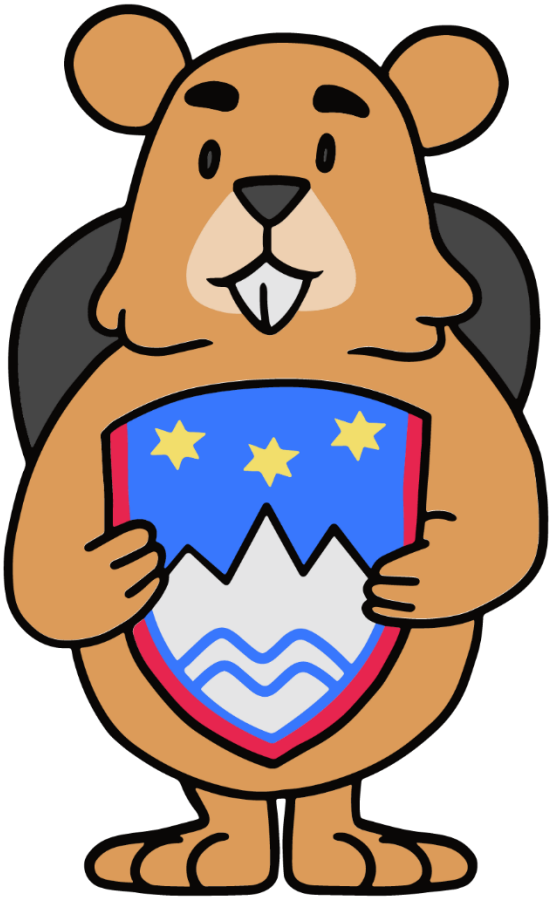
Bober 2025/26

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

Alenka Kavčič (UL FRI)
Nežka Rugelj (OŠ Trzin)
Rok Požar (UP FAMNIT)
Špela Cerar (UL PEF)



Bebras

ACM Slovenija

Naloge in rešitve

Uredniki zbirke nalog in rešitev:

Alenka Kavčič (UL FRI)
Nežka Rugelj (OŠ Trzin)
Rok Požar (UP FAMNIT)
Špela Cerar (UL PEF)

Razvoj tekmovalnega sistema:

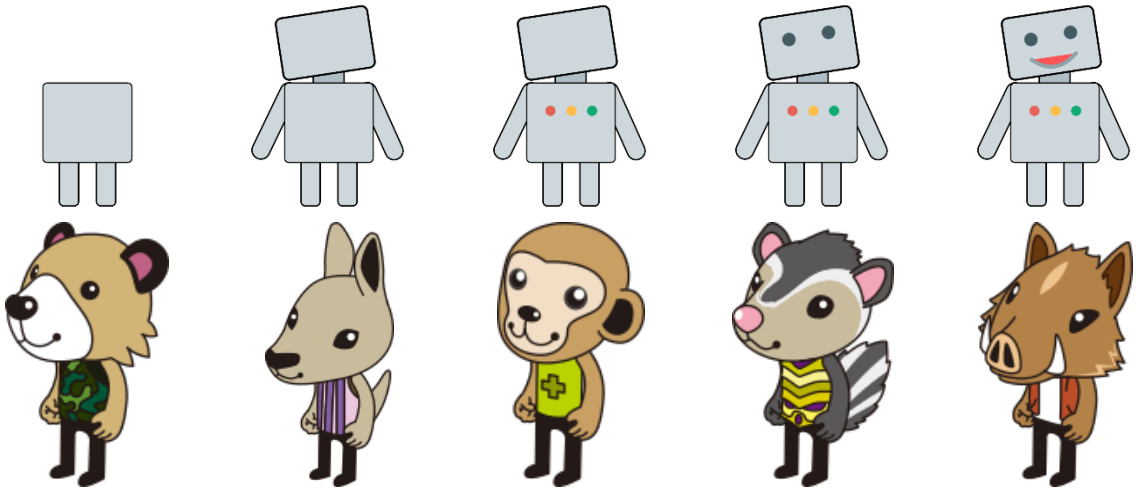
Gašper Fele Žorž (UL FRI)
Gregor Jerše (UL FRI)

Kazalo nalog

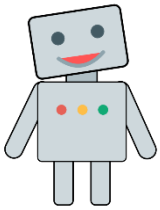
Sestavljanje robotov 1

2. razred

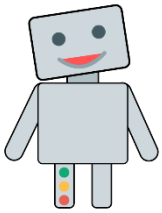
Pet živali dela v tovarni robotov. Vsaka od njih ima drugačno nalogo. Naloge opravljajo v naslednjem vrstnem redu:



Pravilno izdelan robot izgleda takole:



Ampak nekdo se je zmotil, zato novi robot izgleda takole:



Kdo se je zmotil?

Rešitev

Pravilen odgovor je  , saj so lučke, ki jih doda, na robotovi nogi namesto na njegovem trupu. Vsi ostali deli robota so na pravih mestih, saj so ostale živali svoje delo opravile brez napak.

Računalniško ozadje

Pri programiranju pogosto prihaja do napak. Iskanju napak v programski kodi pravimo razhroščevanje.

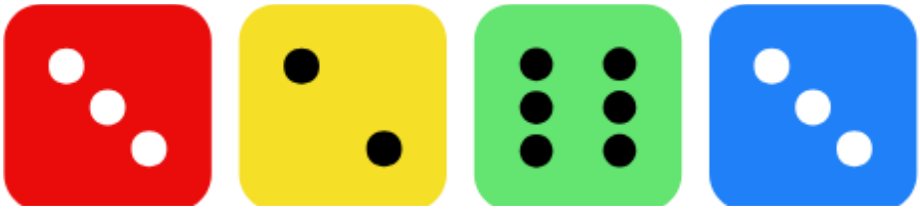
Nariši pošast 1

2. in 3. razred

Igrajmo se igro nariši pošast. Imamo 4 kocke: rdečo, rumeno, zeleno in modro. Barva kocke skupaj s številom pik na njej določa lastnosti pošasti, in sicer:

	**Rdeča kocka**	Število oči
	**Rumena kocka**	Število rogov
	**Zelena kocka**	Število rok
	**Modra kocka**	Število zob

Vrgli smo kocke in padle so takole:



RDEČA KOCKA **RUMENA KOCKA** **ZELENA KOCKA** **MODRA KOCKA**

Nariši pošast in pri tem upoštevaj vrednosti na kockah.

Rešitev

Kakršnakoli pošast s 3 očmi, 2 rogoma, 6 rokami in 3 zobmi.

Računalniško ozadje

Tako kot računalnik sledi le podanim navodilom, ste to storili tudi vi. Ste morda opazili, da so bile pošasti, ki ste jih narisali zelo raznolike? To se zgodi, če navodila niso dovolj natančno podana.

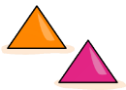
Navodila za gradnjo 1

2. in 3. razred

Na voljo imaš:

6 kock

1 most

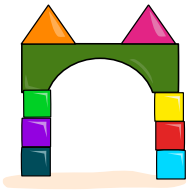


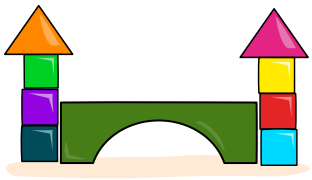
2 piramidi

Prijatelj ti da naslednja navodila:

1. Vzemi 3 kocke in iz njih zgradi stolp.
2. S preostalimi 3 kockami zgradi še en stolp.
3. Postavi most.
4. Vzemi 2 piramidi in ju postavi na strukturo.

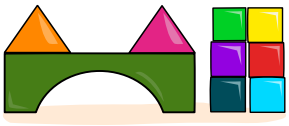
Po teh navodilih lahko zgradiš različne strukture, kot sta na primer ti dve:



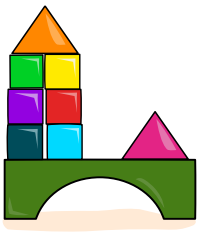


Katero od naslednjih struktur lahko zgradiš, če slediš zgornjim navodilom?

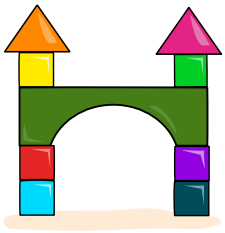
A)



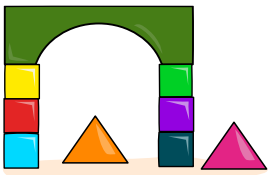
B)



C)



D)



Rešitev

Odgovor A je pravilen.

Odgovor B ni pravilen, saj je most pod kockami, čeprav bi morali najprej postavili stolpa in naj ju ne bi premikali.

Odgovor C je napačen, ker kock nismo zložili v 2 stolpa s po tremi kockami.

Odgovor D ni pravilen, ker piramid nismo postavili na strukturo.

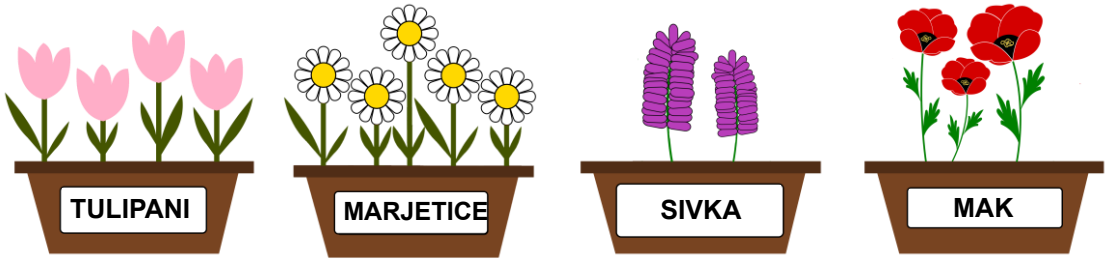
Računalniško ozadje

Računalnik zna slediti le natančnim navodilom in naredi le tisto, kar mu naročimo.

Rože

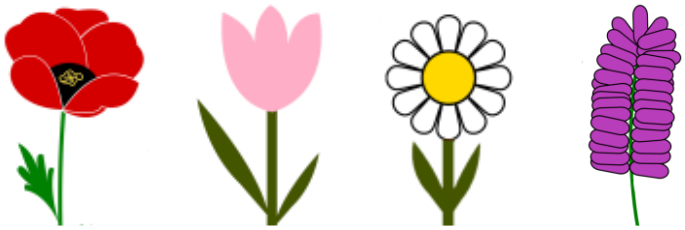
2. in 3. razred

Jana pripravlja šopek. Rože jemlje iz 4 posod, v katerih so naslednje rože:



Jana iz posod vzame 1 mak  , 1 tulipan  in 3 marjetice  .

V kateri posodi je ostalo največ rož? Obkroži vrsto rož, ki so v tej posodi.



Rešitev

Pravilen odgovor je tulipan.

V posodi s tulipani so ostali 4 – 1 = 3 tulipani.

V posodi z marjeticami sta ostali 5 – 3 = 2 marjetici.

V posodo s sivko sta ostali 2 – 0 = 2 vejici sivke.

V posodi z makom sta ostala 3 – 1 = 2 maka.

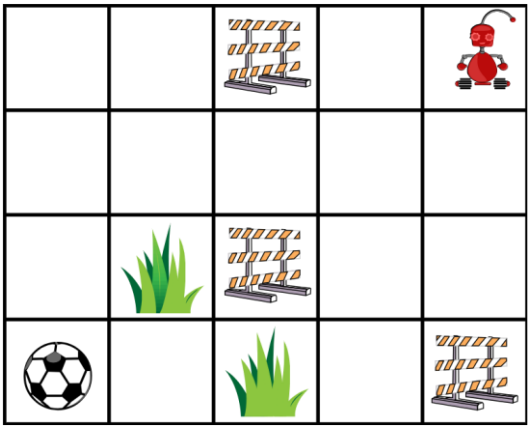
Računalniško ozadje

Vsaka posoda predstavlja prostor za shranjevanje podatkov. V računalništvu pogosto shranjujemo podatke in po potrebi spreminjamo njihove vrednosti, tako kot smo to počeli, ko smo iz posod jemali različno število rož.

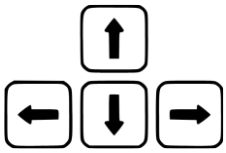
Rokov robot

2. do 4. razred

Rok vodi robota po mreži od mesta, kjer se nahaja, do žoge in pri tem pazi, da se robot ne zaleti v nobeno od ovir (to so ograje in trava).



Robota usmerja s tipkami s puščicami. Robot se premika v smeri puščic.

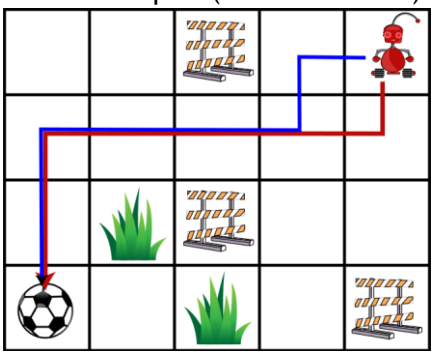


Rok lahko v eni potezi premakne robota za eno polje.

V kakšnem zaporedju mora pritisniti na tipke, da bo robot prišel do žoge po najkrajši možni poti? Nariši puščice v pravilnem vrstnem redu.

Rešitev

Možni sta 2 poti (modra in rdeča):



S puščicami to predstavimo kot

Modra pot:

Rdeča pot:

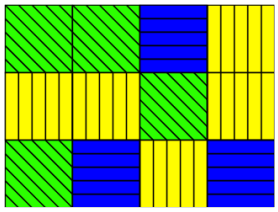
Računalniško ozadje

Z zapisom zaporedja tipk, ki jih mora pritisniti Rok, si napisal preprost program.

Barvanje kvadratov

3. razred

Eva in Tom barvata kvadrate. Eva jih barva z rumeno  , Tom pa z modro  . Če oba pobarvata isti kvadrat, se ta kvadrat obarva zeleno  .



Koliko kvadratov je pobarvala Eva?

Rešitev

Eva je pobarvala vse rumene kvadrate - teh je 5 – in vse zelene kvadrate – to so 4. Skupaj je pobarvala 5 + 4 = 9 kvadratov.

Računalniško ozadje

Tudi pri risanju slik v Slikarju v resnici barvaš mrežo kvadratov, ki jim pravimo slikovne točke oz. piksli. Slika je v tem primeru v računalniku shranjena kot mreža slikovnih točk, za katere hranimo podatke o barvi.

Aleksov zaklad

3. in 4. razred

Raziskovalec Aleks je našel star zemljevid z zakladom!

Zemljevid prikazuje mrežo z različnimi simboli: tabor, trava, gora, voda in zaklad.

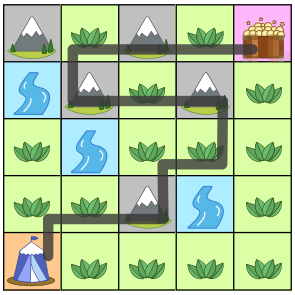
Aleks začne svojo pot do zaklada v taboru. Po poti upošteva naslednja pravila:

1. Ne sme stopiti v vodo.
2. Prečkati mora vsaj 1 goro.
3. Premika se lahko le gor ↑, dol ↓, levo ← ali desno →.

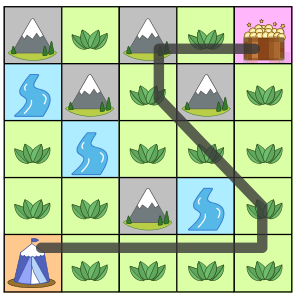
Spodaj so prikazane 4 poti. Po kateri naj gre Aleks, da bo upošteval vsa zgornja pravila?

A)

B)



C)

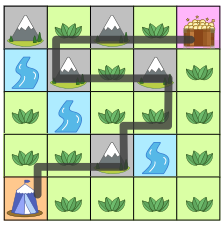


D)

Rešitev

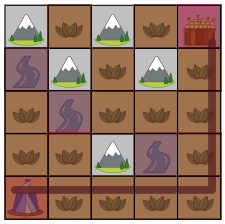
Odgovor A ni pravilen, saj pot poteka preko vode.

Odgovor B upošteva vse omejitve in je pravilen.



Odgovor C ni pravilen, saj pot poteka po diagonali.

Odgovor D ni pravilen, saj pot ne prečka nobene gore.



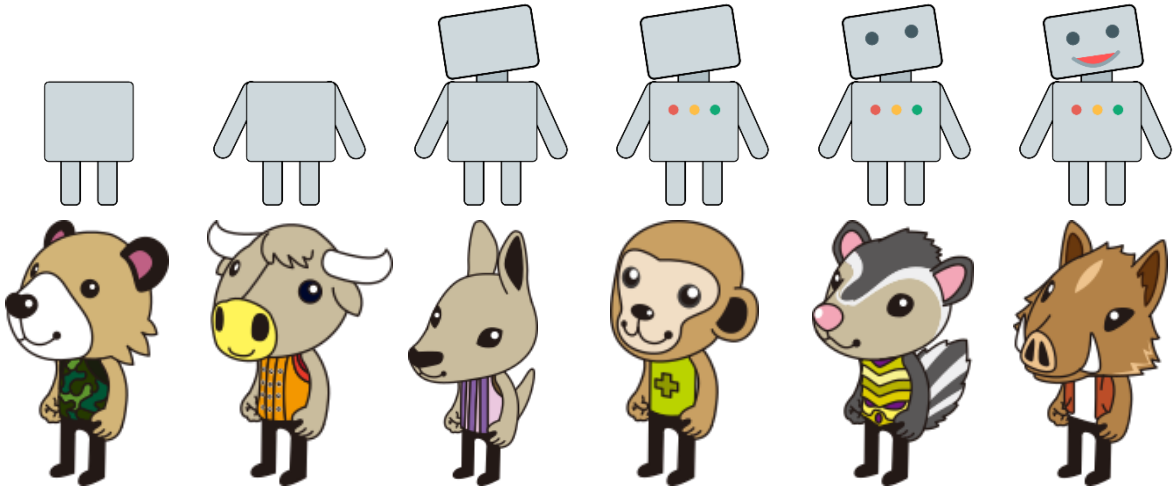
Računalniško ozadje

Pri programiranju pogosto naletimo na omejitve, ki jih moramo upoštevati. Te omejitve preverjamo s pogojnimi stavki.

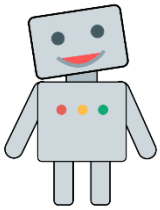
Sestavljanje robotov 2

3. in 4. razred

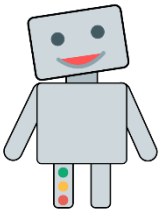
Šest živali dela v tovarni robotov. Vsaka od njih ima drugačno nalogo. Naloge opravljajo v naslednjem vrstnem redu:



Pravilno izdelan robot izgleda tako:



Ampak nekdo se je zmotil, zato novi robot izgleda takole:



Kdo se je zmotil?

Rešitev

Pravilen odgovor je  , saj je njena naloga, da doda na robotov trup lučke, ki jih je pomotoma dodala na robotovo nogo. Vsi ostali deli robota so na pravih mestih, saj so ostale živali svoje delo opravile brez napak.

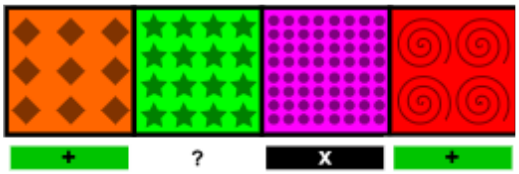
Računalniško ozadje

Pri programiranju pogosto prihaja do napak. Iskanju napak v programski kodi pravimo razhroščevanje.

Ugotovi geslo

3. in 4. razred

V igri Ugotovi geslo igralci iščejo skrito geslo, sestavljeno iz 4 barvnih polj. Po vsakem ugibanju igralec izve, na katerih mestih so pravilne barve na naslednji način:

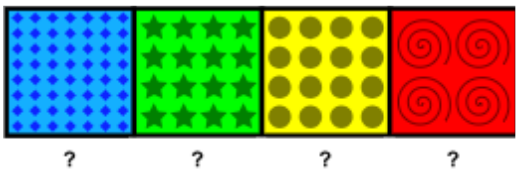


Oznake pod barvami pomenijo:

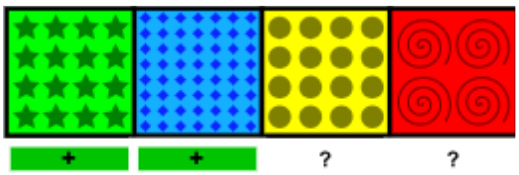
+	Barva je na pravem mestu.
?	Pravilna barva, ampak na napačnem mestu.
X	Te barve ni v geslu.

Mija je ugibala dvakrat in dobila naslednje odzive:

1. poskus:

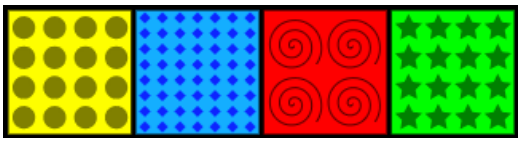


2. poskus:

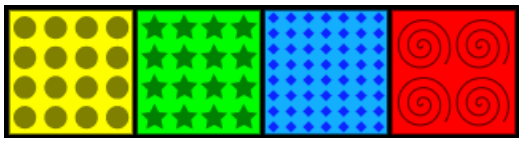


Kakšno je pravilno geslo?

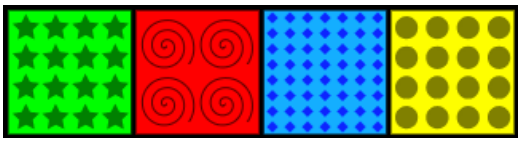
A)



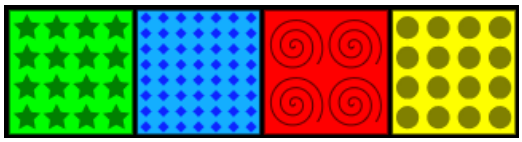
B)



C)

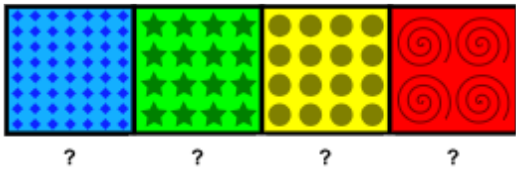


D)



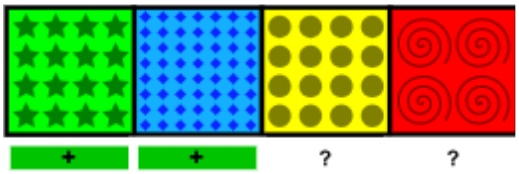
Rešitev

1. poskus:



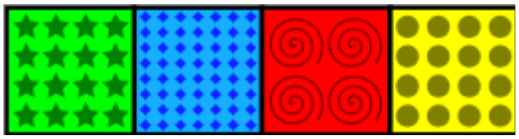
Iz 1. poskusa izvemo, da so v geslu modra, zelena, rumena in rdeča barva, ampak ne v tem vrstnem redu.

2. poskus:



Iz 2. poskusa izvemo, da sta na 1. mestu zelena in na 2. mestu modra barva, rumena ni na 3. mestu, rdeča pa ne na 4.

Na 1. mestu mora torej biti zelena barva, na 2. modra barva, 3. in 4. mesto pa moramo zamenjati, da je na 3. mestu rdeča in na 4. mestu rumena barva. Tako dobimo vrstni red zelena, modra, rdeča in rumena barva, kar ustreza odgovoru D:



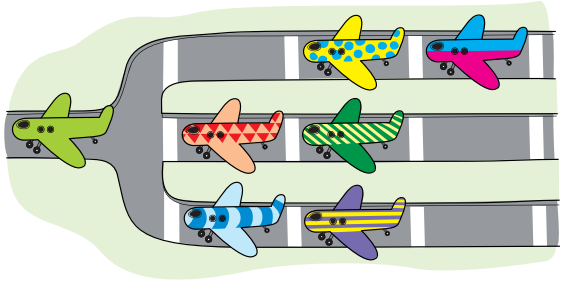
Računalniško ozadje

V računalništvu za dostop do podatkov pogosto uporabljamo gesla, s katerimi zavarujemo podatke, ki jih ne želimo deliti z neznanci. Pri tem moramo paziti, da gesel ni tako preprosto uganiti kot pri tej igri.

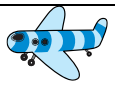
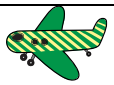
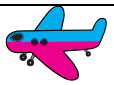
Letala

4. in 5. razred

Na sliki vidiš sedem letal, ki čakajo v 3 vrstah za vzlet z ene vzletne steze. Letala ne morejo preskakovati vrste.



Tu je urnik vzletov, na katerem so nekatera letala že vrisana, ostala pa moraš še dodati:

Čas vzleta	10.45	10.50	10.55	11.00	11.05	11.10	11.15
Letalo							

V kakšnem vrstnem redu lahko vzletijo letala ob upoštevanju urnika?

