●	●	●	●	●
Sončnica	Saudova Arabija						●	●	●	●	●
Stara roba, nova roba	Velika Britanija						●	●	●	●	●
Vzorec pajkove mreže 2	Kanada						●	●	●	●	●
Herkul in Hidra	Uzbekistan								●	●	●
Popoldanski dremež	Tajvan								●	●	●
Primerjave	Litva								●	●	●
Rokoborba	Avstralija								●	●	●
Tatovi umetnin	Slovenija								●	●	●
Krajši zapis	Švica										●
Prigrizki	Filipini										●
Selitev ptic	Avstralija										●
Skupinsko učenje	Švica										●
Usmerjanje robota	Češka										●

Državno tekmovanje

			6.	7.	8.	9.	SŠ
Jagodni tat	Švica		●	●			
Trije bobri	Litva		●	●	●	●	
Barvanje ograje	Litva		●	●	●	●	
Pikčaste kukavice	Kanada		●	●	●	●	●
Genome Decoding	Rusija		●	●	●	●	●
Nujno srečanje	Litva		●	●	●	●	●
Hočem pistacijo!	Nova Zelandija		●	●	●	●	●
Kosilnica	Nemčija		●	●	●	●	●
Premikanje krogel	Južna Koreja		●	●	●	●	●
Keglji	Italija		●	●	●	●	●
Težave z mizo	El Salvador		●	●	●	●	●
Pretvarjanje	Turčija		●	●	●	●	●
Presedanje učencev	Španija		●	●	●	●	●
Skrivni dnevnik	Češka				●	●	●
Združene države	Filipini				●	●	●
Revizijski odbor	Češka				●	●	●
Izdelava pručk	Litva						●
Prijetna temperatura	Litva						●
Ogled muzeja	Rusija						●
Resničnostna tabela	ZDA						●

Avtorji nalog

Ahto Truu, Estonija

Aidan Mooney, Irska

Alenka Kavčič, Slovenija

Alieke Stijf, Nizozemska

Alina Gabriela Boca, Romunija

Alisher Ikramov, Uzbekistan

Allira Crowe, Avstralija

Anja Koron, Slovenija

Anton Chukhnov, Rusija

Arnheidur Gudmundsdottir, Islandija

Bashar Hussain, Sirija

Bence Gaál, Madžarska

Bundit Thanasopon, Tajska

Chris Roffey, Velika Britanija

Christian Datzko, Švica

Corina-Elena Vint, Romunija

Dong Yoon Kim, Južna Koreja

E Li, Kitajska

Eljakim Schrijvers, ZDA

Emil Kelevedjiev, Bolgarija

En-hsuan Lee, Tajvan

Eslam Wageed, Egipt

Esraa Almajhad, Saudova Arabija

Eugenio Bravo, Španija

Ezgi Arzu Güneş, Turčija

Ezra Templonuevo, Filipini

Fabian Frei, Švica

Fatih Kürşat Cansu, Turčija

Filiz Kalelioğlu, Turčija

Florentina Voboril, Avstrija

Gabriela Lourdes Rodríguez Parada, Salvador

Gerald Futschek, Avstrija

Graeme Buckie, Avstralija

Greg Lee, Tajvan

Hannah Piper, Avstralija

Hongjin Yeh, Južna Koreja

Hsueh-Wang Chang, Tajvan

Inggriani Liem, Indonezija

Iorgulescu Tiberiu, Pakistan

J.P. Pretti, Kanada

Janez Demšar, Slovenija

Javier Bilbao, Španija

Jean-Philippe Pellet, Švica

Jihye Kim , Južna Koreja

JingWen Hai, Kitajska

Jiří Vaníček, Češka

João Rico, Portugalska

Jonatan Pipping, Švedska

Joseph Grace, Nova Zelandija

Jovana Popović, Srbija

Judith Lin, Tajvan

Juraj Hromkovic, Švica

Karsten Schulz, Nemčija & Avstralija

Kirsten Schlüter, Nemčija

Kris Coolsaet, Belgija

Kwangsik Moon, Južna Koreja

Kyra Willekes, Nizozemska

Laura Ungureanu, Romunija
Linda Björk Bergsveinsdóttir, Islandija
Lorenzo Repetto, Italija
Lucia Budinská, Slovaška
Lynn Liu, Tajvan
Madhavan Mukund, Indija
Margot Phillipps, Nova Zelandija
Maria Ilie-Niculescu, Romunija
Marielle Léonard, Francija
Marija Marolt, Slovenija
Mark Edward M. Gonzales, Filipini
Marta Burzanska, Poljska
Mārtiņš Opmanis, Latvija
Marvin G. Hall, Jamajka
Matias Hiron, Francija
Mauro Torelli, Italija
Mayad Farhoud, Sirija
Michael Weigend, Nemčija
Mija Zaļūksne, Latvija
Milan Lukić, Srbija
Milan Rajković, Srbija
Mile Jovanov, Severna Makedonija
Mochammad Irfan Noviana, Indonezija
Nalo Reiter, Švica & Romunija
Nežka Rugelj, Slovenija
Pavel Kohutovič, Češka
Pedro Marcelino, Portugalska
Pedro Ribeiro, Portugalska
Pieter Waker, Južna Afrika
Qing Zou, Kitajska
Raluca Constantinescu, Romunija
Regula Lacher, Švica
Rodrigo Santamaría, Salvador
Ruwan Devasurendra, Avstralija
Sarah Atkins, Avstralija
Sarah Chan, Kanada
Sébastien Combéfis, Belgija
Sergei Pozdniakov, Rusija
Shang Fei, Kitajska
Shu-Jyuan Yang, Tajvan
Soojin Jun, Južna Koreja
Susannah Quidilla, Avstralija
Suzanne Datzko, Švica
Špela Cerar, Slovenija
Taina Lehtimäki, Irska
Timur Sitdikov, Uzbekistan
Tom Naughton, Irska
Tomas Šiaulys, Litva
Ungyeol Jung, Južna Koreja
Vaida Masiulionytė-Dagienė, Litva
Vaidotas Kinčius, Litva
Valentina Dagienė, Litva
Violetta Lonati, Italija
Viorica Mariela Ungureanu, Pakistan
Vipul Shah, Indija
Vu Luan, Vietnam
Wolfgang Pohl, Nemčija
Ya-Chun Hsu, Tajvan
Yasemin Gülbahar, Turčija
YongJu Jeon, Južna Koreja
Zsuzsa Pluhár, Madžarska