A) 

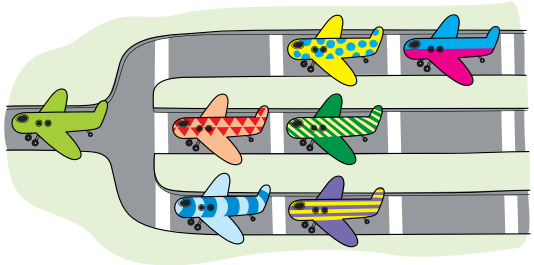
B) 

C) 

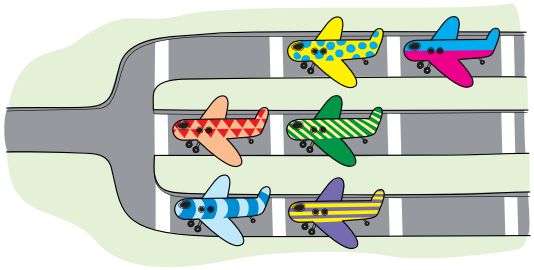
D) 

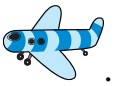
Rešitev

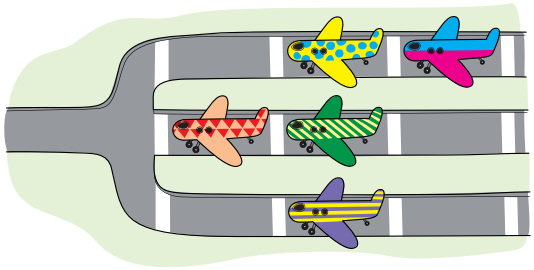
Na začetku so letala razporejena tako:



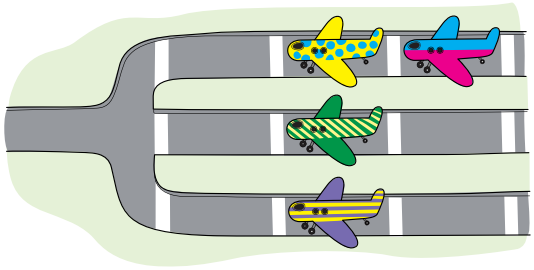
S slike vzletne steze lahko razberemo, da zeleno letalo  že stoji na vzletni stezi, zato bo vzletelo kot prvo ob 10.45. Po njegovem vzletu je stanje tako:



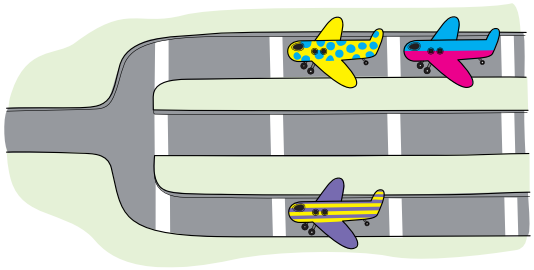
Iz urnika vzletov razberemo, da bo ob 10.50 vzletelo modro letalo  . Po vzletu ostanejo naslednja letala:



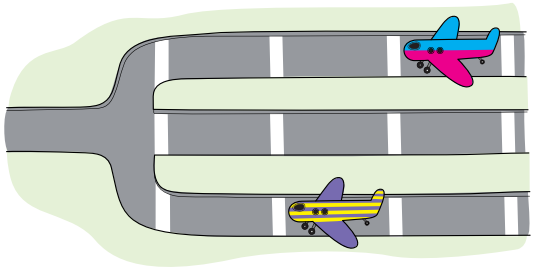
Ob 10.55 lahko vzletijo le prva letala iz vsake vrste, to so: rumeno letalo  , rdeče letalo  in vijolično letalo  . Ker mora po urniku ob 11.00 vzleteli temno zeleno letalo  , pred njim pa je vzlet le enega letala, ob 10.55 poleti rdeče letalo  . Ko vzleti, je stanje na letališču takšno:



Ob 11.00 vzleti temno zeleno letalo  . Zdaj imamo pred vzletno stezo naslednja letala:



Ob 11.05 lahko poletita le letali, ki sta prvi v obeh vrstah, in sicer rumeno letalo in vijolično letalo . Ker po urniku ob 11.10 poleti modro-roza letalo , ob 11.05 vzleti letalo, ki je pred njim v vrsti, to je rumeno letalo . Po vzletu ostaneta na letališču 2 letali:



Ob 11.10 po urniku vzleti modro-roza letalo in kot zadnje ob 11.15 vzleti še vijolično letalo .

Pravilen odgovor je tako D.

Čas vzleta	10.45	10.50	10.55	11.00	11.05	11.10	11.15
Letalo							

Računalniško ozadje

Računalnik namesto letal razvršča procese, ki jih mora izvesti. Nekateri od teh procesov so že časovno določeni, ostali pa se izvedejo glede na čas prihoda.

Nariši pošast 2

4. in 5. razred

Prijatelji igrajo igro Nariši pošast. Imajo 5 kock: rdečo, rumeno, zeleno, modro in vijolično. Barva kocke skupaj s številom pik na njej določa lastnosti pošasti, in sicer:

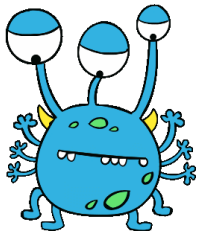
	Rdeča kocka	Število oči
	Rumena kocka	Število rogov
	Zelena kocka	Število rok
	Modra kocka	Število parov zob
	Vijolična kocka	Število nog
	Vsota najmanjših dveh števil	Število lis

Kocke so padle tako:

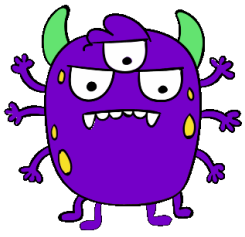
Rdeča kocka **Rumena kocka** **Zelena kocka** **Modra kocka** **Vijolična kocka**

Glede na zgornje vrednosti kock so prijatelji narisali štiri pošasti. Katera slika upošteva vsa navodila?

A)



B)



C)

D)

Rešitev

Iz vrednosti kock izvemo, da mora pošast imeti:

- Rdeča kocka: 3 oči
- Rumena kocka: 2 roga
- Zelena kocka: 6 rok
- Modra kocka: 3 pare zob, to je 3 x 2 = 6 zob
- Vijolična kocka: 4 noge

- Najmanjši dve vrednosti sta 2 in 3, 2 + 3 = 5, zato ima pošast 5 lis.

Pošast na sliki A ni narisana po navodilih, saj ima 6 lis.

Pošast na sliki B je narisana po vseh navodilih.

Pošast na sliki C ima le dve nogi in pet zob.

Pošast na sliki D nima rok, ima le dve nogi in nima lis.

Edina slika, ki upošteva vsa navodila, je B.

Računalniško ozadje

Tako kot računalnik sledi le podanim navodilom, so to storili tudi prijatelji v nalogi. Pošasti so bile različne, ampak prijatelji niso natančno sledili navodilom, zato ste pri narisanih pošastih našli napake. Iskanju in odpravljanju napak pri programiranju rečemo razhroščevanje.

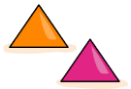
Navodila za gradnjo 2

4. in 5. razred

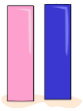
Na voljo imaš:

6 kock

1 most



2 piramidi

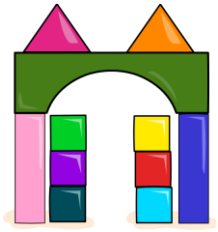


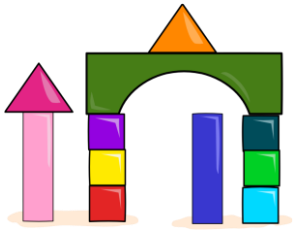
2 kvadra

Prijatelj ti da naslednja navodila:

1. Vzemi 3 kocke.
2. Kocke postavi eno na drugo, da zgradiš stolp.
3. S preostalimi tremi kockami zgradi še en stolp.
4. Poleg stolpov iz kock postavi 2 kvadra.
5. Na zgrajeno strukturo postavi most.
6. Vzemi 2 piramidi in ju postavi na vrh strukture.

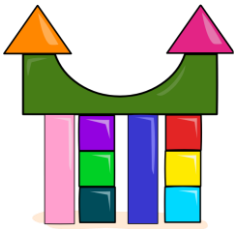
Po teh navodilih lahko zgradiš raznolike strukture, kot sta na primer ti dve:



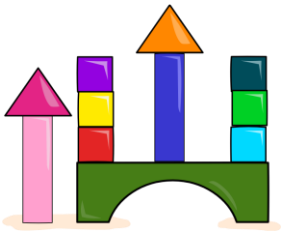


Katere od naslednjih struktur **ne moreš** zgraditi, če upoštevaš zgornja navodila?

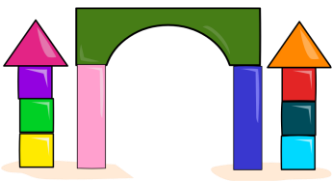
A)



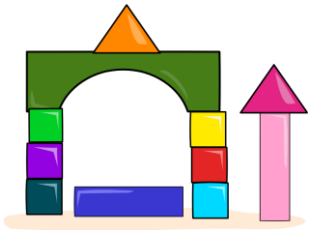
B)



C)

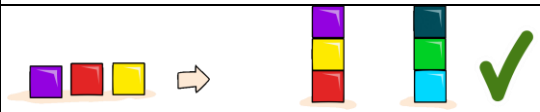
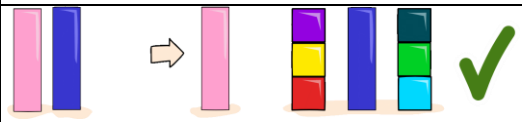
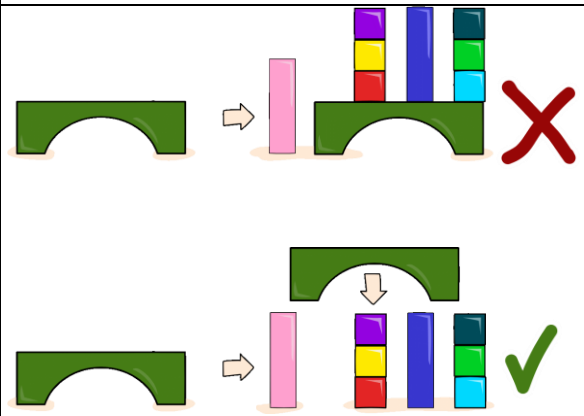


D)



Rešitev

Strukture A, C in D lahko zgradiš, strukture B pa ne. Poglejmo, kje je težava:

1. Vzemi 3 kocke. 2. Kocke postavi eno na drugo, da zgradiš stolp.	
3. S preostalimi tremi kockami zgradi še en stolp.	
4. Poleg stolpov iz kock postavi 2 kvadra.	
5. Na zgrajeno strukturo postavi most.	

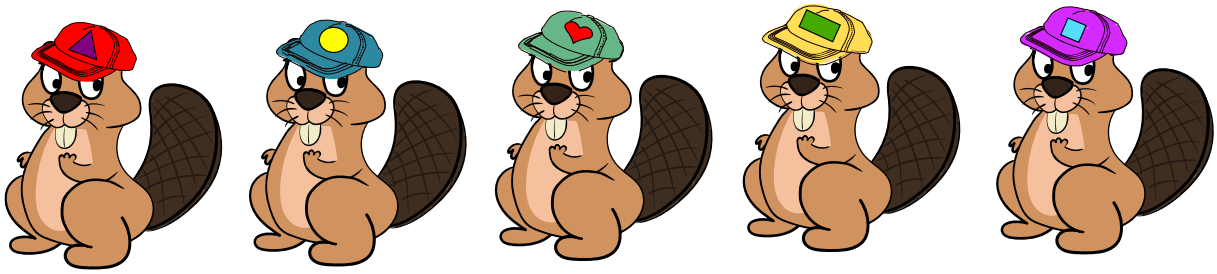
Računalniško ozadje

Računalnik zna slediti le natančnim navodilom in naredi le tisto, kar mu naročimo.

Tekma

4. in 5. razred

Pet bobrov tekmuje v teku. Tečejo eden za drugim, od desne proti levi. Vsak bober nosi kapo z drugačnim našitkom. Po prvi minuti tečejo bobri v naslednjem vrstnem redu:



1. 2. 3. 4. 5.

Med tekmo se zgodijo naslednji premiki:

1) Bober z modrim kvadratom na kapi se premakne za dve mesti naprej.
2) Nato se bober z zelenim pravokotnikom na kapi premakne za eno mesto naprej.
3) Nato se bober z rdečim srcem na kapi premakne za dve mesti naprej.

Po tem se tekma konča. Kakšen je vrstni red na koncu tekme?

A)


B)


C)


D)


Rešitev

Po prvi minuti bobri tečejo v naslednjem vrstnem redu:

V 1. premiku se bober z modrim kvadratnim našitkom na kapi premakne za 2 mesti naprej, zato je vrstni red takšen:

V 2. premiku se bober z zelenim pravokotnim našitkom premakne za 1 mesto naprej in vrstni red je takrat takšen:

V 3. premiku se bober s srcem na kapi premakne za 2 mesti naprej in vrstni red postane takšen:

To je tudi zadnji premik na tekmi, zato je tak tudi končni vrstni red. Pravilen je odgovor C.

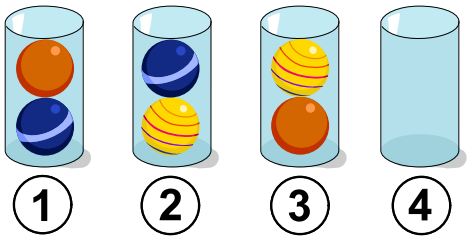
Računalniško ozadje

V tej nalogi so bobri postavljeni v urejeno vrsto, tako da vsak od njih zaseda svoje mesto. Tudi pri programiranju je vrstni red podatkov lahko pomemben. Za shranjevanje takih podatkov lahko uporabimo sezname.

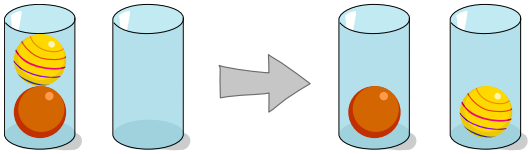
Bonboni

4. do 7. razred

David je kupil 4 škatlice s po dvema bonbonoma. Pojedel je oba bonbona iz zadnje škatlice, ki je sedaj prazna. Preostali bonboni so treh barv.



David želi bonbone prerazporediti tako, da bosta bonbona istih barv v isti škatlici. Premakne lahko le tisti bonbon iz vsake škatlice, ki nima nad sabo nobenega drugega bonbona. V vsaki škatlici sta lahko največ dva bonbona naenkrat.



S katero kombinacijo potez lahko David pride do želene razvrstitve?

A) Premik iz 1. v 4. škatlico - Premik iz 3. v 1. škatlico - Premik iz 3. v 4. škatlico - Premik iz 2. v 3. škatlico
B) Premik iz 2. v 4. škatlico - Premik iz 3. v 2. škatlico - Premik iz 1. v 3. škatlico - Premik iz 1. v 4. škatlico
C) Premik iz 3. v 4. škatlico - Premik iz 2. v 4. škatlico - Premik iz 1. v 3. škatlico - Premik iz 1. v 2. škatlico
D) Premik iz 2. v 4. škatlico - Premik iz 1. v 2. škatlico - Premik iz 3. v 1. škatlico - Premik iz 1. v 4. škatlico

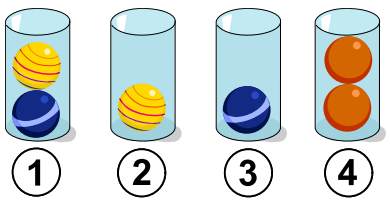
Rešitev

Pravilen odgovor je B:

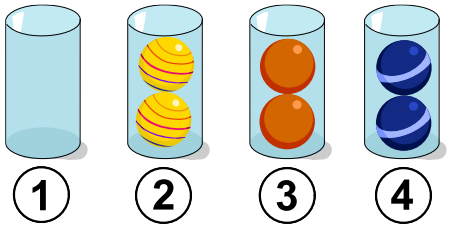
Premik iz 2. v 4. škatlico
Premik iz 3. v 2. škatlico
Premik iz 1. v 3. škatlico
Premik iz 1. v 4. škatlico

Nalogo seveda lahko rešimo na več možnih načinov, od ponujenih odgovorov pa le B pripelje do prave razvrstitve. Namreč, končna stanja po izvedbi ukazov so naslednja:

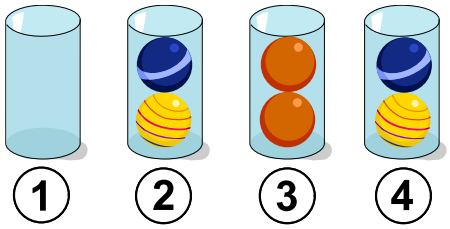
- razvrstitev po izvedbi ukazov A:



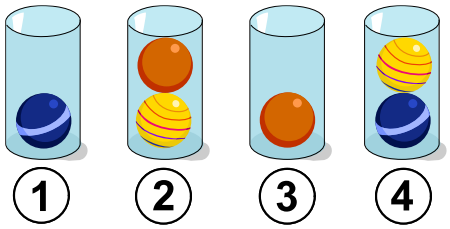
- razvrstitev po izvedbi ukazov B:



- razvrstitev po izvedbi ukazov C:



- razvrstitev po izvedbi ukazov D:



Računalniško ozadje

V tej nalogi škatlice ponazarjajo sklad. Sklad je podatkovna struktura, ki temelji na načelu »zadnji noter, prvi ven« ali LIFO (angl. *last in, first out*). V našem primeru je škatlica sklad, bonbon pa je podatek, shranjen v njem. Ko izvedemo ukaz »premik«, iz škatlice vzamemo vrhnji (»zadnji«) bonbon in ga vstavimo na vrh druge škatlice.

Skoki

4. do 7. razred

Pri igri Skoki sestavimo igralno ploščo iz osmih različnih kart. Vsaka karta je označena s številko in puščico, ki kaže bodisi desno bodisi levo.

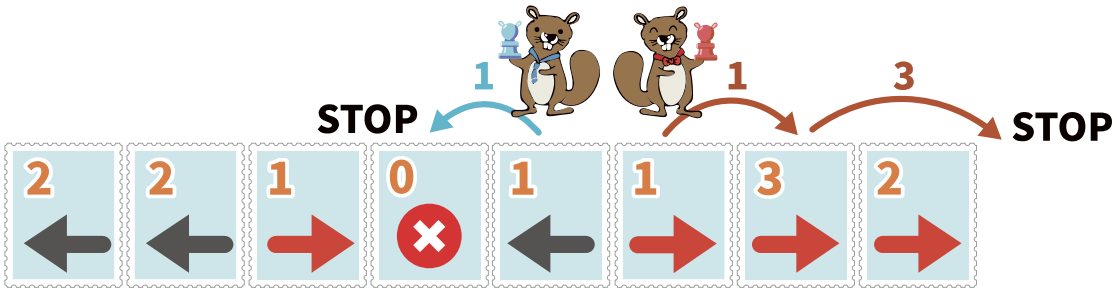
Vsak igralec izbere, na katero karto bo najprej postavil svojo figuro.

Nato skače s karte na karto, kot kažejo puščice (smer skoka) in številke (dolžina skoka). Igralec konča igro, ko pristane na karti s številko 0 ali izven igralne plošče.

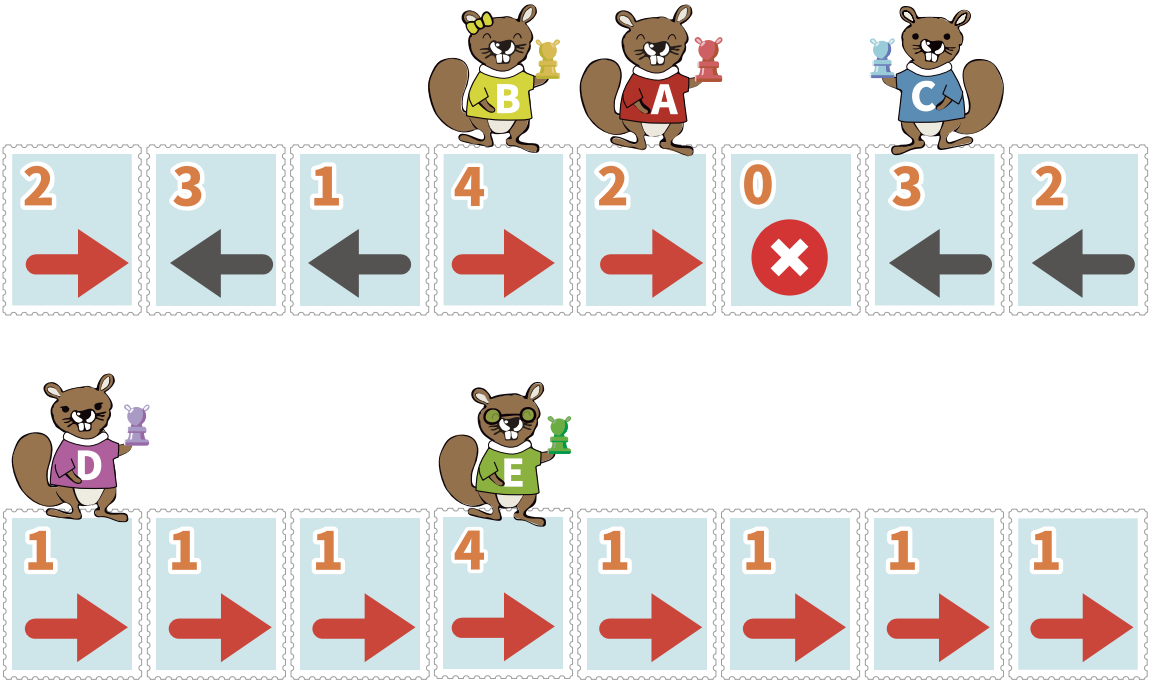
Igralec točke sešteva tako, da sešteva točke na vsaki karti, na kateri pristane, vključno s karto, na kateri začne.

Igralec, ki doseže največ točk, je zmagovalec.

Na primer: modri bober na sliki je dosegel 1 točko, rdeči bober pa 4 točke.



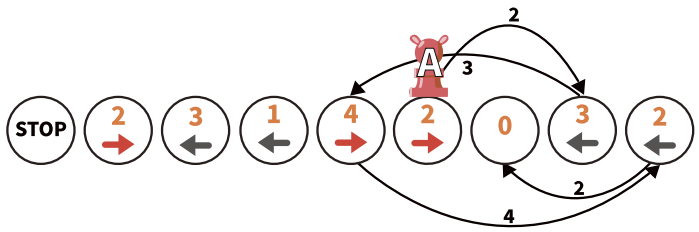
Danes igra pet igralcev na dveh igralnih ploščah. Svoje figure (bobre A, B, C, D in E) so postavili na začetna mesta:



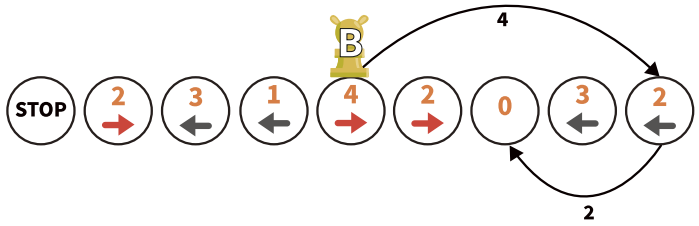
Kateri bober bo zmagovalec?

Rešitev

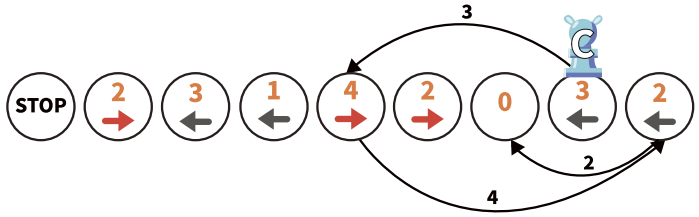
Bober A doseže 11 točk:



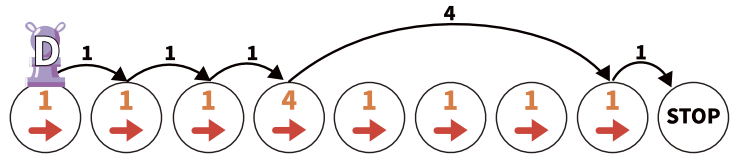
Bober B doseže 6 točk:



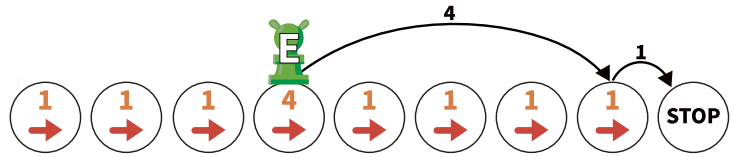
Bober C doseže 9 točk:



Bober D doseže 8 točk:



Bober E doseže 5 točk:



Največ točk doseže bober A.

Računalniško ozadje

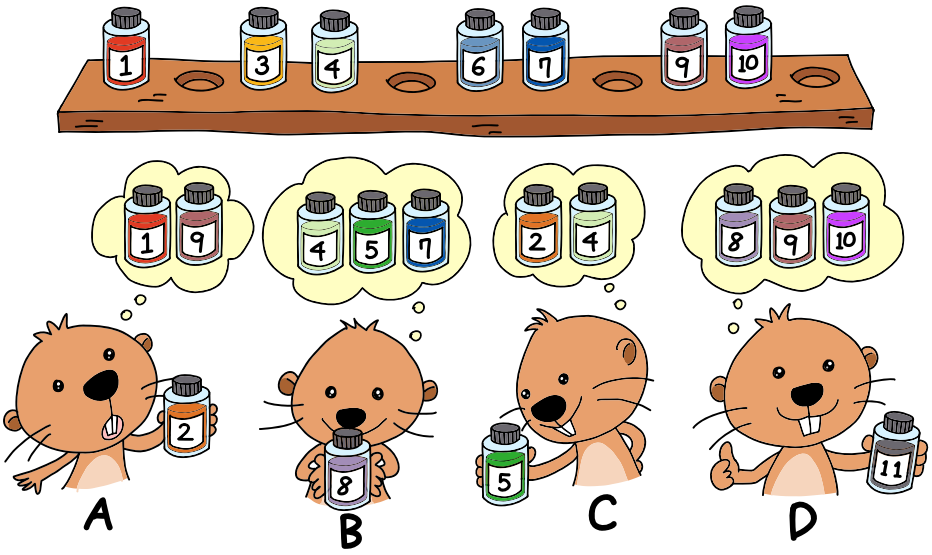
Skoke med posameznimi mesti lahko prikažemo z usmerjenim grafom.

Peščene slike

4. do 9. razred

Učenci ustvarjajo slike iz barvnega peska. Najprej na papir nanesejo lepilo, nato pa nanj stresejo pesek. Ker se lepilo hitro suši, morajo že pred začetkom dela vzeti pesek v barvah, ki jih potrebujejo.

Vsaka barva peska je shranjena v svojem kozarcu, oštevilčenem s številkami od 1 do 11. Učitelj podaja kozarce peska učencem. Na sliki vidimo številke kozarcev, ki jih držijo v rokah Aleks, Brina, Cvetka in Davor, v oblačkih pa je narisano, katere kozarce še potrebujejo.



Vsak učenec mora pred začetkom dela dobiti vse barve peska, ki jih potrebuje za svojo sliko. Ko dokonča sliko, vrne kozarce. Če kozarec uporablja že kdo drug, počaka, da ta vrne kozarec peska.

Kdo od učencev bo zadnji dokončal svojo sliko?

Rešitev

Aleks ima v rokah kozarec 2, potrebuje še kozarca 1 in 9, ki sta še na polici, zato lahko začne z delom takoj.

Brina ima v roki kozarec 8, potrebuje še kozarce 4, 5 in 7. Ker je kozarec 5 pri Cvetki, mora počakati, da ta dokonča svojo sliko.

Cvetka ima v roki kozarec 5 in potrebuje kozarca 2 in 4. Kozarec 2 ima trenutno Aleks, zato lahko Cvetka dokonča sliko za njim. Ko Cvetka konča, lahko sliko ustvari Brina, saj ji od zasedenih manjka le kozarec 5.

Davor ima kozarec 11, potrebuje še kozarce 8, 9 in 10. Ker ima kozarec 8 v rokah Brina, mora počakati, da ona dokonča svojo sliko.

Vrstni red je zato naslednji: Aleks, Cvetka, Brina in zadnji bo Davor.

Računalniško ozadje

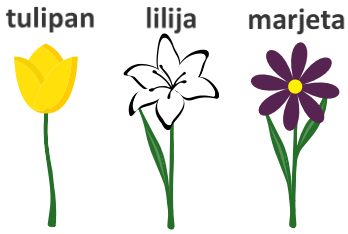
Naloga prikazuje situacijo, v kateri bi lahko prišlo do tako imenovanega smrtnega objema. Do smrtnega objema bi prišlo, če bi dva učenca čakala eden na drugega, da bi lahko dokončala svoji sliki. Na primer, če bi Brina vzela kozarec 4, potrebuje še kozarec 5, ki ga ima Cvetka, ki potrebuje kozarec 4. Nobena od njiju ni pripravljena drugi prepustiti kozarca, ki ga potrebuje, dokler ne dokonča slike. Čakanje na drugi kozarec bi se lahko končalo z dogovorom ali pa s koncem šolske ure, ko bi učitelj pobral vse kozarce s peskom.

Emine rože

5. razred

Ema razvršča rože v tri vaze.

Ima tri vrste rož:



Vsaka roža je lahko rumena, bela ali vijolična.

Vsaka roža ima lahko 0, 1 ali 2 lista na steblu.

Ema vedno da v vse vaze enako število rož. Vedno tudi poskrbi, da so v vsaki vazi rože enake vrste, enake barve ali z enakim številom listov. V vseh vazah morajo biti razvrščene po enakem pravilu. Kot kaže slika, ima Ema trenutno 3 vaze s 6 rožami, za katere veljajo različna pravila (v prvi in drugi vazi so rože enake barve, v tretji pa so rože z enakim številom listov na steblu):

Vzela bo še 3 rože (na sliki desno) in preuredila vseh 9 rož v vaze tako, da bodo rože v **vseh vazah** urejene **po natanko enem** od zgoraj napisanih pravil.

Katera od možnosti prikazuje eno od vaz, ki so urejene po istem pravilu?

A)

B)

C)

D)

Rešitev

Pravilen odgovor je D.

Ema ima naslednjih 9 rož:

V vsaki vazi morajo biti po 3 rože.

Ema ima 2 beli roži, 3 vijolične rože in 4 rumene rože. Ker je število rož posamezne barve različno, rož ne more razporediti v vaze po barvi, kot so v primeru A.

2 roži sta brez listov, 3 rože imajo 1 list in 4 rože imajo po 2 lista na steblu, zato rož v vaze ne more razporediti niti po številu listov na steblu, kot so v primeru B.

Ker ima 3 tulipane, 3 lilije in 3 marjete, lahko rože v vaze razporedi po njihovi vrsti:

Torej je pravilen odgovor D, pri katerem so v vazi skupaj tri lilije.

Odgovor C ni pravilen, saj ne sledi nobenemu od pravil.

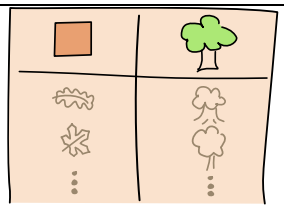
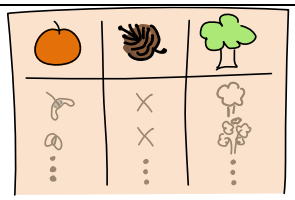
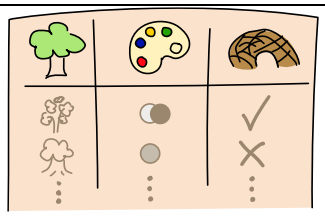
Računalniško ozadje

Pri objektnem programiranju se pogosto srečujemo z razredi. V našem primeru vse rože sodijo v isti razred, vsaka od rož pa ima svoje lastnosti: vrsto, barvo in število listov na steblu.

Bobri in les

5. do 7. razred

Emil in njegovi prijatelji obožujejo pohodništvo. Med pohodi njegovi prijatelji zbirajo podatke o drevesih, ki jih vidijo, in jih zapisujejo v dolge tabele:

	Severin zbira podatke o oblikah listov ( ) in pripadajočih vrstah dreves ( ).
	Katarina zbira podatke o plodovih ( ), pripadajočih vrstah dreves ( ) in ali gre za iglavce ( ).
	Lara zbira podatke o barvi lesa ( ), pripadajočih vrstah dreves ( ) in ali je les primeren za gradnjo bobrišč ( ).

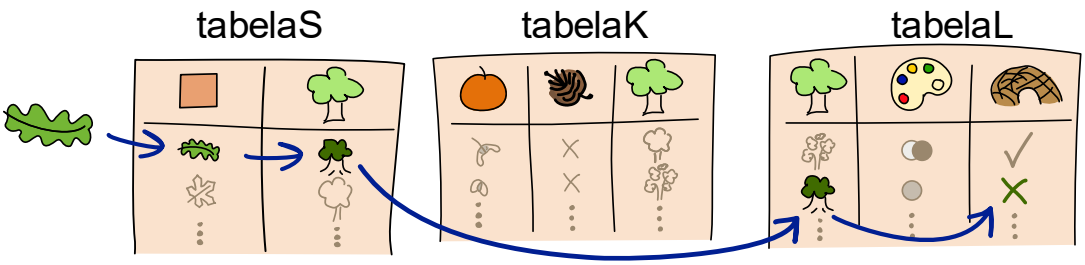
Emil je v gozdu našel drevesni list in pozna njegovo obliko. Rad bi vedel, ali je les te vrste dreves primeren za gradnjo bobrišč. Koga od svojih prijateljev mora vprašati in v kakšnem vrstnem redu, da bo našel odgovor?

Rešitev

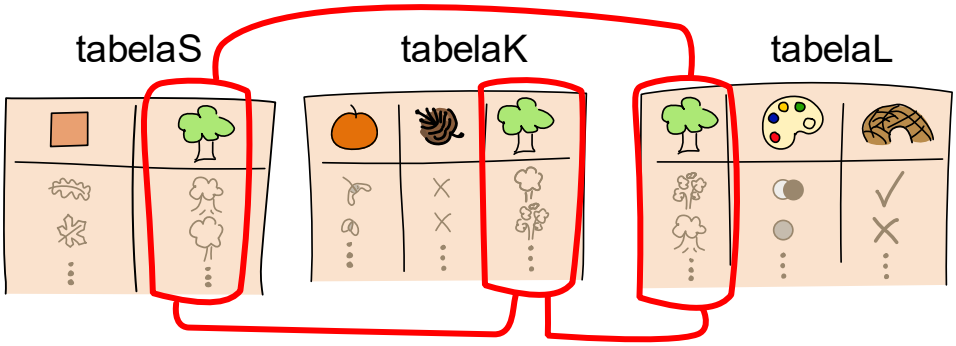
Najprej mora vprašati Severina, nato pa še Laro.

Podatke o primernosti lesa za gradnjo bobrišč ( ) lahko najdemo samo v Larini tabeli (tabelaL). Če Emil pozna samo list drevesa, ne bo mogel izbrati ustrezne vrstice iz Larine tabele. Potrebuje podatke o vrsti drevesa ( ) ali o barvi lesa ( ). Katarinina tabela (tabelaK) ne vsebuje podatkov o listu, medtem ko jih Severinova tabela vsebuje (tabelaS). Če Emil pozna obliko lista, lahko tega poišče v Severinovi tabeli in v najdeni vrstici z listom pridobi še manjkajoče podatke o vrsti drevesa ( ). Nato lahko to vrsto drevesa poišče v Larini tabeli in v najdeni vrstici pridobi še manjkajoče podatke o lesu.

Na primer, če je list hrastov, lahko Emil izbere pravilno vrstico v Larini tabeli in ugotovi, da hrastov les ni primeren za gradnjo bobrišč.



Tabele so med seboj povezane preko enega skupnega podatka: vrste dreves (). Če lahko iz katere koli tabele izbereš določen podatek o drevesu, imaš dostop do vseh ostalih podatkov v vseh tabelah.



Računalniško ozadje

Naloga temelji na iskanju informacij in povezovanju podatkov, podobno kot pri relacijskih bazah podatkov. Vsak prijatelj zbira določene podatke o drevesih, ti pa so organizirani v tabelah, ki hranijo različne lastnosti dreves. S povezovanjem tabel preko skupnih podatkov – te imenujemo ključi (v našem primeru je to vrsta drevesa) lahko pridobimo podatke iz vseh tabel.

Takšen pristop se uporablja v bazah podatkov, analizi znanstvenih podatkov in programskih aplikacijah, ki temeljijo na povezanih podatkih. Naloga tako ponazarja, kako iz omejenih podatkov dosežemo popolno rešitev.

Perzijska preproga

5. do 7. razred

Sara ima pletilnega robota, ki izdeluje preproge s štirimi različnimi vzorci. Vsak vzorec se robotu posreduje z dvomestno kodo:

00	
11	
01	
10	

Sara posreduje ta pletilni program robotu. Vsaka koda robotu pove, kateri vzorec naj vplete v posamezen kvadrat preproge. Robot plete od leve proti desni, od zgoraj navzdol. Katero od naslednjih preprog bo robot izdelal na podlagi Sarinega programa?

00-10-10-10-01-10-10-10-00

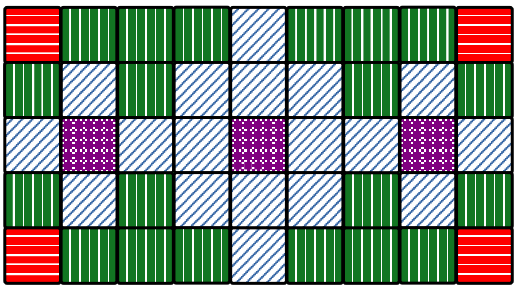
10-01-10-01-01-01-10-01-10

01-11-01-01-11-01-01-11-01

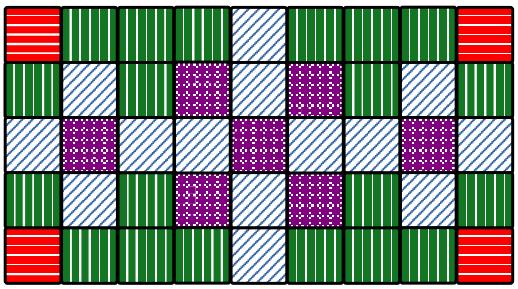
10-01-10-01-01-01-10-01-10

00-10-10-10-01-10-10-10-00

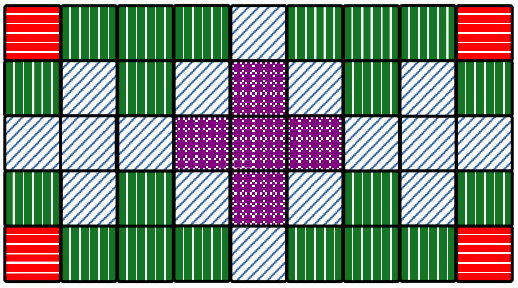
A)



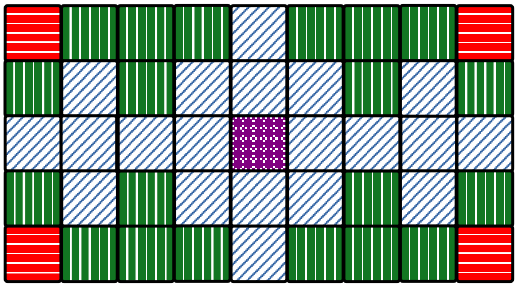
B)



C)



D)



Rešitev

Pravilen odgovor je A.

Za rešitev te naloge lahko dekodiramo celoten program vrstico po vrstico in ga nato primerjamo z možnostmi, da najdemo pravilno; vendar je ta metoda precej zamudna in bo vzela veliko časa.

1. Preberi program vrstico po vrstico, od leve proti desni, od zgoraj navzdol.
2. Razdeli program na vrstice in dekodiraj vsako dvomestno kodo z uporabo tabele kod.
3. Zgradi celoten vzorec preproge.
4. Primerjaj vzorec preproge s podanimi štirimi možnostmi in poišči tisto, pri kateri se vzorca ne razlikujeta.

Bolj učinkovita rešitev je pregled razlik med podanimi možnostmi, da omejimo obseg in hitro najdemo odgovor. Uporabimo lahko naslednje korake:

1. Določi položaje, kjer obstajajo razlike med možnostmi.
2. Poišči kodo za ta kvadrat v programu in določi pravilen vzorec v tabeli kod.
3. Odstrani možnosti, kjer je vzorec v tem kvadratu napačen.

Ponovi korake 1–3, dokler ne ostane le ena možnost, ki je pravilna.

Primer:

- Možnosti A in B se razlikujeta v 4. in 6. kvadratu druge in četrte vrstice. V programu je koda za 4. in 6. kvadrat druge in četrte vrstice je 01 (vzorec s poševnimi črtami); ta vzorec je tudi na preprogi A, ne pa na preprogi B. Zato odstranimo možnost B, ker ni pravilna.
- Možnosti C in D se razlikujeta v 4. in 6. kvadratu tretje vrstice. Koda za 4. in 6. kvadrat tretje vrstice je 01 (vzorec s poševnimi črtami); ta vzorec je na tem mestu v preprogi D, medtem ko ima preproga C na tem mestu vzorec 11 (s pikami). Zato odstranimo tudi možnost C.
- Ostaneta le še možnosti A in D. Ti dve možnosti se razlikujeta v 2. in 8. kvadratu tretje vrstice. V programu je koda za 2. in 8. kvadrat tretje vrstice 11 (vzorec s pikami), a tega vzorca ni na tem mestu v preprogi D. Zato odstranimo tudi možnost D.
- Ostane nam le preproga A, ki mora ustrezati programu in je tako pravilna rešitev.

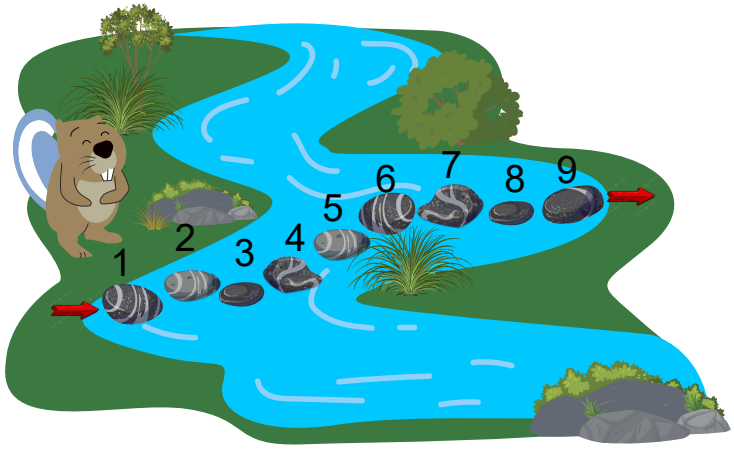
Računalniško ozadje

Računalniki barv ne shranjujejo z imeni, ampak z binarnimi kodami, sestavljenimi iz ničel in enic. Podobno ta naloga uporablja dvomestne binarne kode, ki predstavljajo različne vzorce, zaporedje teh kod pa robotu pove, kako izdelati preproge. V računalniku se vse – dokumenti, fotografije, glasba, videi in številke – pretvori v binarno kodo, da se lahko prikazuje, shranjuje in prenaša. Naloga pokriva algoritme, dekodiranje in logično razmišljanje, saj gre za pretvorbo kod v vizualne vzorce. Računalnik preverja vsak kvadratek vrstico po vrstico izredno hitro, medtem ko človek pri tem porabi več časa, lahko pa nalogo reši hitreje, če opazi razlike med možnostmi in preveri le te.

Prečkanje reke

5. do 7. razred

Ob vstopu v narodni park je Gregor prejel kodo za prečkanje reke s skakanjem s kamna na kamen.



Kako prebrati kodo?

A pomeni skoči 2 kamna naprej.

- Če je Gregor že na kamnu 9, ostane na njem.
- Če je Gregor na kamnu 8, lahko skoči na ciljno obalo.

B pomeni pojdi 1 kamen nazaj.

- Če je Gregor na začetni obali, ostane, kjer je.

Na primer, če sledi naslednji kodi:

B↷A↷A↷A↷B↷A↷A↷B↷A

bo Gregor prečkal reko na naslednji način:

B ↷	A ↷	A ↷	A ↷	B ↷	A ↷	A ↷	B ↷	A
Ostani na mestu	Kamen 2	Kamen 4	Kamen 6	Kamen 5	Kamen 7	Kamen 9	Kamen 8	Na ciljni breg

Katera od spodnjih kod bo prav tako Gregorja pripeljala na ciljni breg?

A) **B↷B↷A↷A↷A↷B↷A↷A**

B) **A↷A↷B↷B↷A↷A↷A↷B**

C) **A↷A↷A↷A↷B↷B↷A↷A**

D) 

E) 

Rešitev

Pravilen odgovor je C.

Pri odgovoru A je na prvih dveh mestih kode znak B, zato se Gregor še ne premakne in je na začetnem bregu. Sledi trikrat znak A, kar pomeni, da se Gregor premakne trikrat po 2 kamna naprej in pristane na kamnu 6. Sledi znak B, zato stopi nazaj na kamen 5. Nato sledita še 2 znaka A, tako da se premakne dvakrat po 2 kamna naprej, to je na kamen 9, kjer se koda konča.

Pri odgovoru B sta najprej dva znaka A, kar pomeni, da se Gregor premakne z začetnega brega na kamen 4. Sledita 2 znaka B, zato se premakne za 2 kamna nazaj na kamen 2. Sledijo 3 znaki A, zato skoči najprej na kamen 8. Na koncu se zaradi zadnjega znaka B premakne nazaj na kamen 7.

Pri odgovoru C so najprej 4 znaki A, zato se premakne z začetnega brega na kamen 8, nato sledita 2 znaka B, tako da se premakne nazaj na kamen 6. Na koncu sta še 2 znaka A: prvi ga pripelje na kamen 8, drugi pa na ciljni breg.

Pri odgovoru D so v kodi samo znaki B, kar pomeni, da se Gregor sploh ne premakne z začetnega brega.

Pri odgovoru E je najprej znak A, zato se premakne na kamen 2. Sledi znak B, kar pomeni, da se premakne na kamen 1. Sledita 2 znaka A, kar pomeni, da se premakne na kamen 5. Sledita še 2 znaka B, kar pomeni, da se premakne 2 kamna nazaj, to je na kamen 3. Na koncu sta še 2 znaka A, kar pomeni, da se premakne na kamen 7, kjer tudi ostane.

Računalniško ozadje

Tako kot si ti sledil kodi, sledi programski kodi tudi računalnik, ko izvaja posamezen računalniški program.

Robot v labirintu 1

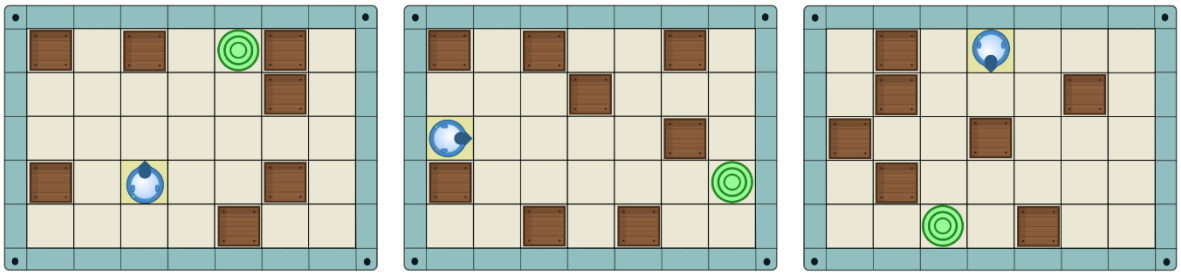
5. do 7. razred

Robot se premika po sobi, polni ovir, in mora priti do cilja. Premikanje robota nadzorujemo s tremi preprostimi ukazi:

- ***Naprej***. Robot se premika naprej v smeri, v katero gleda (robot se bo premikal v desno), dokler ne zadene ob steno ali ob oviro. Takrat se ustavi. Robot se ustavi tudi, ko pride na cilj.
- ***Obrat levo***. Robot se na mestu obrne za 90° v levo (se ne premakne s polja).
- ***Obrat desno***. Robot se na mestu obrne za 90° v desno.

Robot bo sledil podanim ukazom po vrsti.

Imamo tri različne sobe. V vsaki je en robot, ki ga želimo pripeljati do cilja s čim manj ukazi. Katero zaporedje ukazov (program) bo robote v vseh treh sobah uspešno pripeljalo do cilja?



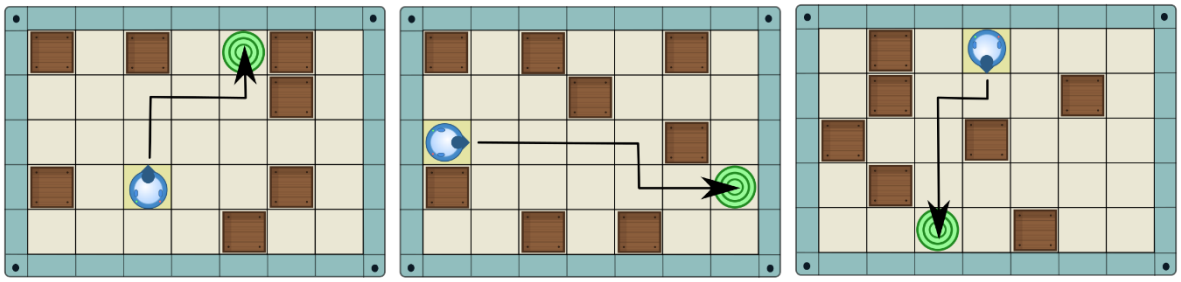
Rešitev

Zaporedje ukazov, ki vse tri robote uspešno spravi do cilja, je:

Naprej. Obrat desno. Naprej. Obrat levo. Naprej.

Robot v vsaki sobi lahko pride do cilja z uporabo različnih ukazov. Robot v prvi (levi) sobi bi na primer lahko prejel tudi ukaze Obrat desno, Naprej, Obrat levo in Naprej, ki bi ga pripeljali do cilja. To zaporedje ukazov je še krajše.

A le eno zaporedje ukazov (to je Naprej, Obrat desno, Naprej, Obrat levo in Naprej) pripelje do cilja vse tri robote. Poti robotov so prikazane s črno črto.



Za podane sobe je to tudi edina rešitev, saj iščemo najkrajše tako zaporedje ukazov.

In kako ugotovimo, da je to edina najkrajša rešitev? Opazimo lahko naslednje:

1. Robot gleda v določeno smer in potrebuje ukaz, da se obrne.
2. Cilj je v vseh teh sobah desno od robota. Torej mora rešitev vključevati vsaj en obrat v desno.
3. Po prvi oviri in obratu v desno je cilj levo od robota. Torej mora rešitev vključevati tudi vsaj en obrat v levo.

Tako mora najkrajša rešitev vsebovati najprej obrat v desno, nato obrat v levo, vmes pa so premiki naprej. Zaradi robota v drugi (srednji) sobi hitro ugotovimo, da se zaporedje ukazov ne more začeti z obratom. Zadnji premik naprej bo robota pripeljal do cilja, kjer se bo ustavil.

Računalniško ozadje

V nalogi smo robota usmerjali s pomočjo treh enostavnih ukazov. Lahko rečemo, da ti ukazi sestavljajo enostaven programski jezik. Čeprav je nabor ukazov zelo omejen, lahko te ukaze poljubno krat ponovimo, kar nam omogoča izdelavo tudi bolj kompleksnih programov. Še vedno pa se ti programi izvajajo zaporedno, torej ukaz za ukazom v podanem vrstnem redu. Vsak ukaz se mora najprej zaključiti, preden se začne izvajati naslednji ukaz. Tako zaporedno izvajanje je eden od osnovnih konceptov pri programiranju.

Skrivno sporočilo

5. do 7. razred

Prijatelja Bruno in Pavel se vsak teden srečata na različnih krajih. Svoje mesto srečanja zakodirata z uporabo naslednje kvadratne mreže s črkami.

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

Bruno je Pavlu poslal šifrirano sporočilo z mestom srečanja za ta teden:

2L 1G 6De 2Do 1L 2Do 7L 1G 1De 1G

Da razvozla šifrirano sporočilo, mora Pavel narediti naslednje:

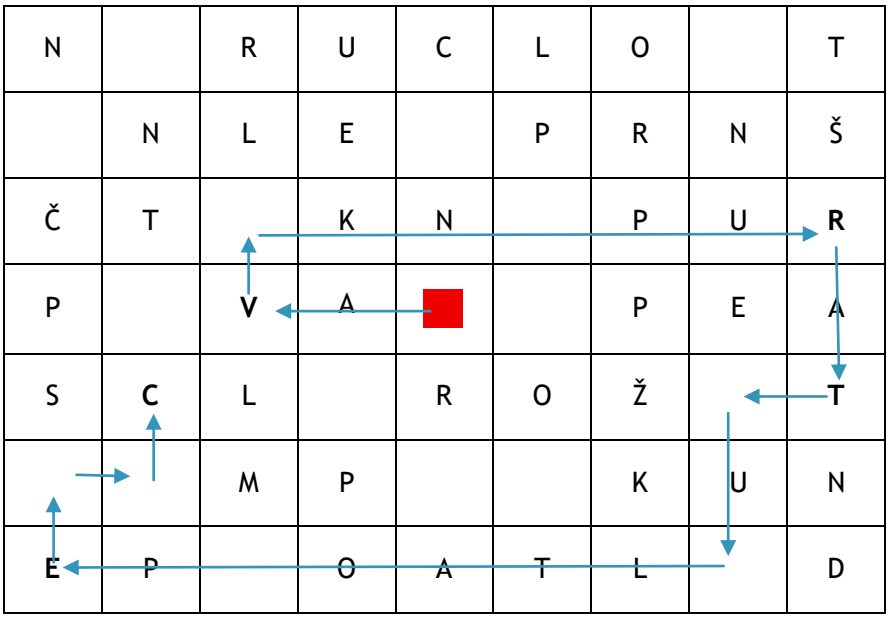
- Začne na rdečem kvadratu.
- Premika se po navodilih:
  - G – gor
  - Do – dol
  - De – desno
  - L – levo
- Številka pred vsako črko mu pove, za koliko kvadratov v tisto smer naj se premakne.
- Vsakič, ko navodilo zaključi na kvadratu s črko, jo zapiše.
- Črke prebere v zaporedju, v katerem jih je zbral, in tako odkrije kraj srečanja.

Kje se dobita?

Rešitev

Dobita se pri vrtcu.

Začnemo v rdečem kvadratu in sledimo navodilom v sporočilu. Posamezne puščice prikazujejo zaporedje navodil. V smeri poti zberemo črke V, R, T, E in C, da dobimo kraj srečanja.



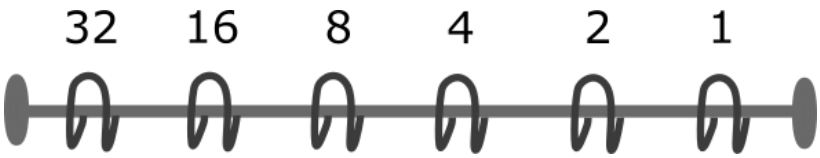
Računalniško ozadje

Šifriranje in dešifriranje spadata na področje kriptografije. Kriptografija se ukvarja z zaščito podatkov tako, da so dostopni samo tistim, ki imajo ustrezne ključe ali navodila za njihovo razvozlanje. V praksi se uporablja pri varnem prenosu podatkov na internetu, elektronskem bančništvu, zaščiti gesel, digitalnih podpisih in številnih drugih področjih, kjer je varnost podatkov ključnega pomena.

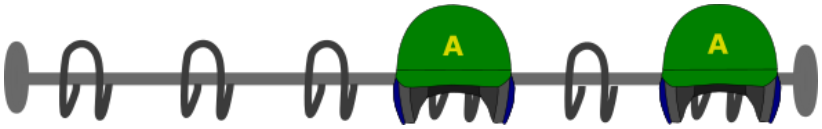
Štetje točk

5. do 7. razred

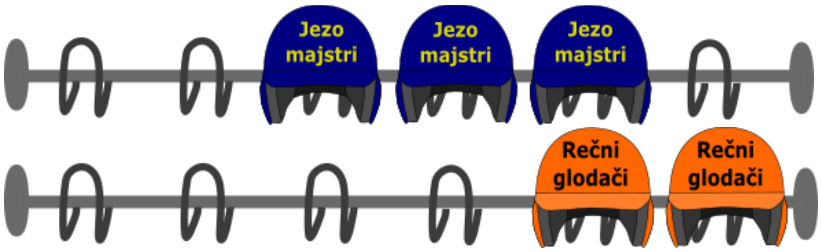
Bobri obožujejo bejzbol. Ker ne znajo pisati na papir, za beleženje točk uporabljajo sistem z obešanjem bejzbolskih čepic na obešalne kljukice. Vsaka kljukica ima določeno vrednost točk, ki jih prikazuje spodnja slika.



Dobljene točke ekipe preberejo tako, da seštejejo vrednosti pri vseh kljukicah, na katerih so obešene čepice te ekipe. Na primer, pet točk prikažejo tako, da postavijo čepici na mesto z vrednostma 4 točke in 1 točka:



Ekipi Jezomajstri in Rečni glodači sta v nedeljo odigrali tekmo. To so dosežene točke obeh ekip:



S kakšno razliko v točkah je zmagala zmagovalna ekipa?

Rešitev

Pravilen odgovor je 11 točk.

Ekipa Jezomajstrov je zbrala 8 + 4 + 2 = 14 točk, ekipa Rečnih glodačev pa 2 + 1 = 3 točke. Razlika je torej 11 točk.

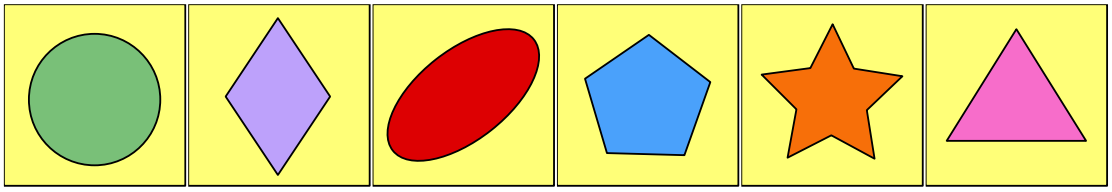
Računalniško ozadje

Bobri za beleženje točk uporabljajo dvojiški sistem, kot ga uporablja računalnik za shranjevanje in obdelavo podatkov.

Kocka

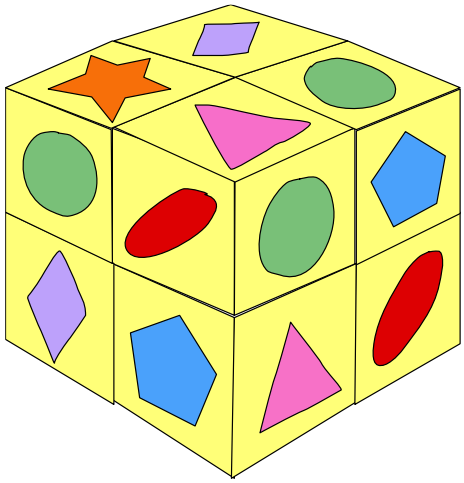
6. do 7. razred

Bebrasova kocka ima na vsaki ploskvi različno obliko. Vseh šest oblik na Bebrasovi kocki je prikazanih spodaj.



Zelen krog	Vijolični romb	Rdeča elipsa	Moder petkotnik	Oranžna zvezda	Roza trikotnik

Osem enakih Bebrasovih kock je zloženih skupaj v eno veliko kocko.



Katera oblika je na ploskvi nasproti zelenega kroga na vsaki Bebrasovi kocki?

Rešitev

Nasproti zelenega kroga na Bebrasovi kocki je vijolični romb.

V skladu osmih kock vidimo ob zelenem krogu naslednje oblike: oranžno zvezdo (zgoraj levo), rdečo elipso in roza trikotnik (zgoraj na sredini) ter moder petkotnik (zgoraj desno). Nobena od teh oblik ne more biti nasproti zelenemu krogu. Edina oblika, ki se nikoli ne pojavi ob zelenem krogu, je vijolični romb.

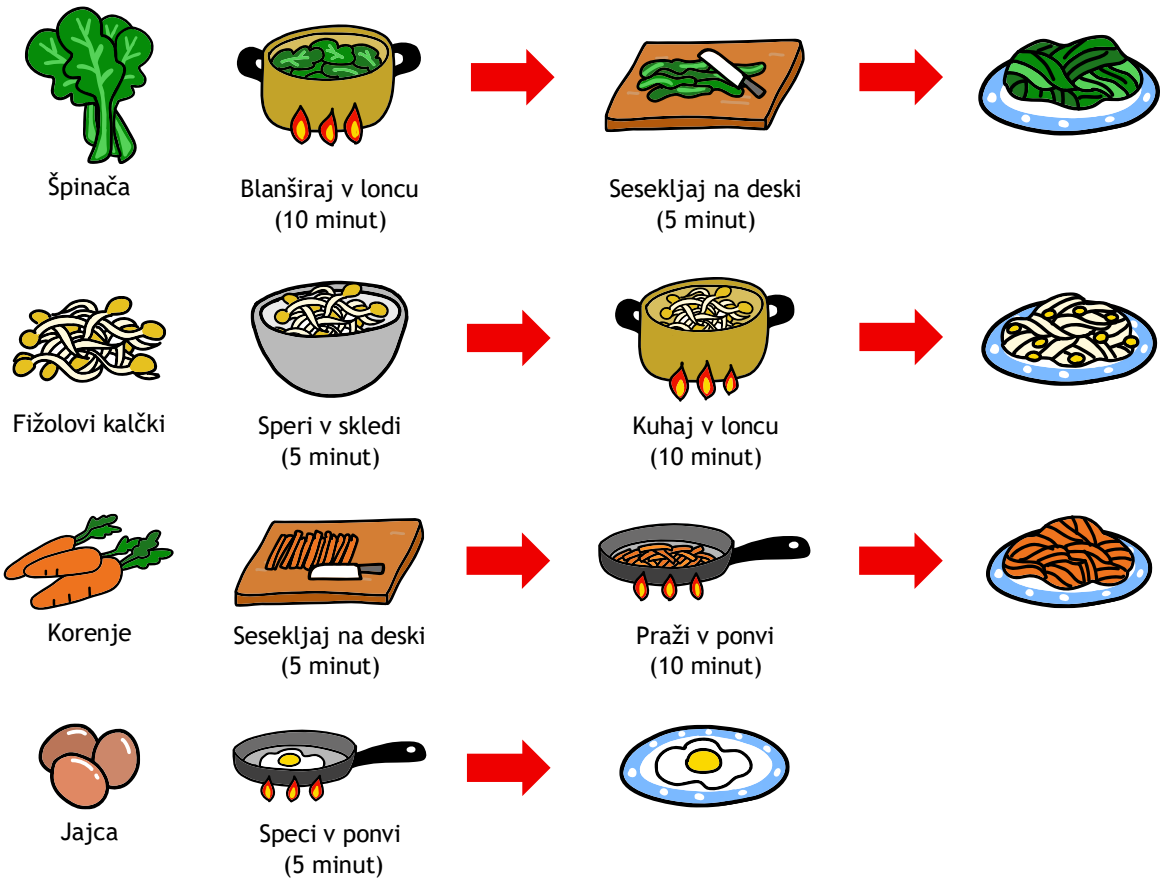
Računalniško ozadje

Izločanje možnosti, da bi prišli do pravilne izbire, je temeljna veščina pri računalništvu, na primer pri razhroščevanju, da odkrijemo glavni vzrok napake. Prostorsko razmišljanje je prav tako pomembno v računalništvu, na primer za natančno prikazovanje tridimenzionalnih predmetov v računalniški grafiki.

Bibimbap

6. do 9. razred

Pri pripravi bibimbapa, tradicionalne korejske jedi, se različne sestavine kuhajo posebej in se nato združijo, da se jed dokonča.

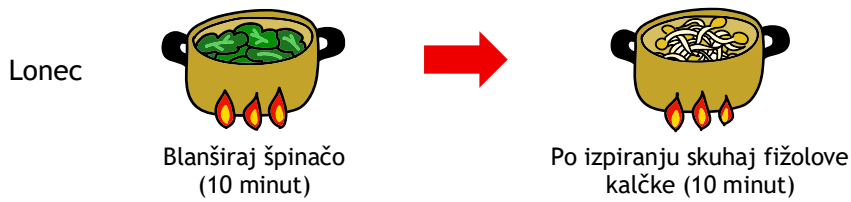


Kuhinja je opremljena s po enim loncem, ponvijo, desko in skledo. Šef kuhar Bober je zelo spreten pri opravljanju več nalog hkrati. Na primer, hkrati lahko 10 minut blanšira v loncu in 5 minut peče jajce na ponvi. Najmanj koliko časa potrebuje kuhar Bober za pripravo vseh štirih sestavin bibimbapa?

Rešitev

Kuhar Bober bo potreboval najmanj 20 minut.

Poglejmo, v kakšnem vrstnem redu bo uporabil posamezne kuharske pripomočke.



Uporabi lahko vse kuhinjske pripomočke hkrati, tako da najprej blanšira špinačo, speče jajca, opere fižolove kalčke in naseklja korenje. Nato lahko po blanširanju špinače, ki traja najdlje od teh opravil, še prepraži korenje in seseklja špinačo, medtem ko se kuhajo fižolovi kalčki. Vse sestavine za bibimbap je nemogoče pripraviti v manj kot 20 minutah, saj se lonec uporablja 10 minut za blanširanje špinače in 10 minut za kuhanje fižolovih kalčkov, skupaj torej 20 minut.

Računalniško ozadje

Paralelna obdelava je način izvajanja več nalog hkrati, da se skrajša skupni čas in učinkovito izkoristijo viri. To zahteva sinhronizacijo, da ne pride do motenj med nalogami in da se skupni viri, kot so ponve ali lonci, upravljajo brez medsebojnih konfliktov. Razporejanje, ki določa, katera naloga se bo začela kdaj, je prav tako ključno za učinkovitost paralelne obdelave. Če sinhronizacija ali razporejanje nista izvedena pravilno, so lahko naloge zakasnjene ali pa se viri porabljajo neučinkovito, zaradi česar traja dlje. Zato je pomembno natančno načrtovati potek nalog in vrstni red uporabe virov, da bi bila paralelizacija učinkovita. Paralelna obdelava se v praksi uporablja pri delovanju večjedrnih procesorjev, grafičnih kartic, obdelavi velikih podatkov ter treniranju modelov umetne inteligence.

Mačke

6. do 9. razred

Luka mora odpeljati pet mačk k veterinarju na cepljenje. Ima štiri prenosne transportne kletke za živali. Vsaka kletka ima omejeno nosilnost v kilogramih.

Kletka	Nosilnost
A	10 kg
B	15 kg
C	20 kg
D	5 kg

Mačka	Teža
Malček	3 kg
Črniček	7 kg
Rjavko	10 kg
Barvica	5 kg
Lenček	6 kg

Luka mora odpeljati vse mačke in za prevoz uporabiti čim manj transportnih kletk, ne da bi pri tem prekoračil njihovo nosilnost.

Katera od naslednjih možnosti bo omogočila prevoz vseh mačk z uporabo čim manjšega števila transportnih kletk?

A) Malček in Črniček v kletki B; Rjavko, Barvica in Lenček v kletki C.
B) Malček in Črniček v kletki A, Rjavko v kletki B, Barvica v kletki D, Lenček v kletki C.
C) Malček in Barvica v kletki A, Črniček in Lenček v kletki B, Rjavko v kletki C.
D) Malček, Črniček in Rjavko v kletki C, Barvica v kletki D, Lenček v kletki A.
E) Rjavko in Barvica v kletki B, Malček, Črniček in Lenček v kletki C.

Rešitev

Pravilen odgovor je E.

Rešitev:

Rjavko (10 kg) + Barvica (5 kg) = 15 kg se prilega v kletko B (15 kg),

Malček (3 kg) + Črniček (7 kg) + Lenček (6 kg) = 16 kg se prilega v kletko C (20 kg).

Za hitro iskanje pravilnega odgovora lahko preštejemo število uporabljenih kletk pri vsaki možnosti. Možnosti A in E uporabljata 2 kletki, možnosti C in D uporabljata 3 kletke, možnost B pa 4 kletke. Zato začnemo z možnostma A in E. Izkaže se, da možnost A ni izvedljiva, saj bi tri mačke skupaj tehtale 21 kg, kar presega nosilnost kletke C (20 kg). Tako ostane le možnost E, ki hkrati ustreza omejitvam teže.

Računalniško ozadje

Ta naloga modelira razporejanje virov znotraj omejitev. Spominja na različico problema nahrbtnika, kombinatorične optimizacije, kjer je dano število predmetov in njihova teža ter skladišče z omejitvijo teže; rešitev maksimizira število predmetov, ki jih je mogoče shraniti, ne da bi presegli omejitev teže. V tem primeru so mačke predmeti, skladišče pa predstavljajo transportne kletke za živali, kjer ima vsaka kletka omejitev nosilnosti. Nalogo smo obrnili in

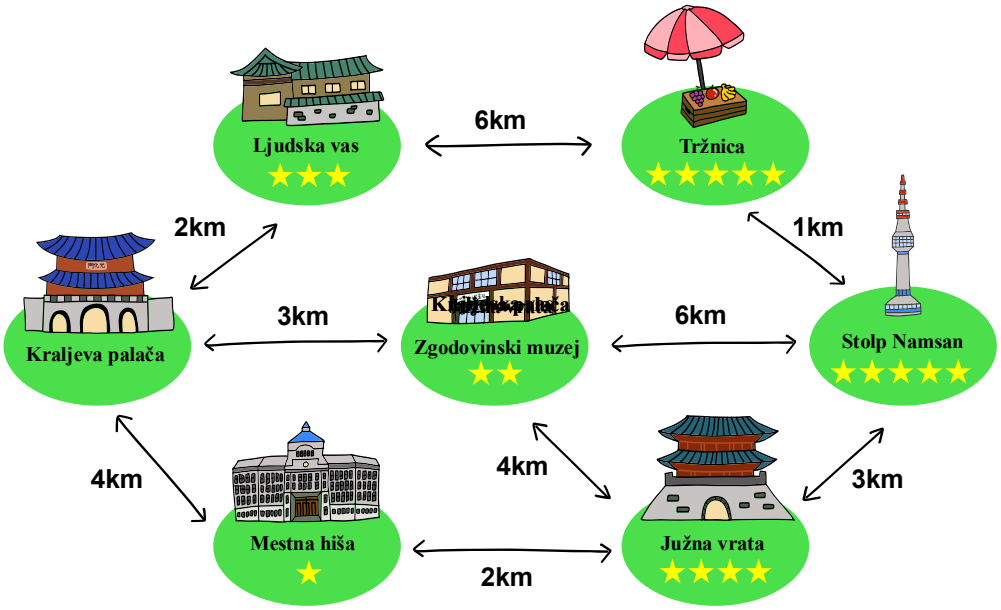
njen cilj je zdaj minimizirati število uporabljenih kletk, to je velikost skladišča, v katero moramo shraniti vse predmete.

Dopustne kombinacije je treba oceniti, da se optimizira shranjevanje, kar je ključna ideja pri delovanju operacijskih sistemov, upravljanju pomnilnika in datotečnih sistemih.

Potovanje po Seulu

6. do 9. razred

Seul nudi turistične avtobuse za obiskovalce. Turistični avtobusi povezujejo glavne turistične atrakcije v Seulu. Na spodnji sliki so prikazane glavne atrakcije, razdalje med njimi, zvezdice ob atrakcijah pa kažejo njihovo priljubljenost.

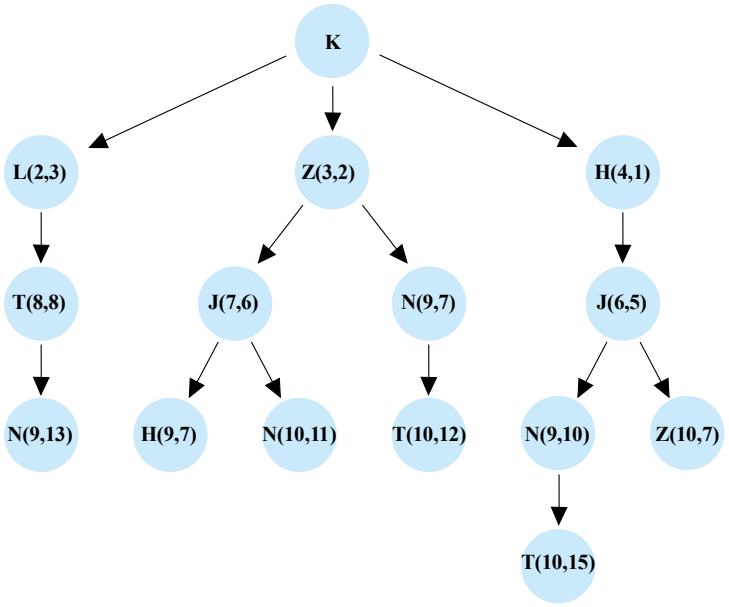


Po obisku Kraljeve palače želi Suzana uporabiti turistični avtobus, vendar ima prevozno kartico, ki ji omogoča potovanje do **največ** 10 kilometrov. Največ koliko zvezdic lahko zbere Suzana? Vsako atrakcijo je mogoče obiskati samo enkrat.

Rešitev

Pravilen odgovor je 15 zvezdic.

Eden od načinov za reševanje problema je uporaba diagrama, sestavljenega iz vozlišč, kjer vsako vozlišče predstavlja eno od atrakcij. Na vrhu diagrama je začetno vozlišče, označeno z začetnico: K za Kraljevo palačo. Od vozlišča K narišemo vse možne poti, ki vodijo od Kraljeve palače proti drugim atrakcijam. Zanimajo nas le poti, katerih skupna dolžina ne presega 10 km. Zato vsako vozlišče v diagramu poleg atrakcije vsebuje še skupno razdaljo od začetka (od vozlišča K) in skupno število zbranih zvezdic.



Na primer, ena možna pot je obisk Ljudske vasi, ki je 2 km od začetka in ima oceno 3 zvezdice. To vozlišče je označeno kot L(2,3). Od tam lahko greste do Tržnice (T): skupna prepotovana razdalja znaša 8 km (2 km + 6 km), skupno število zvezdic pa se poveča za 5, to je na 8. Vozlišče je tako označeno kot T(8,8). Naslednja izbira je lahko le Stolp Namsan (N), ki skupno razdaljo poveča na 9 km in skupno število zvezdic na 13 (8 + 5), zato je vozlišče označeno kot N(9,13).

Za iskanje najboljše poti pregledamo vozlišča na koncih vseh dopustnih poti in izberemo tisto z najvišjo vrednostjo zvezdic, pri tem pa ostajamo znotraj omejitve 10 km. V diagramu je vozlišče z najvišjo vrednostjo zvezdic T(10,15).

To ustreza poti: Kraljeva palača → Mestna hiša → Južna vrata → Stolp Namsan → Tržnica ali krajše: K → H(4,1) → J(6,5) → N(9,10) → N (10,15).

Računalniško ozadje

V tej nalogi se srečamo z optimizacijskim problemom, ki vključuje sprejemanje odločitev z namenom maksimiranja učinkovitosti (na primer števila točk ali dobička) znotraj omejenih virov (na primer časa ali denarja). Pri reševanju takih nalog pogosto iščemo dopustne rešitve, ki zadovoljijo vse omejitve, pri čemer je cilj izbrati tisto, ki je najboljša med njimi. Takšne probleme v praksi srečamo na področjih, kot so načrtovanje turističnih poti, logistika, transportni sistemi, upravljanje projektov in druge dejavnosti, kjer je potrebno učinkovito razporejanje virov.

Urejanje majic

6. do 9. razred

Bober Borut dela v tovarni, ki izdeluje majice. Vsaka majica ima dve lastnosti:

Barva: modra, rumena, rožnata

Vzorec: enobarvna, pikčasta, črtasta

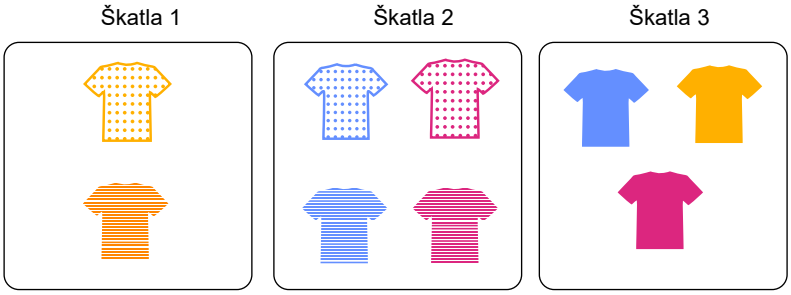
Borut ima stroj, ki majice razvršča v tri različne škatle glede na prejeta navodila.

V ponedeljek je dal naslednje navodilo, kako ravnati z vsako majico:

> Če je majica enobarvna ali črtasta, jo daj v Škatlo 2;
> sicer če je majica rožnata, jo daj v Škatlo 3;
> sicer jo daj v Škatlo 1.

Stroj je majice razvrstil v škatle, kot je prikazano:

V torek je Borut želel, da bi bile majice razvrščene takole:



Katero od navodil naj uporabi?

A) Če je majica rumena, jo daj v Škatlo 1;
sicer če je majica enobarvna, jo daj v Škatlo 3;
sicer jo daj v Škatlo 2.

B) Če je majica enobarvna, jo daj v Škatlo 3;
sicer če je majica rumena, jo daj v Škatlo 1;
sicer jo daj v Škatlo 2.

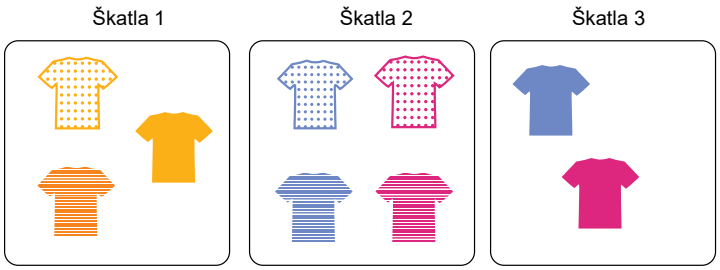
C) Če je majica modra ali rožnata, jo daj v Škatlo 2;
sicer če je majica enobarvna, jo daj v Škatlo 3;
sicer jo daj v Škatlo 1.

D) Če je majica pikčasta ali črtasta, jo daj v Škatlo 2;
sicer če je majica rumena, jo daj v Škatlo 1;
sicer jo daj v Škatlo 3.

Rešitev

Pravilen odgovor je B.

Možnost A je napačna, ker po navodilih najprej vse rumene majice postavi v Škatlo 1 ne glede na vzorec. To pomeni, da bi enobarvna rumena majica šla v Škatlo 1, čeprav želi Borut imeti vse enobarvne majice v Škatli 3. Prvi stavek *če* torej postavi vse rumene majice v Škatlo 1. Stavek *sicer če* od preostalih (modrih in rožnatih) majic izloči enobarvne majice in jih postavi v Škatlo 3. Zadnji stavek *sicer* pa vse preostale majice (torej črtaste in pikčaste majice v modri in rožnati barvi) postavi v Škatlo 2. Spodnja slika prikazuje, kako so majice razvrščene po navodilih iz možnosti A.



Možnost B pravilno razvrsti majice glede na želeni izhod. Najprej so vse enobarvne majice vseh treh barv (modra, rumena in rožnata) postavljene v Škatlo 3. Nato so preostale rumene majice (pikčaste in črtaste) postavljene v Škatlo 1. Nazadnje so vse druge preostale majice (pikčaste in črtaste v modri in rožnati barvi) postavljene v Škatlo 2. To natančno ustreza temu, kar je Borut želel za razvrščanje v torek. Spodnja slika prikazuje, kako so majice razvrščene po navodilih iz možnosti B.

Možnost C je napačna, ker najprej vse modre in rožnate majice postavi v Škatlo 2 ne glede na njihov vzorec. To pomeni, da bi modre in rožnate majice z enobarvnim vzorcem tudi šle v

Škatlo 2, čeprav bi glede na Borutove želje morale biti vse enobarvne majice v Škatli 3. Spodnja slika prikazuje, kako so majice razvrščene po navodilih iz možnosti C.

Možnost D je napačna, ker po tem, ko so vse pikčaste in črtaste majice postavljene v Škatlo 2, preostale rumene majice (to so enobarvne rumene) postavi v Škatlo 1, preostale modre in rožnate pa v škatlo 2. Vendar pa bi glede na želen izhod morale biti vse enobarvne majice v Škatli 3. Spodnja slika prikazuje, kako so majice razvrščene po navodilih iz možnosti D.

Računalniško ozadje

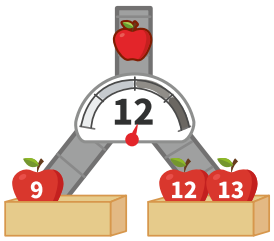
Ta naloga ponazarja nekatere ključne računalniške koncepte:

- Logični operatorji: Stroj razvršča majice po določenih pravilih, kar je podobno uporabi stavkov *če-sicer* (angl. *if-else*) in logičnih operatorjev (AND, OR) v programih.
- Razvrščanje po pravilih: Razvrščanje majic v nalogi deluje podobno, kot računalniki razvrščajo podatke po določenih kriterijih (npr. filtri e-pošte ali sistemi priporočil).
- Sledenje izvajanja: Zaporedje preverjanja pogojev vpliva na rezultat, tako kot pri programih, kjer vrstni red logičnih operacij določa obnašanje programa.

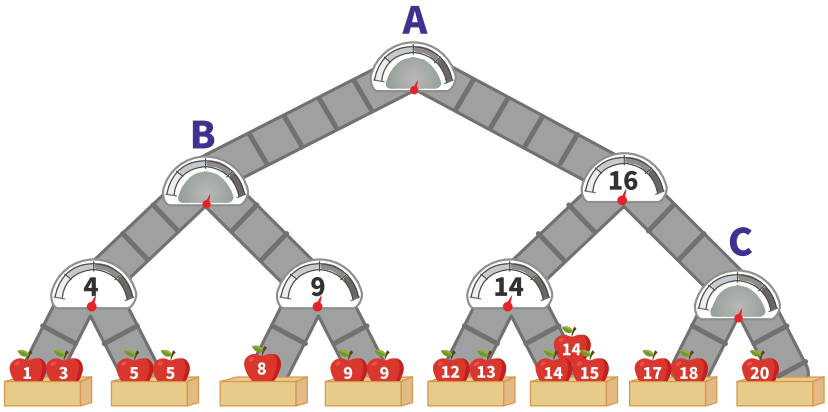
Razvrščanje jabolk

6. do 9. razred, srednja šola

V bobrovem sadovnjaku se jabolka razvrščajo v osem kategorij glede na težo, pri čemer so jabolka iste kategorije poslana na isto pakirno območje. Da bi obiranje potekalo učinkovito, gospod Bober sestavi stroj za samodejno razvrščanje. Jabolka postavi na vrh stroja. Ko jabolko potuje skozi stroj, prehaja skozi tehtalne senzorje, ki določijo njegovo pot: če je jabolko enako ali težje od na senzorju nastavljene vrednosti, pade skozi desni žleb; sicer pade skozi levi žleb. Številka na vsakem jabolku predstavlja njegovo težo.



Stroj je že razvrstil nekaj jabolk, rezultati pa so prikazani na spodnji sliki. Zasloni na tehtalnih senzorjih A, B in C so pokvarjeni, zato moramo ugotoviti njihove vrednosti.



Katere vrednosti so lahko na tehtalnih senzorjih A, B in C, da bo stroj deloval pravilno?

A) 

B) 

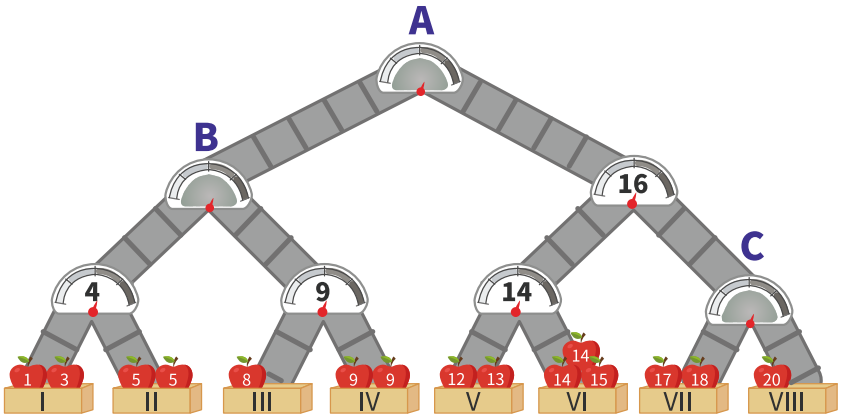
C) 

D) 

Rešitev

Pravilen odgovor je C.

Slika spodaj prikazuje stroj za razvrščanje in jabolka, razvrščena v 8 razredov (od I do VIII).



Pravilo, ki mu sledimo, je naslednje: »Jabolka, katerih teža je enaka ali večja od nastavljene vrednosti, gredo na desno, jabolka z manjšo težo od nastavljene vrednosti pa na levo.« Na podlagi razvrščenih jabolk na sliki lahko zato sklepamo, kakšne so možne nastavitve senzorjev A, B in C.

Senzor A: Jabolka, težka od 1 do 9, so na njegovi levi (razredi I–IV), tista od 12 do 20 pa na njegovi desni (razredi V–VIII). Zato mora biti njegova nastavljena vrednost 10, 11 ali 12.

Senzor B: Jabolka, težka od 1 do 5, so na njegovi levi (razreda I in II), tista težka od 8 do 9 pa na njegovi desni (razreda III in IV). Zato mora biti njegova nastavljena vrednost 6, 7 ali 8.

Senzor C: Jabolka, težka 17 do 18, so na njegovi levi (razred VII), tista s težo 20 pa na njegovi desni (razred VIII). Zato mora biti njegova nastavljena vrednost 19 ali 20.

Na podlagi zgornjih opažanj lahko sklepamo:

Možnost A je napačna, ker nastavljena vrednost senzorja B ne more biti 5. Če bi bil B nastavljen na 5, bi jabolko s težo 5 šlo na desno v razred III. Rezultat pa kaže, da jabolko s težo 5 konča v razredu II (na levi).

Možnost B je napačna, ker nastavljena vrednost senzorja C ne more biti 17. Če bi bil C nastavljen na 17, bi jabolka s težo 17 in 18 šla na desno v razred VIII. Namesto tega pa so uvrščena v razred VII (na levi).

Možnost C je pravilna, saj so vse vrednosti senzorjev skladne z opaženimi rezultati sortiranja.

Možnost D je napačna, ker nastavljena vrednost senzorja A ne more biti 13. Če bi bil A nastavljen na 13, bi jabolko s težo 12 šlo na levo. Vendar je uvrščeno v razred V (na desni).

Računalniško ozadje

V tej nalogi stroj za razvrščanje deluje podobno kot dvojiško iskalno drevo, podatkovna struktura v računalništvu, ki omogoča učinkovito urejanje in iskanje podatkov. Vsako vozlišče ima največ dve poddrevesi – levo in desno – pri čemer so vrednosti v levem poddrevesu manjše ali enake, v desnem pa večje od trenutnega vozlišča. To omogoča, da iskanje poteka po principu razpolavljanja, kar bistveno pospeši dostop do podatkov. Dvojiška iskalna drevesa se uporabljajo v bazah podatkov, računalniških igrah in drugih programih, kjer je potreben hiter dostop do urejenih informacij, ter predstavljajo osnovo za naprednejše, uravnotežene strukture, kot so AVL ali rdeče-črna drevesa.

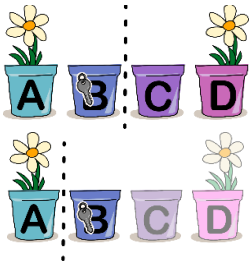
Cvetlični lončki 1

8. do 9. razred

Barbara ima pred vhodnimi vrati vrsto cvetličnih lončkov. Nekateri lončki so prazni, v drugih pa je posajena ena roža. Ključ od vhodnih vrat je skrila v enega izmed lončkov. Prijateljici pove, kako lahko poišče ključ:

> »Najprej si oglej vse lončke. Če je skupno število rož sodo, je ključ skrit v levi polovici lončkov, sicer je skrit v desni polovici. Nato poglej tisto polovico lončkov, v kateri je ključ. Postopek ponavljaj, dokler ti ne ostane en sam lonček – v njem je skrit ključ.«

Podala je tudi primer. Če so na pragu 4 lončki in je ključ skrit v lončku B, lahko prijateljica najde ključ takole:



Pogleda vse štiri lončke A, B, C in D. V njih sta posajeni 2 roži, to je ***sodo*** število. Torej mora biti ključ skrit v lončkih na ***levi*** polovici, v A ali B.

Pogleda lončka A in B, posajena je ena sama roža. Ker je 1 ***liho*** število, mora biti ključ skrit na ***desni*** polovici, v lončku B.

Naslednji dan je Barbara pred vrata postavila 8 cvetličnih lončkov in ključ skrila v lonček C. Najmanj koliko rož mora posaditi, da lahko prijateljica najde ključ po opisanem postopku?

Rešitev

Posaditi mora najmanj 2 roži.

Rešitev, kam posaditi rože, je več, saj sta roži lahko posajeni v različnih lončkih: ena mora biti posajena v lončku A ali B, druga pa v enem od lončkov E, F, G ali H. Eno od možnih rešitev prikazuje naslednja slika:

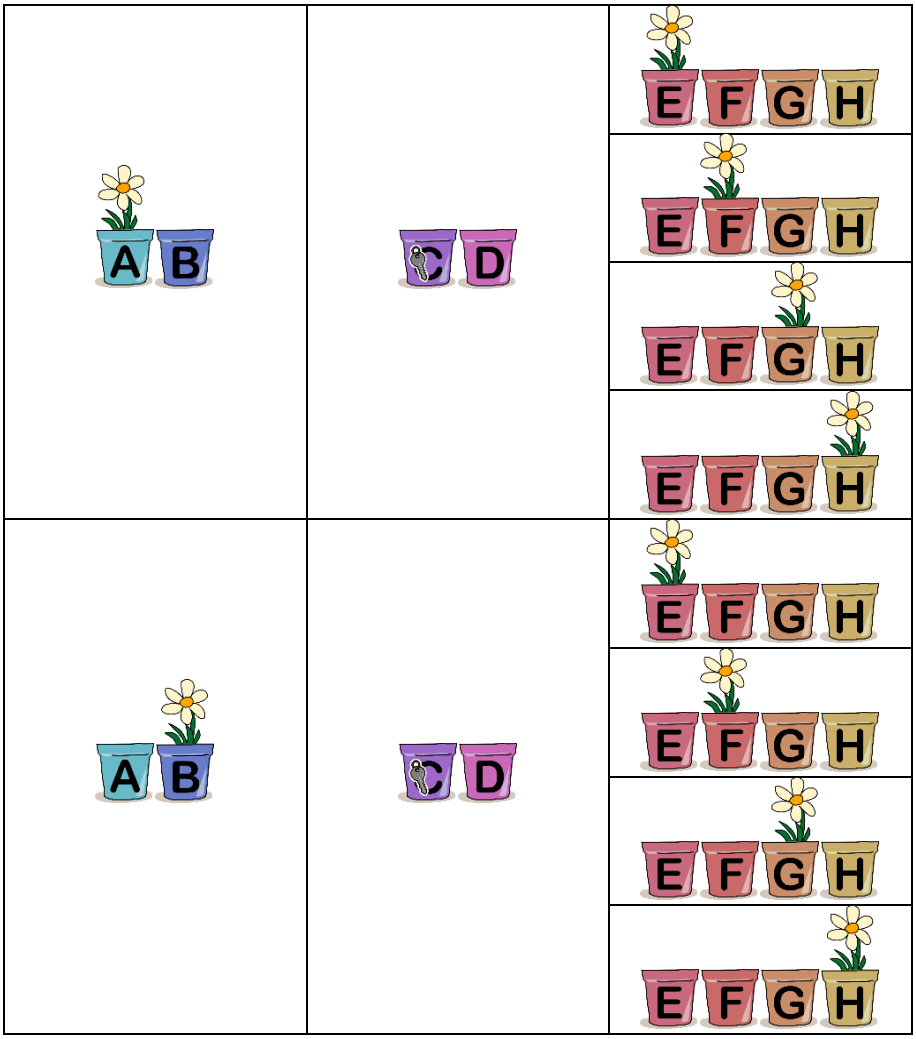
Poglejmo, kako pridemo do rešitve naloge. Pri iskanju ključa moramo najprej prešteti vse rože v lončkih. Ker je ključ v lončku C, moramo v naslednjem koraku pregledati lončke od A do D, torej levo polovico. Zato mora biti v vseh lončkih skupaj posajeno sodo število rož.

V lončkih A, B, C in D mora biti število rož liho, da po navodilih iščemo ključ v desni polovici, torej v lončkih C in D. Na koncu pa mora biti sodo število rož tudi v teh dveh lončkih (C in D), da za ključ pogledamo v levo polovico lončkov, to je lonček C. Torej morajo biti izpolnjeni ti trije pogoji, da lahko najdemo ključ.

In kakšno je najmanjše število posajenih rož? V lončkih C in D mora biti sodo število rož, najmanjše tako število pa je 0. Torej sta lončka C in D prazna, brez rož. V lončkih A, B, C in D mora biti liho število rož. Najmanjše liho število je 1, torej mora biti 1 roža posajena ali v

lončku A ali v lončku B (za lončka C in D smo že ugotovili, da morata biti brez rož). Ker pa mora biti skupno število rož v vseh lončkih sodo, na levi polovici (lončki A, B, C in D) pa imamo posajeno eno rožo, moramo eno rožo posaditi tudi na desni polovici, in sicer v katerem koli lončku E, F, G ali H. Tako moramo skupaj posaditi najmanj dve roži.

Spodnja tabela prikazuje vse možne rešitve, teh je 8.



Računalniško ozadje

Barbara je za označevanje položaja ključa uporabila rože. Ali povedano bolj natančno, položaj ključa je zakodirala kot zaporedje lončkov brez rože ali z rožo. Njena prijateljica je lahko dekodirala položaj ključa iz tega zaporedja. Ta koda je uporabna le zato, ker vsako zaporedje določa natanko en položaj – tako prijateljica točno ve, kje iskati ključ.

V računalništvu so kode bistvene za predstavitev podatkov v obliki, ki jo lahko obdeluje računalniška strojna oprema, na primer za pošiljanje sporočil prek interneta. Mnoge kode uporabljajo daljša zaporedja, kot bi bila nujno potrebna, saj lahko odvečni deli zaporedja pomagajo zaznati ali celo popraviti napake v primeru, ko so deli zaporedja po nesreči spremenjeni. Take so na primer Hammingove kode.

Vedno desno

8. do 9. razred

Robot Robo je pokvarjen! Zato mora v garažo na nujno popravilo, vendar se zaradi okvare premika na nenavaden način:

1. Vedno zavije desno, če je desni zavoj mogoč.
2. Če ne more zaviti desno, gre naravnost.
3. Nikoli ne zavije levo.

Problem? Brez pomoči bi lahko Robo šel po napačni poti in se zataknil ali celo izgubil! Da bo varno prispel do garaže po sivi cesti, mu moramo preprečiti napačne zavoje, zato bomo na nekaterih mestih na cesti postavili stene.

Ena stena () je že postavljena. Najmanj koliko sten moramo še dodati, da zagotovimo, da Robo pride do garaže, ne da bi se izgubil?

Rešitev

Dodati moramo še najmanj 5 sten, tako da bo skupaj z že postavljeno steno ob cesti 6 sten.

Robo bo vedno zavijal desno, če je mogoče, zato bi brez sten lahko šel v napačno smer. Če pogledamo steno 1 na spodnji sliki, vidimo, da preprečuje Robu, da bi zavijal desno v slepo ulico. Ne pozabite, da Robo sploh ne more zaviti levo! S postavitvijo sten na ta način mu lahko

preprečimo, da se zatakne. Stene 2, 3, 4 in 6 na spodnji sliki so prav tako potrebne, ker bi sicer Robo zavijal desno in prišel do mesta, kjer je na voljo le levi zavoj, in bi se tam zataknil. Stena 5 na spodnji sliki je prav tako potrebna, saj preprečuje, da bi Robo zašel v zanko, kjer bi vedno zavijal desno.

Računalniško ozadje

Ta naloga je povezana z izogibanjem oviram v robotiki. Avtonomni roboti in samovozeča vozila uporabljajo algoritme za odločanje, da se premikajo po zapletenih okoljih. Senzorji zaznavajo ovire, programska oprema pa prilagodi pot, da se izogne trkom in učinkovito doseže cilj.

V resničnem svetu algoritme za izogibanje ovir uporabljajo droni, GPS-navigacija, igralna umetna inteligenca in samovozeča vozila. Tudi roboti v skladiščih (npr. v Amazonovih distribucijskih centrih) sledijo natančnim potem, uporabljajo podobno logiko in se izogibajo oviram, da učinkovito dosežejo ciljne lokacije.

Cvetlični vrt

8. do 9. razred, srednja šola

Ema bo na vrtu posadila rože. Na gredici ima prostor za 4 različne vrste rož.

V rastlinjaku si ogleduje sadike 9 različnih vrst rož. Vsaka vrsta rož potrebuje določeno število dni od posaditve, da se razcveti. In nato cveti določeno število dni.

Vrsta rože	Število dni od posaditve do cvetenja	Število dni cvetenja
marjetica	8	4
sivka	3	4
lilija	10	2
ognjič	4	3
vijolica	13	3
potonika	13	4
vrtnica	3	6
sončnica	12	5
tulipan	2	2

Ema želi izbrati take rože, da bo imela na gredici cvetoče rože vsaj 15 dni zapored (tj. da bo v tem času cvetela vsaj ena vrsta rož). Zato mora rože zelo skrbno izbrati. Katere štiri vrste rož naj izbere?

Rešitev

Pravilen odgovor je tulipan, vrtnica, marjetica, sončnica.

Do rešitve lahko pridemo na dva načina, z algoritmičnim pristopom ali z vizualizacijo podatkov.

Algoritmični pristop

Rože cvetijo zelo malo časa. Zadnje bodo lahko cvetele potonike in sončnice še 16. dan od posaditve. Če želimo imeti cvetočo gredico vsaj 15 dni, moramo izbrati vsaj eno vrsto rož, ki začne čimprej cveteti. Najbolj zgodaj zacvetijo tulipani, ki cvetijo že 2. dan po posaditvi in cvetijo dva dni. Zato moramo kot naslednje izbrati rože, ki zacvetijo najkasneje na 4. dan. Izberemo lahko sivko ali vrtnico (obe zacvetita 3. dan) ali ognjič (zacveti 4. dan). Od navedenih najdlje cvetijo vrtnice (do 8. dne), zato so najboljša izbira. Naslednje rože morajo zacveteti najkasneje na 9. dan. Tu nimamo izbire – marjetice so edine, ki cvetijo 9. dan (razcvetijo se že 8. dan in cvetijo 4 dni). Tako imamo cvetoče rože do 11. dne. Potrebujemo še rože, ki zacvetijo najkasneje na 12. dan. To so sončnice, ki tudi edine cvetijo 12. dan po posaditvi. Cvetijo 5 dni, torej do 16. dne od posaditve. Tako smo s tulipani, vrtnicami, marjeticami in sončnicami zagotovili cvetoče rože na gredici od 2. do 16. dne, torej skupaj 15 dni, kar ustreza Emini želji.

Vizualizacija podatkov

Vse podatke o rožah, začetku in koncu cvetenja prikažemo grafično, najlažje kar z diagramom v obliki tabele. Cvetenje posamezne vrste rož je označeno s **c** v tabeli.

Vrsta rož	Število dni od dneva posaditve																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
marjetica									c	c	c	c					
sivka				c	c	c	c										
lilija											c	c					
ognjič					c	c	c										
vijolica														c	c	c	
potonika														c	c	c	c
vrtnica				c	c	c	c	c	c								
sončnica													c	c	c	c	c
tulipan			c	c													

Z grafičnim prikazom hitro vidimo, da tulipani začnejo cveteti prvi in da jih moramo posaditi, če želimo doseči 15 dni cvetočega vrta. Na 7. dan cvetijo le vrtnice, zato moramo izbrati tudi te. Podobno velja za 9. dan, ko edine cvetijo marjetice, in 12. dan, ko cvetijo le sončnice. Zato moramo izbrati te štiri vrste rož, da bo od dneva 2 do dneva 16 cvetela vsaj ena vrsta rož. V tabeli so obarvane oranžno.

Računalniško ozadje

V nalogi opisan problem bi z računalnikom lahko rešili z uporabo algoritma *pokrivanja intervalov* (angl. *interval covering*), ki je posebna vrsta *požrešnega algoritma* (angl. *greedy algorithm*). Požrešni algoritem išče rešitev tako, da vedno izbira trenutno najboljšo možnost, ne da bi upošteval njen vpliv na končno rešitev. Rešitev gradi postopoma, po korakih, pri čemer na vsakem koraku izbere najbolj obetavno možnost, v upanju, da bodo te izbire vodile do končne rešitve.

Karta svetlosti

8. do 9. razred, srednja šola

Karto svetlosti za sliko pripravimo tako, da sliki najprej dodamo okvir belih pikslov (kvadratkov), nato pa vsak piksel slike predstavimo s številom:

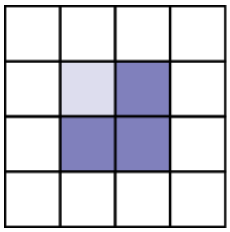
Če je piksel svetlejši od svojega desnega soseda, dobi število **1**.

Če ima piksel enako svetlost kot njegov desni sosed, dobi število **0**.

Če pa je piksel temnejši od svojega desnega soseda, dobi število **-1**.

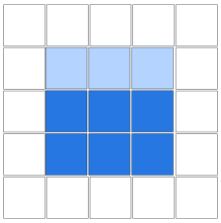
Poglejmo primer:

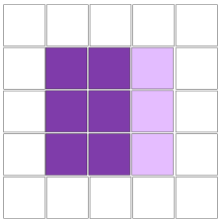
Sliko velikosti 2 x 2 piksla obkrožimo z belimi piksli (spodaj levo) in za vsak piksel slike določimo število svetlosti glede na zgoraj navedena pravila (spodaj desno).

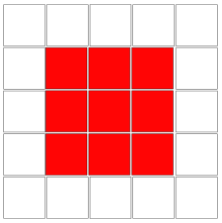


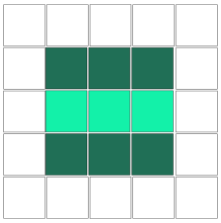
1	-1
0	-1

Seveda imajo lahko slike enako karto svetlosti, čeprav izgledajo različno. Katera od spodnjih slik ima karto svetlosti drugačno od ostalih treh?

A) 

B) 

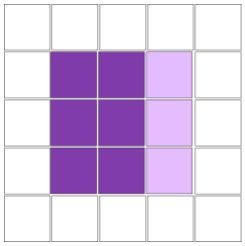
C) 

D) 

Rešitev

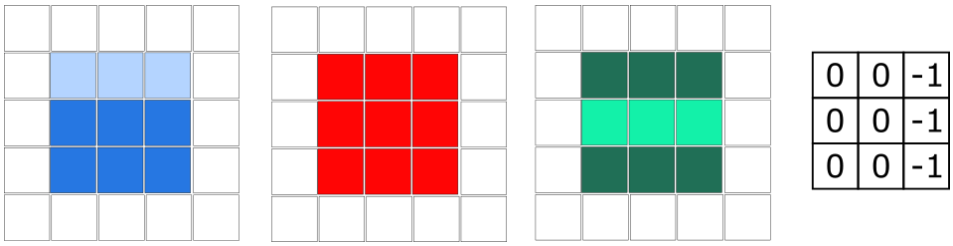
Pravilen odgovor je B.

Do rešitve lahko pridemo tako, da izračunamo karte svetlosti za vse štiri slike. Najprej izračunajmo karto svetlosti za sliko pri odgovoru B (to je pri pravilnem odgovoru):



0	-1	-1
0	-1	-1
0	-1	-1

Ostale tri slike (pri odgovorih A, C in D) pa imajo naslednjo karto svetlosti:



Lahko pa odgovor poiščemo tudi hitreje. Karto svetlosti namreč izračunamo tako, da gledamo vrednosti dveh zaporednih pikslov po vrsticah (za vsako vrstico posebej). Torej nam karta svetlosti pokaže spremembe v svetlosti na sliki v vodoravni smeri (ne pa v navpični). Tri slike pri podanih odgovorih imajo v vrsticah piksle enake svetlosti, zato bodo imele tudi enake karte svetlosti. Le ena slika – tista pri odgovoru B – ima v vrsticah spremembo v svetlosti pikslov. Torej bo imela drugačno karto svetlosti od ostalih treh.

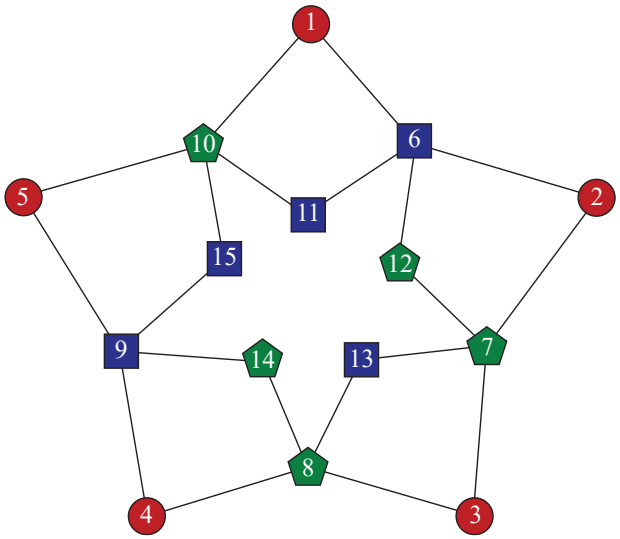
Računalniško ozadje

Pri računalniškem vidu sliko »predelamo« z različnimi filtri tako, da iz nje izluščimo določene značilke, ki prikazujejo posebne strukturne lastnosti slike (kot so na primer robovi, oglišča, področja enake barve itd.). Tako računalnik lažje interpretira vizualne informacije in detektira posamezne objekte na sliki.

Luči

8. do 9. razred, srednja šola

Sofija ima petnajst programabilnih luči treh vrst: ●, ■ in ⬟. Oštevilčila jih je od 1 do 15 in jih povezala v zvezdasto obliko.



Programirano delovanje luči je naslednje:

- Vsaka ● luč je nadzorovana s stikalom.
- Vsaka ■ luč se prižge, če sta obe najnižje oštevilčeni luči, ki sta povezani z njo, prižgani.
- Vsaka ⬟ luč se prižge, če je prižgana *natanko ena* od dveh najnižje oštevilčenih luči, ki sta povezani z njo. Z drugimi besedami, ena od teh luči je prižgana, druga pa ne.

Vse luči so ugasnjene, nato pa Sofija prižge luči 1, 2 in 4, v tem vrstnem redu. To sproži prižiganje drugih luči, v skladu s Sofijinim programiranjem. Katere luči od 11 do 15 so prižgane, ko se prižiganje luči ustali?

A) Luči 11, 12, 13, 14 in 15.
B) Samo luči 12, 13 in 15.
C) Samo luči 11, 12, 13 in 14.
D) Samo luči 11, 13 in 14.

Rešitev

Pravilen odgovor je D.

Od ● luči so prižgane le 1, 2 in 4. Nato se premikamo proti središču zvezde, da določimo, katere preostale luči se prižgejo.

Ker je luč 6 ■ luč in sta luči 1 in 2 prižgani, se luč 6 prižge.

Ker je luč 7 ⬟ luč in je luč 2 prižgana, luč 3 pa ugasnjena, se luč 7 prižge.

Ker je luč 8  luč in je luč 4 prižgana, luč 3 pa ugasnjena, se luč 8 prižge.

Ker je luč 9  luč, a luči 4 in 5 nista obe prižgani, luč 9 ostane ugasnjena.

Ker je luč 10  luč in je luč 1 prižgana, luč 5 pa ugasnjena, se luč 10 prižge.

Ker je luč 11  luč in sta luči 6 in 10 obe prižgani, se luč 11 prižge.

Ker je luč 12  luč, a sta luči 6 in 7 obe prižgani, luč 12 ostane ugasnjena.

Ker je luč 13  luč in sta luči 7 in 8 prižgani, se luč 13 prižge.

Ker je luč 14  luč in je luč 8 prižgana, luč 9 pa ugasnjena, se luč 14 prižge.

Ker je luč 15  luč, a luči 9 in 10 nista obe prižgani, luč 15 ostane ugasnjena.

Tako se od petih luči, najbližjih središču zvezde, prižgejo le luči 11, 13 in 14.

Računalniško ozadje

V tej nalogi je lahko vsaka luč prižgana ali ugasnjena, kar računalniški strokovnjaki označujejo z binarnima številkama 1 in 0 oziroma z logičnima (Boolovima) vrednostma resnično ("true") in neresnično ("false").

Računalniški vezja, vključno s tistimi v strojni opremi, kot je centralna procesna enota (CPU), uporabljajo logična vrata za spreminjanje Boolove vrednosti in izvajanje izračunov.  luči v tej nalogi so primeri vrat AND: dokler sta obe vhodni vrednosti resnični, bo tudi izhodna vrednost resnična; sicer bo izhodna vrednost neresnična.  luči v tej nalogi so primeri vrat XOR, kar pomeni " izključujoči ali" (anlg. *exclusive or*): če je natanko ena vhodna vrednost resnična, bo izhod resničen; sicer bo izhod neresničen.

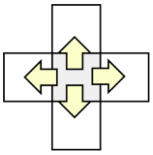
CPU uporablja logična vrata za vse potrebne izračune. Vse, kar počnete na računalniku in zahteva procesiranje, poteka na CPU in s tem gre skozi potencialno milijarde logičnih vrat!

Meglen dan

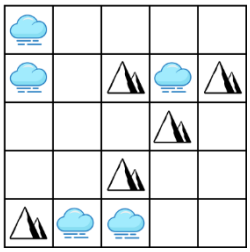
8. do 9. razred, srednja šola

Na zelo meglen dan v deželi bobrov se je megla razširila vsepovsod.

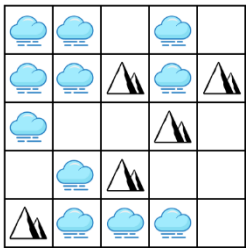
Zemljevid terena dežele lahko prikažemo kot mrežo. Megla najprej nastane v nekaterih kvadratkih mreže in se vsako uro razširi na vse štiri sosednje kvadratke: levo, desno, gor in dol. Tako se megla sčasoma razširi na vse kvadratke, razen na tiste z gorami.



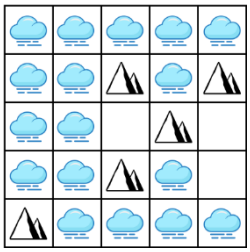
Spodnje slike prikazujejo razširjanje megle:



Začetek

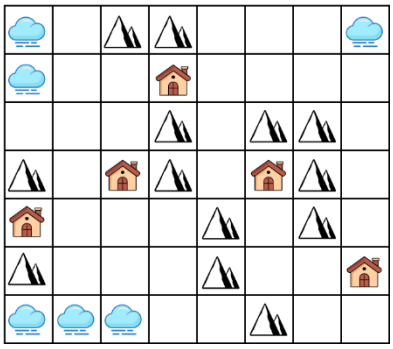


Po 1 uri



Po 2 urah

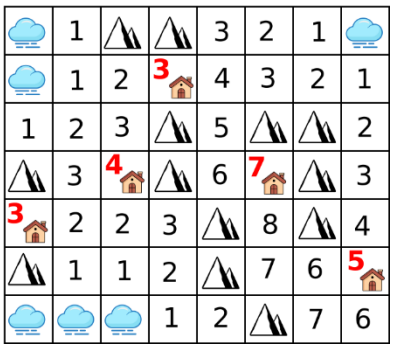
Spodnja slika prikazuje zemljevid s trenutnim stanjem meglenih predelov. Čez koliko ur bo megla prekrila vse hiše?



Rešitev

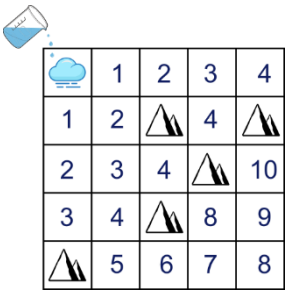
Vse hiše bo megla prekrila čez 7 ur.

Rešitev lahko poiščemo tako, da simuliramo razširjanje megle in tako ugotovimo, kdaj bodo z meglo prekrite vse hiše. Na spodnji sliki številka v kvadratku pove, po koliko urah bo megla prekrila ta kvadratek. Začnemo z vpisovanjem 1 na sosednje kvadratke vseh kvadratkov z meglo. Seveda pri tem izpustimo vse kvadratke z gorami. Nato za vse kvadratke s številko 1 poiščemo vse sosednje kvadratke, ki še niso prekriti z meglo, in vanje vpišemo 2 (do njih se bo megla priplazila po dveh urah). Nato gledamo sosednje kvadratke od številke 2. Postopek ponavljamo, dokler nimamo številk v vseh kvadratkih s hišami. Vidimo, da megla zadnjo hišo prekrije po sedmih urah.



Vendar ni potrebno, da izračunamo in vpišemo vse te številke, da lahko najdemo odgovor na vprašanje. Lahko se namreč osredotočimo le na vsako od hiš in zanje poiščemo »razdaljo« do najbližjega meglenega kvadratka. Največja od teh razdalj je 7 (za hišo, ki je »skrita« za gorami).

Računalniško ozadje



Razširjanje megle je podobno poplavljanju. Če bi v kvadratek z meglenim oblakom zlili vodo, bi se voda razlila v sosednje kvadratke, najprej na 1, nato na 2, nato 3 in tako naprej, dokler ne bi zalila celega območja. Gore so za vodo bariera, preko njih se ne preliva.

V naši nalogi se poplavljanje začne sočasno iz vseh meglenih oblakov.

V računalništvu ta postopek imenujemo *algoritem poplavljanja*. Ti algoritmi se pogosto uporabljajo v računalniških omrežjih (npr. za usmerjanje paketov) in v računalniški grafiki (npr. za spreminjanje barve površine).

Prevajalni sistem

8. do 9. razred, srednja šola

Bobri so postavili prevajalni sistem med jeziki antilop (A), bobrov (B), cekinčkov (C), delfinov (D) in emujev (E). Natančnost prevoda med jeziki je ocenjena z vrednostmi med 0,0 in 1,0 (višja kot je vrednost, bolj natančen je prevod) in je prikazana v spodnji tabeli:

	 A	 B	 C	 D	 E
 A	**1,0**	**0,9**	**0,2**	**0,7**	**0,6**
 B	**0,9**	**1,0**	**0,5**	**0,8**	**0,7**
 C	**0,2**	**0,5**	**1,0**	**0,9**	**0,7**
 D	**0,7**	**0,8**	**0,9**	**1,0**	**0,7**
 E	**0,6**	**0,7**	**0,7**	**0,7**	**1,0**

Včasih posredni prevod da boljši rezultat kot neposredni; natančnost posrednega prevoda je produkt vseh natančnosti, prek katerih poteka prevod.

Na primer, neposredni prevod **C** → **A** ima natančnost **0,2**.

Če pa gremo po poti **C** → **B** → **A**, dobimo **0,5 × 0,9 = 0,45**, kar je bolje kot **0,2**.

V katerem od danih neposrednih prevodov bi bilo bolje, če bi uporabili posredni prevod?

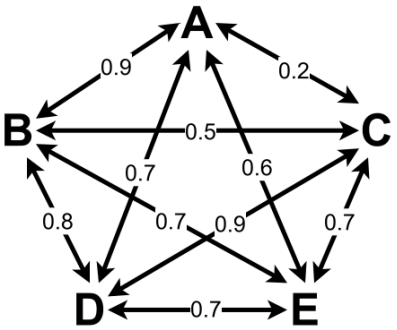
A) A → D
B) B → A
C) B → E
D) D → B

Rešitev

Pravilen odgovor je A.

Posredni prevod je bolje uporabiti v primeru, ko prevajamo iz jezika Antilop v jezik Delfinov, saj ga lahko izboljšamo s posrednim prevodom v jezik Bobrov.
Natančnost prevoda A→ D je 0,7, s posrednim prevodom preko B (A → B → D) pa ga povečamo na: 0,9 × 0,8 = 0,72.

Pri reševanju naloge si lahko pomagamo z grafom, na katerem so označene natančnosti prevodov:



V primeru odgovora B je natančnost prevajanja med jezikoma Bobrov in Antilop 0,9. Ker so natančnosti prevodov med različnimi jeziki vse manjše od 1 in so tako vrednosti produkta manjše od posamezne natančnosti, ne moremo s posrednim prevodom dobiti večjega števila od 0,9.

V primeru odgovora C je natančnost prevoda med jezikoma Bobrov in Emujev enaka 0,7. Da bi izboljšali prevod, bi morala biti natančnost vsaj 0,72 (0,8 × 0,9), vendar noben prevod v jezik emujev nima ocene natančnosti višje od 0,7. Torej ne moremo izboljšati tega prevoda.

V primeru odgovora D je natančnost prevodov med jezikoma Delfinov in Bobrov 0,8. Da bi izboljšali prevod, bi morala biti natančnost vsaj 0,81 (0,9 × 0,9), vendar take možnosti nimamo.

Računalniško ozadje

S podatkovnimi strukturami lahko shranjujemo ne le podatke, temveč tudi odnose med podatki. Eden od načinov prikaza povezav med stvarmi je graf. Sestavljajo ga točke (vozlišča) in črte (povezave/robovi), ki točke povezujejo. Grafe uporabljamo v računalništvu za prikaz razmerij, npr. zemljevide, družbena omrežja ali tok podatkov v omrežju.

V teoriji grafov pogosto obravnavamo algoritme, ki najdejo optimalno pot (z najmanjšim stroškom oziroma največjo koristjo) med izbranima vozliščema. To nalogo lahko razumemo kot problem iskanja najvišje možne natančnosti prevoda z enega jezika v drugega.

Robot v labirintu 2

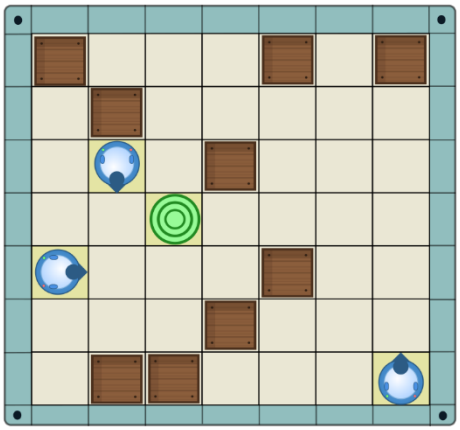
8. do 9. razred, srednja šola

Robot se premika po sobi, polni ovir, in mora priti do cilja. Premikanje robota nadzorujemo s tremi preprostimi ukazi:

- ***Naprej***. Robot se premika naprej v smeri, v katero gleda (robot se bo premikal v desno), dokler ne zadene ob steno ali ob oviro. Takrat se ustavi. Robot se ustavi tudi v primeru, ko pride na cilj.
- ***Obrat levo***. Robot se na mestu obrne za 90° v levo (se ne premakne s polja).
- ***Obrat desno***. Robot se na mestu obrne za 90° v desno.

Robot bo sledil podanim ukazom po vrsti.

V sobi imamo tri robote, ki jih želimo pripeljati do cilja s čim manj ukazi. Roboti so na različnih mestih v sobi in gledajo v različne smeri. Sestavimo zaporedje ukazov (program), ki vse tri robote uspešno pripelje do cilja.



Kolikokrat se vsak robot zaleti v oviro na svoji poti do cilja?

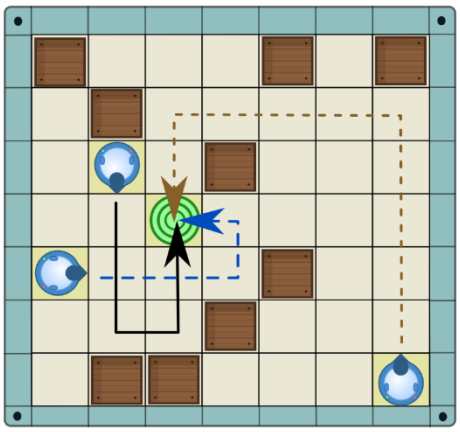
Rešitev

Vsak robot se na svoji poti do cilja dvakrat zaleti v oviro.

Odgovor poiščemo tako, da najprej rešimo problem, tj. sestavimo tako najkrajše zaporedje ukazov, ki vse tri robote pripelje do cilja. Zaporedje ukazov mora biti naslednje:

Naprej. Obrat levo. Naprej. Obrat levo. Naprej.

Na spodnji sliki so prikazane poti vsakega od treh robotov, če sledi navedenemu zaporedju ukazov.



Za to sobo je to tudi edina rešitev, ki vse tri robote pripelje do cilja z najmanjšim številom ukazov. Vidimo, da se vsak od robotov zaleti v oviro natanko dvakrat (ko se ustavi, pred vsakim obratom).

In kako ugotovimo, da je to edina najkrajša rešitev? Opazimo lahko naslednje:

1. Robot gleda v določeno smer in potrebuje ukaz, da se obrne.
2. Cilj je na levi strani vseh treh robotov. Torej mora rešitev vključevati vsaj en obrat v levo.
3. Zaradi robota v kotu desno spodaj se zaporedje ukazov ne more začeti z obratom (saj bi to robota pripeljalo v »slepo ulico« in bi se na poti do cilja moral vrniti nazaj na izhodiščno pozicijo).
4. Po trku ob prvo oviro in obratu v levo je cilj še vedno levo od robota (to velja za vse tri robote). Torej mora rešitev vključevati vsaj še en obrat v levo.

Tako mora najkrajša rešitev vsebovati dva obrata v levo, vmes pa so premiki naprej. Zaporedje ukazov se mora začeti s premikom naprej. Zadnji premik naprej bo robota pripeljal do cilja, kjer se bo ustavil.

Računalniško ozadje

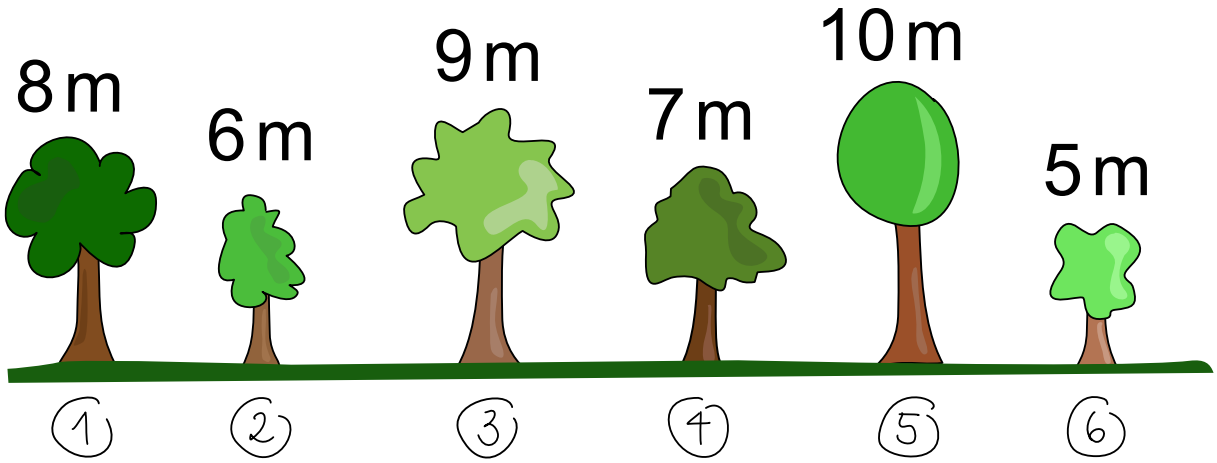
V nalogi smo robota usmerjali s pomočjo treh enostavnih ukazov. Lahko rečemo, da ti ukazi sestavljajo enostaven programski jezik. Čeprav je nabor ukazov zelo omejen, lahko te ukaze poljubnokrat ponovimo v, kar nam omogoča izdelavo tudi bolj kompleksnih programov. Še vedno pa se ti programi izvajajo zaporedno, torej ukaz za ukazom v podanem vrstnem redu. Vsak ukaz se mora najprej zaključiti, preden se začne izvajati naslednji ukaz. Tako zaporedno izvajanje je eden od osnovnih konceptov pri programiranju.

Sekanje dreves

8. do 9. razred, srednja šola

Bobri morajo posekati nekaj dreves za gradnjo jezu. Drevesa so oštevilčena od 1 do 6 in imajo različne višine. Pri izbiri naslednjega drevesa za posek sledijo dvema preprostima praviloma:
Pravilo 1: Naslednje drevo mora biti označeno z višjo številko kot prejšnje izbrano drevo.
Pravilo 2: Naslednje izbrano drevo mora biti nižje od prejšnjega.

Na primer, če se odločijo začeti z drevesom št. 2 (visokim 6 m), morajo nadaljevati z drevesom št. 6 (visokim 5 m). Skupna višina teh dveh dreves je 11 m.



Kakšna je največja skupna višina dreves, ki jih lahko bobri posekajo?

Rešitev

Največja skupna višina dreves, ki jih lahko bobri posekajo, je 21 metrov.

Naloga je najti zaporedje dreves z največjo možno vsoto njihovih višin. Za rešitev sistematično poiščemo vsa možna zaporedja dreves s padajočimi višinami in izračunamo vsoto njihovih višin. Ker iščemo največjo skupno višino, vsako zaporedje poiščemo do konca (na primer, ne bomo obravnavali zaporedja 8, 6, ampak le končno zaporedje 8, 6, 5) ter vzamemo največjo skupno višino med vsemi takimi končnimi zaporedji.

Ideja, kako najti vsa polna zaporedja:

- Izberi drevo št. 1 in njegovo višino dodaj kot prvi element zaporedja.
- Nadaljuj na drevo št. 2.
- Če je drevo št. 2 nižje, njegovo višino dodaj k zaporedju, sicer pojdi na naslednje drevo.
- Nadaljuj do konca vrste in seštej višine v polnem zaporedju.
- Začni znova z drevesom št. 1, tokrat preskoči drevo št. 2 (oz. 3, 4 …) in ponovi isti postopek.
- Ko končaš z vsemi drevesi, imaš vsa možna polna zaporedja, ki se začnejo z drevesom št. 1.
- Celoten postopek ponovi tako, da začneš z drevesom št. 2, nato z drevesom št. 3, in tako naprej, dokler ne upoštevaš vseh dreves.

Tako dobimo vsa možna končna zaporedja, prikazana v tabeli. Rešitev je največja skupna višina iz tabele, to je 21.

Končno zaporedje dreves	Skupna višina
8, 6, 5	19
8, 7, 5	20
8, 5	13
6, 5	11
9, 7, 5	21
9, 5	14
7, 5	12
10, 5	15

Računalniško ozadje

Ta naloga je optimizacijski problem, saj poskušajo bobri najti najboljše možno zaporedje dreves — v skladu z določenimi pravili — da maksimirajo skupno višino posekanih dreves.

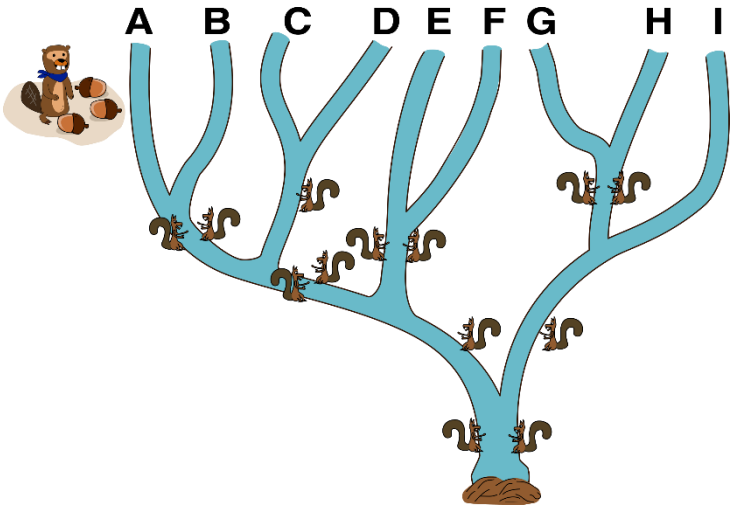
Optimizacijska naloga pomeni sprejeti najboljšo možno odločitev. Predstavljaj si, da imaš veliko možnosti, ti pa želiš tisto, ki je najhitrejša, najkrajša, najcenejša ali najbolj zabavna — to je optimizacija.

Za rešitev smo uporabili metodo, imenovano izčrpno iskanje. To pomeni, da poskusimo vse možne rešitve eno za drugo in nato izberemo najboljšo. Ker je tukaj le šest dreves, lahko to še vedno naredimo ročno. Pri večjih problemih, z več drevesi ali več pravili, je pa računalnik veliko boljši pri hitrem in pravilnem preverjanju mnogih možnosti. Računalničarji pišejo programe, ki pomagajo najti najboljši odgovor, kadar človek nima časa preizkusiti vseh.

Želodi

8. do 9. razred, srednja šola

Bober ima tri želode in stoji pri izvirih devetih potokov (na sliki smo jih označili s črkami A, B, C, D, E, F, G, H in I), ki se vsi združijo pri bobrovem jezu na dnu. Bober želi po potokih spraviti želode do bobrovega jeza.



Vendar ob potokih stražijo veverice in lovijo želode, bodisi same bodisi v paru.

Ena sama veverica:

- Zgreši prvi želod, a vedno ujame drugega.
- Če po potoku prideta dva želoda sočasno, lahko ujame le enega.

Dve veverici v paru:

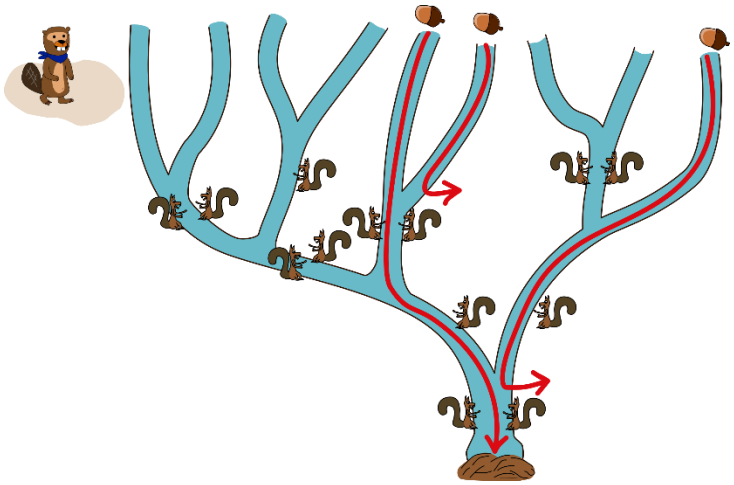
- Ujameta prvi želod, nato pa se prepirata za želod in zgrešita drugi želod.
- Če po potoku prideta dva želoda sočasno, lahko ujameta le enega.

Bober postavi želode v tri različne potoke ter vse spusti sočasno proti jezu. V katere tri potoke naj postavi želode, da bo vsaj en želod prišel mimo veveric do jeza?

Rešitev

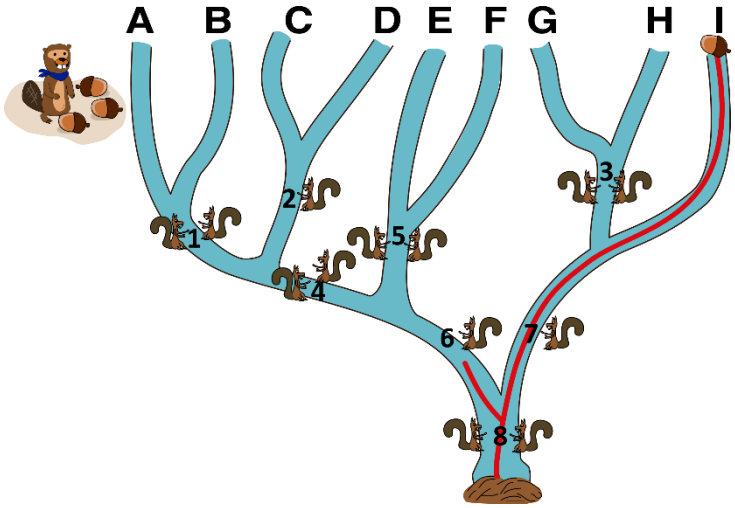
Bober mora želode postaviti v potoke E, F in I.

Spodnja slika prikazuje rešitev. Poti želodov so prikazane z rdečo črto, puščica pa označuje, kje so veverice ujele želod.

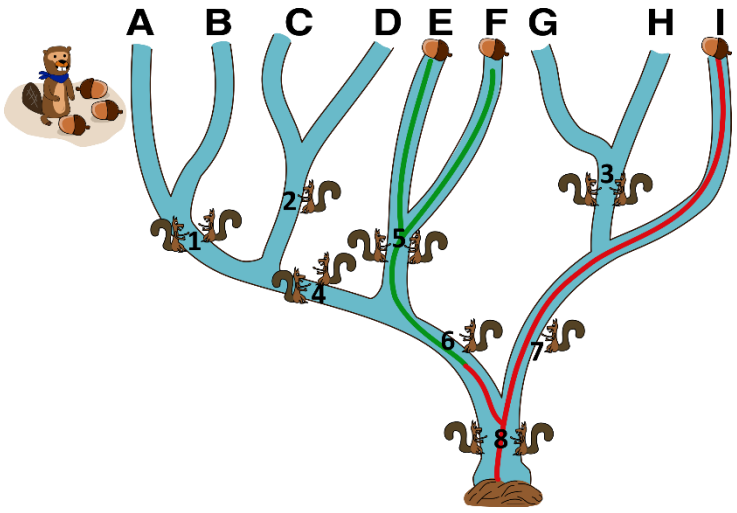


Rešitev naloge najlažje poiščemo tako, da gremo od jeza po potokih navzgor proti toku.

Pri jezu moramo imeti vsaj en želod. Zato morata do zadnjega para veveric (8) priti dva želoda, da bo vsaj eden lahko prišel mimo njiju. Želoda prideta z leve in z desne strani (pot je označena z rdečo črto). Če sledimo poti želoda, ki pride z desne, pridemo še do ene veverice (7). Ker veverica prvega želoda ne ujame, zadostuje le en želod, da ta pride mimo nje. Ta želod lahko pride po potoku I, saj na njem ni nobenih drugih veveric.



Poglejmo še na levo stran po toku navzgor od para veveric 8. Tudi tu naletimo na eno veverico (6), zato zadostuje en sam želod, ki pride mimo nje. Ta želod lahko pride z leve ali z desne strani. Na desni imamo ob poti manj veveric, zato bomo na tej poti potrebovali manj želodov (pot je označena zeleno). Če sledimo poti naprej proti toku, pridemo do para veveric (5). Torej morata do tega para veveric priti dva želoda, eden po potoku E in drugi po potoku F, da bo vsaj en želod nadaljeval pot mimo veveric.



Če bi želod do veverice 6 prišel z leve strani, bi morala do para veveric 4 priti dva želoda, zato bi morali želod postaviti tako v potok C ali D (zadostuje le en želod, da pride mimo veverice 2) kot tudi v oba potoka A in B, saj potrebujemo dva želoda, da pride mimo para veveric 1 vsaj en želod. V tem primeru bi torej potrebovali še tri dodatne želode oz. skupaj štiri želode.

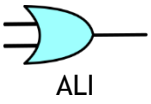
Tako smo ugotovili, da moramo tri želode postaviti v potoke E, F in I, da do jeza pride en želod.

Računalniško ozadje

V nalogi smo s potoki in vevericami pravzaprav zgradili logično vezje.

Veverice v nalogi delujejo kot logični operatorji:

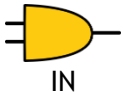
Posamezna veverica se obnaša kot logična vrata ALI (angl. OR).



Potrebujemo le 1 želod, da zagotovimo, da gre 1 želod mimo veverice in nadaljuje po potoku.

Logična operacija ALI pomeni, da mora biti le eden od vhodov resničen (ne pa nujno oba), da je tudi izhod resničen.

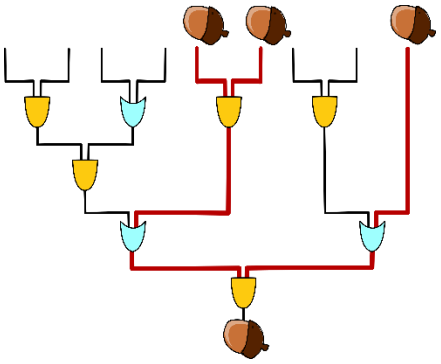
Par veveric se obnaša kot logična vrata IN (angl. AND).



Potrebujemo 2 želoda, da zagotovimo, da gre 1 želod mimo veveric in nadaljuje po potoku.

Logična operacija IN pomeni, da morata biti vedno oba vhoda resnična, da je tudi izhod resničen. Le en resničen vhod ni dovolj.

Sistem potokov in veveric lahko enostavneje prikažemo z diagramom. Veverice zamenjamo z logičnimi vrati IN  oz. ALI  , potoke pa prikažemo s črtami, ki povezujejo logična vrata. Če želod potuje po potoku, je črta vključena (tu označena z debelejšo rdečo), sicer je izključena. Tako lahko rešitev bolj enostavno prikažemo z naslednjim diagramom:



Logična vrata so osnovni gradniki našega digitalnega sveta. V delčku sekunde se odločijo, katero informacijo bodo spustila naprej in katere ne.

Znanstveniki so z uporabo bakterij in encimov izdelali biološka barvila, ki imajo posebno lastnost: ko enemu barvilu dodamo barvilo druge barve (četudi le kapljico), po določenem času reakcije nastane barvilo popolnoma drugačne barve.

Vsako barvilo lahko opišemo z zaporedjem treh nukleotidov iz genske kode barvila (nukleotid je lahko A ali C):

črna	CCC
modra	CCA
zelena	CAC
cian	CAA
rdeča	ACC
vijolična	ACA
rumena	AAC
bela	AAA

Z reakcijo nastalo barvo lahko določimo tako, da pogledamo, kako med seboj reagirajo ustrezni nukleotidi v genskih kodah posameznih barvil:

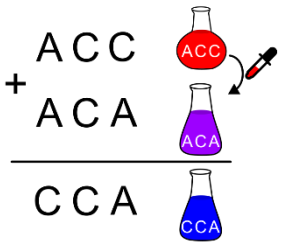
A + A = C

C + C = C

A + C = A

C + A = A

Primer reakcije, če v vijolično barvilo dodamo kapljico rdečega barvila:



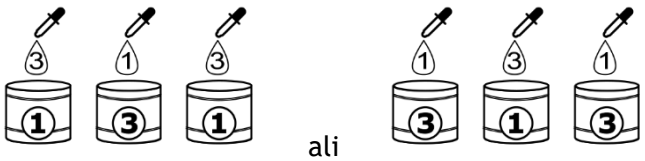
Imamo tri velike, nepremične rezervoarje (z oznakami 1, 2 in 3) z zelenim, vijoličnim in rumenim barvilom. Iz katerega koli rezervoarja lahko vzamemo kapljico barvila in jo dodamo v druge rezervoarje.

Želimo zamenjati barvi v rezervoarjih 1 in 3, tako da bi barvilo v rezervoarju 1 postalo rumeno (AAC), barvilo v rezervoarju 2 bi ostalo vijolično (ACA), barvilo v rezervoarju 3 pa bi postalo zeleno (CAC).

Najmanj koliko kapljic barvil potrebujemo, da to dosežemo?

Rešitev

Barvi lahko zamenjamo z le tremi kapljicami. To lahko naredimo na dva načina: 3→1, 1→3, 3→1 ali pa 1→3, 3→1, 1→3. Oba načina prikazuje spodnja slika:



Najprej preverimo, ali lahko zamenjavo dejansko naredimo s tremi kapljicami, kot prikazuje na primer prva rešitev 3→1, 1→3, 3→1.

Če kapljico rumene (AAC) iz rezervoarja 3 zmešamo z zeleno (CAC) v rezervoarju 1 (3→1), dobimo v rezervoarju 1 rdeče barvilo (AAC + CAC = ACC). Barve v rezervoarjih so sedaj rdeča, vijolična, rumena.

Če kapljico rdeče (ACC) iz rezervoarja 1 zmešamo z rumeno (AAC) v rezervoarju 3 (1→3), dobimo v rezervoarju 3 zeleno barvilo (ACC + AAC = CAC). Barve v rezervoarjih so sedaj rdeča, vijolična, zelena.

Če kapljico zelene (CAC) iz rezervoarja 3 zmešamo z rdečo (ACC) v rezervoarju 1 (3→1), dobimo v rezervoarju 1 rumeno barvilo (CAC + ACC = AAC). Barve v rezervoarjih so sedaj rumena, vijolična, zelena, kar ustreza želeni zamenjavi zelene in rumene barve.

Sedaj pa še preverimo, da zamenjave ne moremo izvesti z manj kot 3 kapljicami. Očitno ena sama kapljica ni dovolj, saj lahko s tem spremenimo barvo le v enem rezervoarju.

Če bi zamenjavo barv lahko naredili z dvema kapljicama, bi morala prva kapljica spremeniti barvo v rezervoarju 1 v rumeno, druga kapljica pa bi morala spremeniti barvo v rezervoarju 3 v zeleno (ali obratno). Razmislimo, kako lahko barvo v rezervoarju 1 spremenimo v rumeno v prvem koraku:

- če ji dodamo kapljico iz 2 (2 → 1), reagira v ACA + CAC = AAA (bela);
- če ji dodamo kapljico iz 3 (3 → 1), reagira v AAC + CAC = ACC (rdeča).

Torej ni možno, da bi barvo v rezervoarju 1 spremenili v želeno barvo (rumeno) z eno samo kapljico iz drugega rezervoarja.

Razmislimo še, ali lahko barvo v rezervoarju 3 spremenimo v zeleno v prvem koraku:

- če ji dodamo kapljico iz 1 (1 → 3), reagira v CAC + AAC = ACC (rdeča);
- če ji dodamo kapljico iz 2 (2 → 3), reagira v ACA + AAC = CAA (cian).

Torej tudi v tem primeru ni možno, da bi barvo v rezervoarju 3 spremenili v želeno barvo (zeleno) z eno samo kapljico iz drugega rezervoarja. Torej le z dvema kapljicama ne moremo zamenjati barv in je najmanjše število potrebnih kapljic 3.

Računalniško ozadje

Pravilo, po katerem reagirata dva nukleotida A in C je logična operacija *ekskluzivni ali* (*XOR*). Če nukleotid A interpretiramo kot resnično in C kot neresnično, vidimo, da je rezultat resničen natanko takrat, ko je resničen natanko eden od obeh operandov. Ali z drugimi besedami, rezultat je resničen, če imata oba operanda različne vrednosti. Če pa imata oba operanda enaki vrednosti (ali resnično ali neresnično), je rezultat neresnično.

Ekskluzivni ali se v računalništvu zelo pogosto uporablja, saj omogoča zamenjavo vrednosti dveh spremenljivk (kot smo v nalogi zamenjali barve dveh barvil) brez potrebe po dodatni spremenljivki. Uporaben je tudi v kriptografiji, saj omogoča enostavno šifriranje sporočil s skritim ključem. Šifrirano sporočilo sestavimo kot sporočilo XOR ključ, odšifriramo pa ga enostavno spet z uporabo XOR in istega ključa.

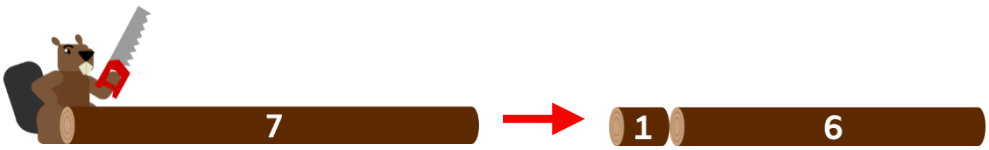
Bober in medved

srednja šola

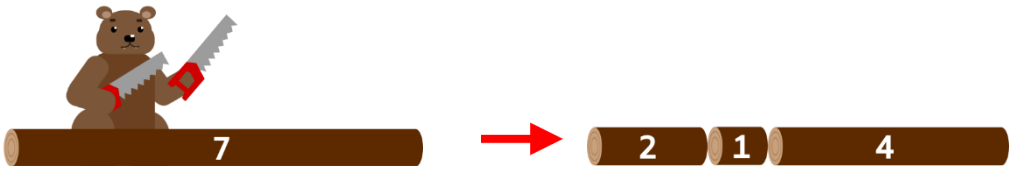
Bober in medved vsak dan žagata 7-metrske hlode na manjše dele, ki jih bober naloži na tovornjak.

Bober potrebuje 1 minuto za vsak rez. Njegov prijatelj medved pa ima dve žagi in lahko sočasno naredi dva reza na razdalji enega metra ali pa le en rez. Za to porabi 1 minuto.

Tako lahko, na primer, bober v 1 minuti razreže 7-metrski hlod na dva dela dolžine 1 meter in 6 metrov:



Podobno lahko medved v 1 minuti razreže 7-metrski hlod na tri dele z dolžinami 2 metra, 1 meter in 4 metre:



Bober in medved lahko hlode žagata sočasno, a ne istega hloda. Ko razžagata hlod, bober potrebuje še 1 minuto, da njegove dele naloži na tovornjak. Medved mu pri tem nikoli ne pomaga.

Danes morata razrezati tri 7-metrske hlode. Pripraviti morata 9 kosov po 1 meter, 3 kose po 2 metra in 2 kosa po 3 metre. Najmanj koliko časa potrebujeta za razrez vseh hlodov in naložitev na tovornjak?

Rešitev

Za to opravilo potrebujeta najmanj 6 minut.

Delo lahko opravita na več načinov. Ena možnost je, da najprej oba razrežeta prvi in drugi hlod, vsak enega. Medtem ko bo bober nalagal razrezane dele na tovornjak, bo medved razrezal še tretji hlod. Nato mora bober na tovornjak naložiti še dele zadnjega hloda:

Minute	Bober	Medved
1–2	Hlod 1: 3 m + 3 m + 1 m (2 reza, 2 min)	Hlod 2: 2 m + 1 m + 2 m + 2 m (1 dvojni rez + 1 rez, 2 min)
3–5	Naloži hlod 1 in hlod 2 (2 min, nato čaka)	Hlod 3: 7 x 1 m (3 dvojni rezi, 3 min)
6	Naloži hlod 3 (1 min)	

Iz zgornjega primera vidimo, da delo lahko opravita v 6 minutah. Ni pa možno, da bi delo opravila prej kot v 6 minutah zaradi naslednjih razlogov:

- Če bober razreže katerikoli hlod, za to potrebuje najmanj 2 minuti (kot je to v primeru prvega hloda zgoraj).
- Bober potrebuje dodatno še 3 minute, da razrezane dele naloži na tovornjak.
- Če bober prvi hlod razreže le z dvema rezoma (torej v 2 minutah), mora medved narediti najmanj pet rezov (dvojnih ali enojnih) na ostalih dveh hlodih. Za to potrebuje 5 minut, nato pa mora bober še naložiti zadnji razrezan hlod, za kar potrebuje še eno minuto. Torej skupaj 6 minut.
- Bober mora vedno delati 1 minuto dlje kot medved, saj mora na tovornjak naložiti še zadnji razrezan hlod. To pomeni, da dela ne moreta opraviti v 5 minutah.
- Če bober ne razreže nobenega hloda (in se ukvarja le z nalaganjem na tovornjak), mora medved razrezati tudi prvi hlod, kar lahko naredi z enim dvojnim rezom (potrebuje 1 minuto). Še vedno pa mu ostaneta za razrez še preostala dva hloda, za katera potrebuje 5 minut. Torej bi medved porabil 6 minut le za rezanje, bober pa bi moral zadnji razrezan hlod še naložiti na tovornjak, kar bi vzelo še dodatno minuto.

Računalniško ozadje

V nalogi sta bober in medved delala sočasno, podobno kot v računalnikih lahko deluje več procesorjev. Vendar pa sta bober in medved dva procesorja, ki delujeta različno: bober je *sekvenčni procesor*, ki lahko izvaja posamezne operacije zaporedno, medtem ko je medved *vzporedni procesor*, ki sočasno izvaja enako operacijo na več podatkih (temu rečemo tudi *en ukaz, več podatkov* ali z angleško kratico SIMD).

V nalogi smo srečali tudi zanimiv koncept vzporedne obdelave z zaporednim urejanjem. To pomeni, da nekatere naloge lahko izvajamo vzporedno, druge pa le zaporedno, zato moramo včasih počakati, da se izvajanje naloge zaključi, preden lahko nadaljujemo.

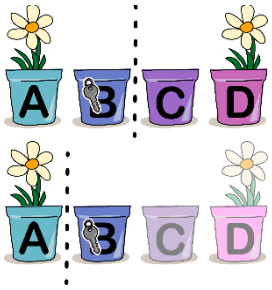
Cvetlični lončki 2

srednja šola

Barbara ima pred vhodnimi vrati vrsto cvetličnih lončkov. Nekateri lončki so prazni, v drugih pa je posajena ena roža. Ključ od vhodnih vrat je skrila v enega izmed lončkov. Prijateljici pove, kako lahko poišče ključ:

> »Najprej si oglej vse lončke. Če je skupno število rož sodo, je ključ skrit v levi polovici lončkov, sicer je skrit v desni polovici. Nato poglej tisto polovico lončkov, v kateri je ključ. Postopek ponavljaj, dokler ti ne ostane en sam lonček – v njem je skrit ključ.«

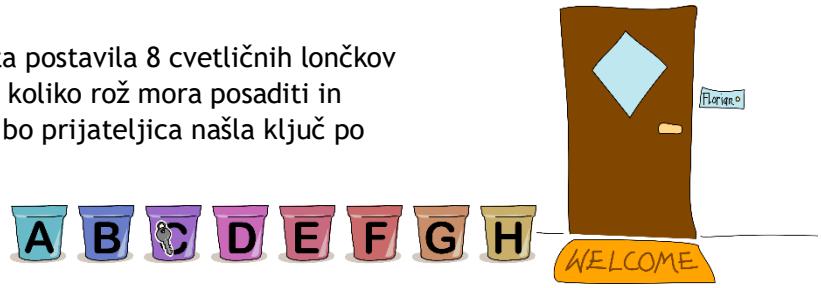
Podala je tudi primer. Če so na pragu 4 lončki in je ključ skrit v lončku B, lahko prijateljica najde ključ takole:



Pogleda vse štiri lončke A, B, C in D. V njih sta posajeni 2 roži, to je ***sodo*** število. Torej mora biti ključ skrit v lončkih na ***levi*** polovici, v A ali B.

Pogleda lončka A in B in vidi, da je posajena ena sama roža. 1 je ***liho*** število. Torej mora biti ključ skrit na ***desni*** polovici, v lončku B.

Naslednji dan je Barbara pred vrata postavila 8 cvetličnih lončkov in ključ skrila v lonček C. Najmanj koliko rož mora posaditi in največ koliko rož lahko posadi, da bo prijateljica našla ključ po opisanem postopku?



Rešitev

Posaditi mora najmanj 2 roži in največ 6 rož.

Rešitev z najmanjšim oz. največjim številom rož je več, saj so rože lahko posajene v različnih lončkih. Eno od možnih rešitev prikazujeta naslednji sliki (levo je rešitev z najmanj rožami, desno pa z največ rožami):

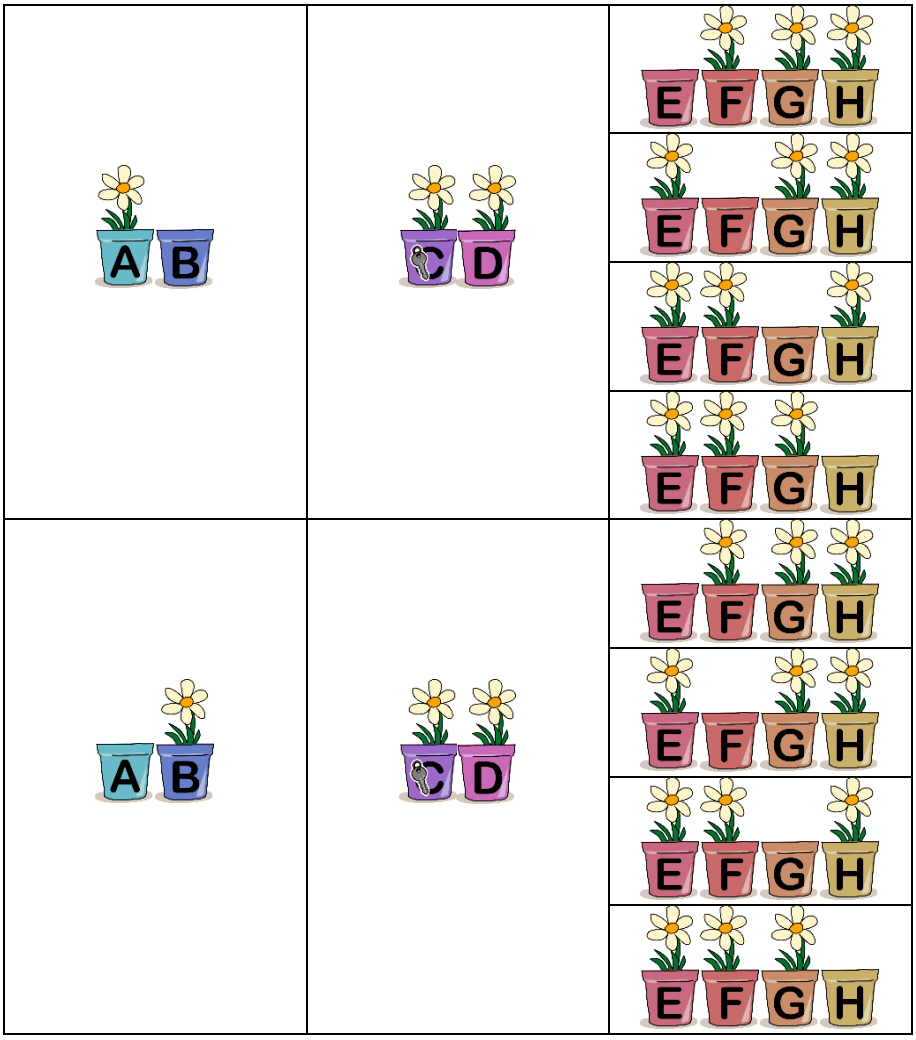
Poglejmo, kako pridemo do rešitve naloge. Pri iskanju ključa moramo najprej prešteti vse rože v lončkih. Ker je ključ v lončku C, moramo v naslednjem koraku pregledati lončke od A do D, torej levo polovico. Zato mora biti v vseh lončkih skupaj posajeno sodo število rož.

V lončkih A, B, C in D mora biti število rož liho, da po navodilih iščemo ključ v desni polovici, torej v lončkih C in D. Na koncu pa mora biti sodo število rož tudi v teh dveh lončkih (C in D), da za ključ pogledamo v levo polovico lončkov, to je lonček C. Torej morajo biti izpolnjeni ti trije pogoji, da lahko najdemo ključ.

In kakšno je najmanjše oz. največje število posajenih rož? Za najmanjše število rož poglej rešitev pri nalogi *Cvetlični lončki 1* (posaditi mora najmanj dve roži). Tu poglejmo še rešitev za največje število rož.

V lončkih C in D mora biti sodo število rož. Če je v vsakem od njiju roža, imamo 2 roži, kar je sodo število. Torej imata lončka C in D oba posajeno rožo. V lončkih A, B, C in D mora biti liho število rož. Največje liho število (ki je manjše ali enako 4) je 3. Če imata lončka C in D vsak po eno rožo, mora biti ena roža tudi v lončkih A in B, da imajo vsi štirje lončki skupaj 3 rože. Torej mora biti 1 roža posajena ali v lončku A ali v lončku B. Ker mora biti skupno število rož v vseh osmih lončkih sodo, na levi polovici pa imamo posajene 3 rože, moramo liho število rož posaditi tudi na desni polovici (lončki E do H). V 4 razpoložljive lončke lahko posadimo največ 3 rože, saj je 3 največje liho število, ki je manjše ali enako 4. Torej je natanko en lonček na desni polovici brez rože, in sicer kateri koli lonček E, F, G ali H. Tako lahko skupaj posadimo največ šest rož.

Spodnja tabela prikazuje vse možne rešitve posaditve 6 rož, teh je 8.



Računalniško ozadje

Barbara je za označevanje položaja ključa uporabila rože. Ali povedano bolj natančno, položaj ključa je zakodirala kot zaporedje lončkov brez rože ali z rožo. Njena prijateljica je lahko

dekodirala položaj ključa iz tega zaporedja. Ta koda je uporabna le zato, ker vsako zaporedje določa natanko en položaj – tako prijateljica točno ve, kje iskati ključ.

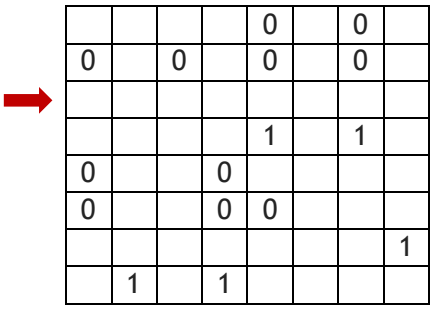
V računalništvu so kode bistvene za predstavitev podatkov v obliki, ki jo lahko obdeluje računalniška strojna oprema, na primer za pošiljanje sporočil prek interneta. Mnoge kode uporabljajo daljša zaporedja, kot bi bila nujno potrebna, saj lahko odvečni deli zaporedja pomagajo zaznati ali celo popraviti napake v primeru, ko so deli zaporedja po nesreči spremenjeni. Take so na primer Hammingove kode.

Dvojiški sudoku

srednja šola

Dvojiški sudoku je izpeljanka navadnega sudokuja. Imamo mrežo 8 x 8, v polja pa lahko vpisujemo le 0 ali 1 tako, da so v vsaki vrstici in v vsakem stolpcu natanko 4 ničle in 4 enice. Poleg tega sta lahko v vrstici ali stolpcu zaporedoma največ dve enaki števki.

Naslednja mreža dvojiškega sudokuja je že delno izpolnjena. Katero zaporedje števk je v tretji vrstici (označena s puščico) pri pravilni rešitvi tega sudokuja?



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

Rešitev

V tretji vrstici je naslednje zaporedje števk: 1 0 1 0 1 0 1 0.

Pravilen odgovor poiščemo tako, da rešimo dvojiški sudoku. Ta ima eno samo rešitev.

Začnemo z upoštevanjem omejitev, da sta lahko zaporedoma največ dve enaki števili. Zato dodamo enico med dve ničli oz. ničlo med enici ter enico pred/za dvema zaporednima ničlama oz. ničlo pred/za dvema zaporednima enicama. Dopolnjena tabela izgleda takole (na novo vpisane številke so rdeče):

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

S postopkom nadaljujemo tudi z upoštevanjem na novo vpisanih števil. Dodamo še:

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

V naslednjem koraku spet nadaljujemo postopek in upoštevamo še v predhodnem koraku vpisana števila:

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

Sedaj imamo nekaj stolpcev in vrstic, v katerih so že vpisane štiri ničle ali štiri enice. Torej jih lahko dopolnimo še z enicami oziroma ničlami:

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

S ponavljanjem pravil, ki smo jih uporabili v predhodnih korakih, lahko vpišemo števila še v vsa preostala polja in iz končne rešitve razberemo zaporedje števil v tretji vrstici:

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

Računalniško ozadje

Nalogo lahko razumemo kot različico *problema z omejitvami* (angl. *Constraint Satisfaction Problem*). To je vrsta problemov, pri katerih morajo biti izpolnjene določene omejitve, da bi našli ustrezno rešitev. V nalogi mora biti vsaka celica v tabeli bodisi 0 bodisi 1, vsaka vrstica in vsak stolpec morata imeti enako število ničel in enic, hkrati pa ne smeta biti več kot dve sosednji celici (vodoravno ali navpično) enaki. To so omejitve te naloge. Algoritem mora zadostiti vsem tem omejitvam, da lahko pravilno reši problem. Razlaga rešitve prikazuje, kako bi ga računalnik lahko rešil z učinkovitim algoritmom.

Algoritme za reševanje problemov z omejitvami uporabljamo na številnih področjih; dva pogosta primera sta načrtovanje urnikov (razporejanje nalog glede na trajanje in nujnost) in dodeljevanje virov (dodeljevanje sredstev različnim nalogam glede na njihove potrebe in čas izvedbe).

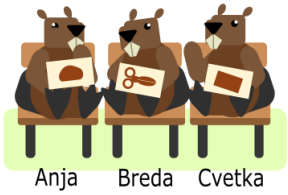
Kamen, papir, škarje

srednja šola

Anja, Breda in Cvetka igrajo različico igre kamen, škarje, papir. Sedijo na stolih in v rokah držijo kartice, na katerih so narisani kamen, škarje in papir. Kartice so vidne vsem.

Spomnimo:

- Kamen premaga škarje.
- Škarje premagajo papir.
- Papir premaga kamen.



Nato določijo, kolikokrat bodo naredile zamenjavo kartic (dve od njih zamenjata kartici). Nato še določijo, kateri dve igralki bosta sodelovali pri vsaki od zamenjav.

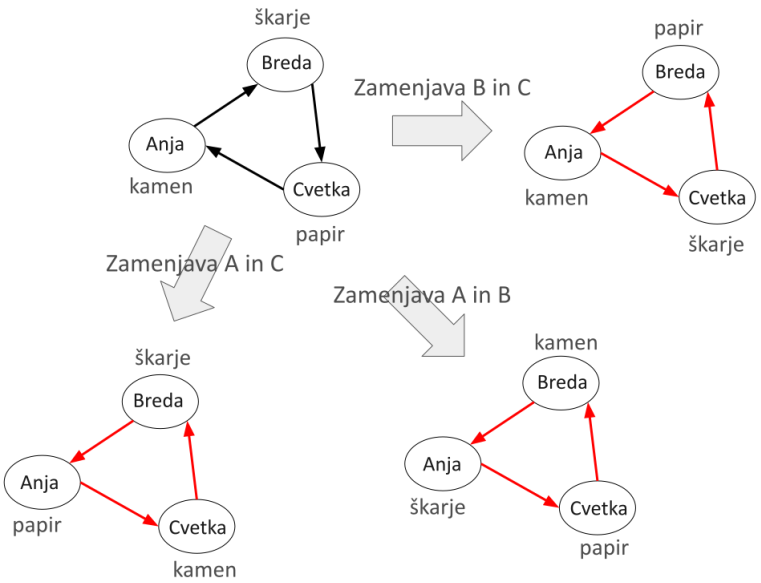
Bredin edini cilj je, da na koncu premaga Cvetko. Kakšno strategijo naj uporabi pri igri?

A) Breda mora le zagotoviti, da ima liho število zamenjav s Cvetko.
B) Ne glede na število zamenjav, ki jih določijo, Breda ne sme nikoli zamenjati kartice s Cvetko.
C) Ne glede na število zamenjav, ki jih določijo, mora Breda vedno zamenjati kartico s Cvetko.
D) Breda mora le zagotoviti, da je skupno število zamenjav sodo, ne glede na to, kdo zamenjuje kartice.

Rešitev

Pravilen odgovor je D.

Nalogo lažje rešimo, če opazimo, da se pri zamenjavi dveh kartic zmage vedno obrnejo, ne glede na to, katero kartico ima kateri igralec. Poglejmo vse tri možne zamenjave in njihove rezultate.



Na sliki puščice prikazujejo rezultat igre. Vidimo, da se pri vsaki zamenjavi zmage obrnejo. Tako bodo po sodem številu zamenjav zmage enake kot na začetku. In po lihem številu zamenjav bodo zmage obrnjene.

Ker želi Breda premagati Cvetko in je taka situacija že na začetku (Breda ima škarje, Cvetka pa papir in škarje premagajo papir), mora Breda poskrbeti le za to, da bodo naredile sodo število zamenjav in s tem ohranile začetno stanje glede zmag.

Računalniško ozadje

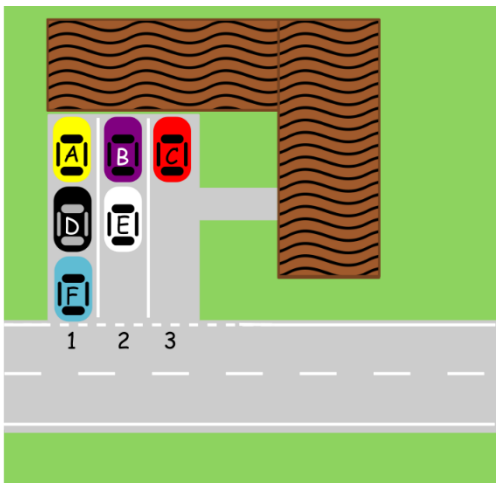
Naloga se tako ukvarja z *invariantami*, to je iskanjem stvari, ki se v določenih okoliščinah ne spremenijo, ali prepoznavanjem okoliščin, v katerih so stvari enake kot prej.

Invariante so ključnega pomena za matematiko in računalništvo, saj nam, če jih uspemo najti, omogočijo zmanjšanje števila primerov, ki jih je treba analizirati, kar omogoča lažje obvladovanje problema in njegovo reševanje.

Parkiranje

srednja šola

Zala je na svojo rojstnodnevno zabavo povabila devet prijateljev. Vsi bodo prišli z avtom in bodo lahko parkirali na dovozu pred Zalino hišo, saj je na njem prostora za natanko devet avtomobilov. Parkirati bodo morali v treh vzporednih vrstah, v vsaki po trije avtomobili eden za drugim. Vsak od gostov lahko ob prihodu parkira na prvem prostem mestu v kateri koli vrsti.



Desna slika prikazuje primer, ko najprej pripelje avto A, nato D kot drugi, tretji je C, sledijo pa še B, E in F. Vidimo, da mora avto F odpeljati z dovoza, preden lahko odpelje avto D.

Gostje bodo na Zalino zabavo prišli v naslednjem vrstnem redu:

Ana, **B**ojan, **C**iril, **D**arja, **E**va, **F**ilip, **G**regor, **H**elena in **I**van.

Z zabave bodo gostje odšli v naslednjem vrstnem redu:

Gregor, **D**arja, **F**ilip, **I**van, **H**elena, **C**iril, **B**ojan, **A**na, **E**va.

Avtomobili gostov so označeni z začetnico njihovih imen. Kako naj gostje parkirajo svoje avtomobile, da noben avto ne blokira drugega, ki mora oditi pred njim?

A) 

B) 

C) 

D) 

Rešitev

Pravilen odgovor je B.

Preverimo situacijo parkiranja pri odgovoru B. V prvi vrsti so parkirani avtomobili A, B in D. To ustreza zaporedju prihodov Ane, Bojana in Darje. V drugi vrsti so C, F in G, kar tudi ustreza vrstnemu redu prihodov: Ciril pride pred Filipom in oba prideta pred Grego. Tudi tretja vrsta ustreza: Eva je prišla pred Heleno in slednja je prišla pred Ivanom. Torej so vsi avtomobili ustrezno parkirani. Preverimo še odhode z zabave. Za avtomobile v prvi vrsti mora Darja zapustiti zabavo pred Bojanom, Bojan pa pred Ano. To ustreza navedenemu zaporedju odhoda gostov. V drugi vrsti imamo C, F in G, zato mora Gregor odpeljati pred Filipom in Filip pred Cirilom. Tudi ta vrstni red ustreza navedenim odhodom gostov. Za avtomobile v tretji vrsti pa mora veljati, da najprej odpelje Ivan, nato Helena in zadnja Eva. Tudi ta vrstni red ustreza navedenim odhodom gostov. Torej so vsi avtomobili ustrezno parkirani in odgovor B je pravilen.

Pri odgovoru A so avtomobili napačno parkirani glede na čas prihoda. Darja je prišla pred Filipom, zato ni mogla parkirati za njim v drugi vrsti. Zato odgovor A ni pravilen.

Pri odgovoru C je problem pri odhodih vozil v drugi vrsti. Prvi z zabave odide Gregor, ki brez težav odpelje. Za njim želi oditi Darja, a ima svoj avto zaparkiran z avtomobiloma Helene in Ivana, ki še nista odšla z zabave. Torej je odgovor C napačen.

Pri odgovoru D pa so vozila sicer parkirana po vrstnem redu prihoda gostov, a z zabave bo prvi odšel Gregor, katerega avto je zaparkiran z avtomobiloma Helene in Ivana, ki bosta z zabave odšla za Grego. Tako je tudi odgovor D napačen.

Računalniško ozadje

Način, kako so gosti parkirali avtomobile na dovozu v vsaki od treh vrst, v računalništvu imenujemo *sklad*. To je podatkovna struktura, ki hrani podatke na poseben način. Sklad deluje podobno kot kup krožnikov. Podatki se kopičijo eden na drugem: nov element vedno dodamo na vrh sklada (operacijo dodajanja imenujemo *push*) in element lahko vedno vzamemo le z vrha sklada, torej zadnji dodan element (operacijo odvzemanja imenujemo *pop*). Zato je sklad pravzaprav neka posebna oblika *vrste*, ki jo označimo z angleško kratico *LIFO*, kar pomeni zadnji noter – prvi ven (angl. *last in, first out*).

V nalogi smo imeli hkrati tri sklade in razvrščanje avtomobilov v njih je podobno računalniškemu problemu, kako razdeliti vire različnim jedrom procesorja pri vzporednem procesiranju.

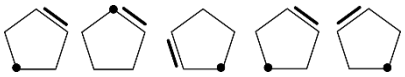
Sporočilo iz vesolja

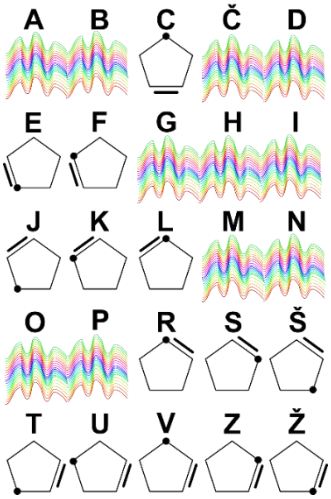
srednja šola

Radioamater Žiga je prestregel signal iz vesolja. V signalu sta bila neznana vesoljska abeceda in sporočilo. Žal pa je zaradi galaktičnega sevanja nekaj črk abecede postalo neberljivih.

Na srečo je Žiga opazil, da abeceda sledi jasnemu pravilu.

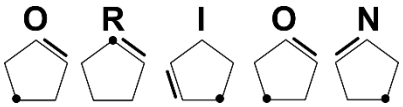
Kakšno je sporočilo?





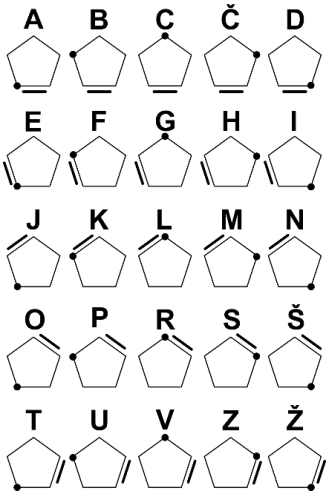
Rešitev

Sporočilo je ORION.



Nerazpoznavne črke v abecedi lahko odkrijemo z upoštevanjem naslednjih treh ugotovitev:

1. Vsak znak vesoljske abecede je petkotnik s črtico in piko. Položaja črtice in pike določata, za katero črko gre.
2. V vsaki vrstici je pet črk. Vse imajo črtico na istem mestu.
3. V vsakem stolpcu je pet črk. Vse imajo piko na istem mestu.



S kombiniranjem vseh 5 x 5 položajev črtic in pik dobimo vseh 25 črk abecede, ki je prikazana na desni.

Nato pa lahko posamezne črke enostavno preberemo iz te tabele.

Računalniško ozadje

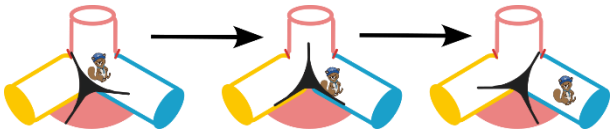
V nalogi smo pokazali idejo, kako lahko predstavimo podatke. V računalništvu temu rečemo kodiranje. Črke slovenske abecede smo predstavili (zakodirali) z znaki neke druge abecede. Za računalnike je kodiranje vsakdanja naloga – uporablja se za predstavitev decimalnih števil, znakov (npr. ASCII in Unicode), pa tudi za bolj kompliciranа kodiranja.

Pri reševanju naloge smo morali uporabiti tudi razpoznavanje vzorcev, saj brez analize ponavljajočega se vzorca v kodiranju ne bi mogli odkriti, kakšni simboli ustrezajo manjkajočim črkam.

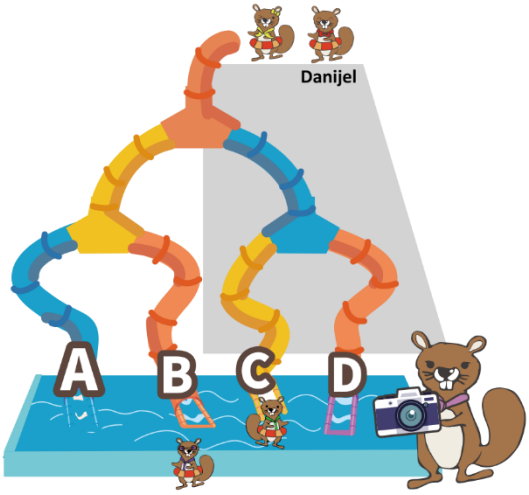
Tobogan

srednja šola

Zabaviščni park v Bobrovi vasi ima novo atrakcijo – vodni tobogan, ki sproti spreminja pot po toboganu. Spodnja slika prikazuje, kako tobogan deluje: na vsakem razcepu je poseben mehanizem, ki nadzoruje smer izhoda. Ta se zamenja ob vsakem prehodu skozi razcep.



Bobrček Danijel želi preizkusiti vodni tobogan, mama bobrovka pa bo njegov pristanek v vodi ovekovečila s fotoaparatom. Da lahko napravi kar najbolj atraktivno sliko, mora vedeti, kje bo Danijel izstopil iz tobogana. Najprej vidi bobra, ki izstopi iz tobogana na izhodu B, naslednji pride na izhodu C. Na vrhu tobogana na vožnjo čaka še en bober, potem pa bo na vrsti že Danijel.



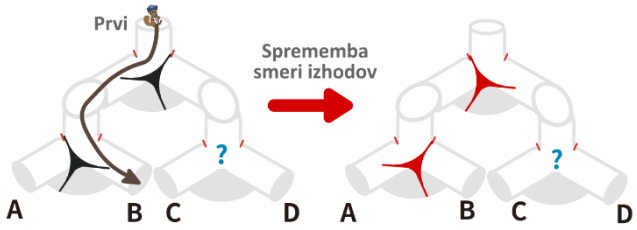
Na katerem izhodu bo iz tobogana izstopil Danijel?

Rešitev

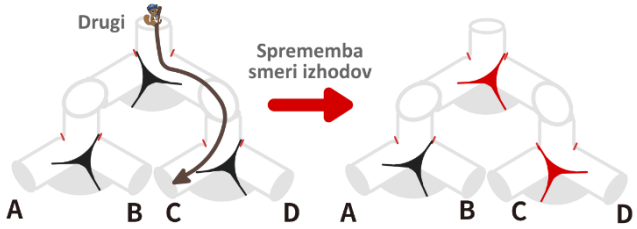
Danijel bo iz tobogana izstopil na izhodu D.

Ker se smer izhoda zamenja pri vsakem prehodu, lahko ugotovimo smeri izhodov posameznih razcepov po tem, ko prva dva bobra uporabita tobogan.

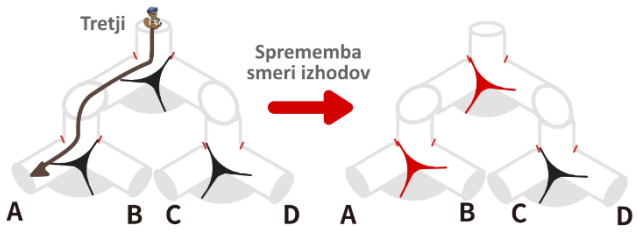
Prvi bober je prišel na izhod B. Iz tega lahko sklepamo, da je šel na prvem razcepu levo in na drugem desno, kot je prikazano na spodnji sliki. Po njegovem prehodu se izhodni smeri teh dveh razcepov spremenita: prvi razcep zdaj vodi v desno, drugi (spodnji levi) pa v levo (situacijo prikazuje desni del slike). Zaenkrat še ne moremo vedeti, v katero stran je obrnjen spodnji desni razcep.



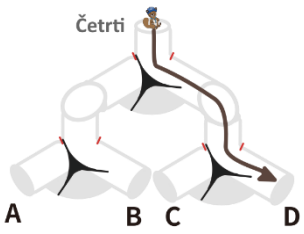
Podobno sklepanje velja za pot drugega bobra – prikazuje jo spodnja slika. Bober je prišel na izhod C, zato je moral na prvem razcepu desno, na drugem pa levo. Torej je drugi (spodnji levi) razcep vodil v levo, preden se je skozenj spustil bober. Po tem pa se je obrnil v desno. Prvi razcep pa se je po spustu bobra obrnil spet levo.



Tako smo za vse tri razcepe tobogana ugotovili, v katero smer trenutno vodijo, levo ali desno. Za naslednjega bobra, ki je na vrsti za spust po toboganu, tako vemo, po kateri poti bo šel: prvi razcep levo, drugi pa tudi levo, tako da pride na izhod A. Nato se oba razcepa obrneta desno. Stanje prikazuje spodnja slika.



Ko se po toboganu spusti Danijel, ga prvi razcep usmeri v desno, drugi pa tudi v desno in tako zaključi vožnjo na izhodu D.



Računalniško ozadje

Razcep v toboganu je podoben preklopnemu digitalnemu vezju, imenovanemu *flip-flop* (bolj učeno se temu reče tudi *bistabilni multivibrator*). To vezje ima dve stabilni stanji in lahko shranjuje informacijo o stanju ter s tem hrani en bit podatkov (eno od dveh stanj predstavlja vrednost 1, drugo pa 0). Vhodni signali lahko spremenijo stanje vezja in tako vezje preklaplja med obema stanjema. Flip-flopi so osnovni elementi za shranjevanje in tudi osnovni gradniki digitalnih elektronskih sistemov, ki se uporabljajo v računalnikih, komunikacijskih napravah in številnih drugih vrstah sistemov.

Pregled nalog

Šolsko tekmovanje

			2.	3.	4.	5.	6.	7.	8.	9.	SŠ
Sestavljanje robotov 1	Kanada		●								
Nariši pošast 1	Mehika		●	●							
Navodila za gradnjo 1	Švica		●	●							
Rože	Urugvaj		●	●							
Rokov robot	Argentina		●	●	●						
Barvanje kvadratov	Slovaška			●							
Aleksov zaklad	Malta			●	●						
Sestavljanje robotov 2	Kanada			●	●						
Ugotovi geslo	Kitajska			●	●						
Letala	Litva				●	●					
Nariši pošast 2	Mehika				●	●					
Navodila za gradnjo 2	Švica				●	●					
Tekma	Hrvaška				●	●					
Bonboni	Pakistan				●	●	●	●			
Skoki	Tajvan				●	●	●	●			
Peščene slike	Južna Koreja				●	●	●	●	●	●	
Emine rože	Brazilija					●					
Bobri in les	Švica					●	●	●			
Perzijska preproga	Iran					●	●	●			
Prečkanje reke	Argentina					●	●	●			
Robot v labirintu 1	Madžarska					●	●	●			
Skrivno sporočilo	Slovaška					●	●	●			
Štetje točk	Japonska					●	●	●			
Kocka	Indija						●	●			
Bibimbap	Južna Koreja						●	●	●	●	
Mačke	Azerbajdžan						●	●	●	●	

			2.	3.	4.	5.	6.	7.	8.	9.	SŠ
Potovanje po Seulu	Južna Koreja						●	●	●	●	
Urejanje majic	Indija						●	●	●	●	
Razvrščanje jabolk	Tajvan						●	●	●	●	●
Cvetlični lončki 1	Nemčija								●	●	
Vedno desno	Indija								●	●	
Cvetlični vrt	Ciper								●	●	●
Karta svetlosti	Nemčija								●	●	●
Luči	Kanada								●	●	●
Meglen dan	Portugalska								●	●	●
Prevajalni sistem	Južna Koreja								●	●	●
Robot v labirintu 2	Madžarska								●	●	●
Sekanje dreves	Ciper								●	●	●
Želodi	Švica								●	●	●
Biobarve	Italija										●
Bober in medved	Pakistan										●
Cvetlični lončki 2	Nemčija										●
Dvojiški sudoku	Poljska										●
Kamen, papir, škarje	Brazilija										●
Parkiranje	Slovenija										●
Sporočilo iz vesolja	Švica										●
Tobogan	Tajvan										●

Avtorji nalog

Sarah Chan, Kanada (Sestavljanje robotov 1 in 2)

Maria Cepeda, Mehika (Nariši pošast 1 in 2)

Susanne Datzko-Thut, Švica (Navodila za gradnjo 1 in 2)

Lucía Crivelli, Urugvaj (Rože)

Raquel Viviana Rodríguez, Argentina (Rokov robot)

Monika Tomcsányiová, Slovaška (Barvanje kvadratov)

Leonard Busuttil, Malta (Aleksov zaklad)

Zhang Dongshuang, Kitajska (Ugotovi geslo)

Valentina Dagienė, Litva (Letala)

Kristina Slišurić in Darija Dasović, Hrvaška (Tekma)

Andrei-Mihai Gavriliu, Pakistan (Bonboni)

Ling-Chian Chang, Tajvan (Skoki)

Joohyun Kim, Hong Jin Yeh in Jihye Kim, Južna Koreja (Peščene slike)

Vitória Quaresma, Brazilija (Emine rože)

Susanne Datzko-Thut, Švica (Bobri in les)

Mehdi Razaghiha, Iran (Perzijska preproga)

María Eugenia Heim, Argentina (Prečkanje reke)

Zsuzsa Pluhár, Madžarska (Robot v labirintu 1 in 2)

Andrea Hrušecká, Slovaška (Skrivno sporočilo)

Takeshi Watanabe in Maiko Shimabuku, Japonska (Štetje točk)

Madhavan Mukund, Indija (Kocka)

Doyong Kim, Byeonggyu Cho in Jihye Kim, Južna Koreja (Bibimbap)

Asmar Mammadova, Azerbajdžan (Mačke)

Doyong Kim, Dong Yoon Kim in Jihye Kim, Južna Koreja (Potovanje po Seulu)

Ashwini Chandrashekhar in Bharath Arunachalam, Indija (Urejanje majic)

Chin-hsin Sun, Tajvan (Razvrščanje jabolk)

Margareta Schlüter, Nemčija (Cvetlični lončki 1 in 2)

Shaumik Khanna, Diya Shah in Parvathy Subramanian, Indija (Vedno desno)

Nikolaos Stratis, Ciper (Cvetlični vrt)

Michael Weigend, Nemčija (Karta svetlosti)

Christine Vender, Kanada (Luči)

Pedro Ribeiro, Portugalska (Meglen dan)

Hyunseok Jeon in Doyong Kim, Južna Koreja (Prevajalni sistem)

Nikolaos Stratis, Ciper (Sekanje dreves)

Susanne Datzko-Thut, Juraj Hromkovič in Patricia Heckendorn, Švica (Želodi)

Mattia Monga, Italija (Biobarve)

Marios Omar Choudary, Pakistan (Bober in medved)

Maciej M. Sysło, Poljska (Dvojiški sudoku)

Leonardo Barichello, Brazilija (Kamen, papir, Škarje)

Blaž Kelvišar, Slovenija (Parkiranje)

Jens Hartmann, Nemčija (Sporočilo iz vesolja)

Chun-Cheng Chen, Tajvan (Tobogan)

Soustvarjalci

Matanat Ahmadova, Azerbajdžan

Zakiya Alizada, Azerbajdžan

Ezraa Almajhad, Saudova Arabija

Kanan Asgarli, Azerbajdžan

Maxat Aubakirov, Kazahstan

Masiar Babazadeh, Švica

Meeri-Liisa Beste, Nemčija

Javier Bilbao, Španija

Zainab Abdullah Yousef Bohulaigh, Južna Afrika

Cesar Bolanos, Portoriko

Eugenio Bravo, Španija

Leonardo Cavalcante, Brazilija

Špela Cerar, Slovenija

Vladimir Costas, Bolivija

Andrew Csizmadia, Velika Britanija

Maria Cujdikova, Slovaška

Susanne Datzko-Thut, Švica

Justina Dauksaite, Nizozemska

Diane Dowling, Velika Britanija

Mónica Ferro, Argentina

Bence Gaál, Madžarska

Gabriela Gomez Pasquali, Paragvaj

Le Thi Huong, Filipini

Cecilia Iglesias, Argentina

Alisher Ikramov, Uzbekistan

Thomas Ioannou, Ciper

Takeharu Ishizuka, Japonska

Yong Ju Jeon, Južna Koreja

Asterios Karagiannis, Grčija

Alenka Kavčič, Slovenija

Deyong Kim, Južna Koreja

Jia-ling Koh, Tajvan

Sophie Koh, Singapur

Anne-Marie Kristiansen, Danska

Greg Lee, Tajvan

Taina Lehtimaki, Irska

Gunwoong Lim, Južna Koreja

Judith Lin, Tajvan
Linda Mannila, Švedska
Yoshiaki Matsuzawa, Japonska
Yi-Hsiu Mei, Tajvan
Hamed Mohebbi, Iran
Mattia Monga, Italija
Anna Morpurgo, Italija
Kamohelo Motloung, Južna Afrika
Santiago Olivera, Urugvaj
Ayeong Park, Južna Koreja
Suchan Park, Južna Koreja
Alex Parry, Velika Britanija
Praphan Pavarangkoon, Tajska
Jean-Philippe Pellet, Švica
Emmanuel Plan, Filipini
Wolfgang Pohl, Nemčija
Camila Porto, Urugvaj
Rok Požar, Slovenija
J.P. Pretti, Kanada
Andrea Rocca, Argentina
Nežka Rugelj, Slovenija
Barbara Schmidt-Thieme, Nemčija
Vipul Shah, Indija
Bronius Skūpas, Litva
Jacqueline Staub, Namčija
Gi Soong Chee, Singapur
Alieke Stijf, Nizozemska
Seiichi Tani, Japonska
Christine Vender, Kanada
Michael Weigend, Nemčija
Kyra Willekes, Nizozemska
Mira Wittenberg, Nemčija
Yi Shan Yeh, Tajvan
Azim Yusof, Brunej
Chen Zhengrong, Kitajska