Bober 2023/24

Bebras

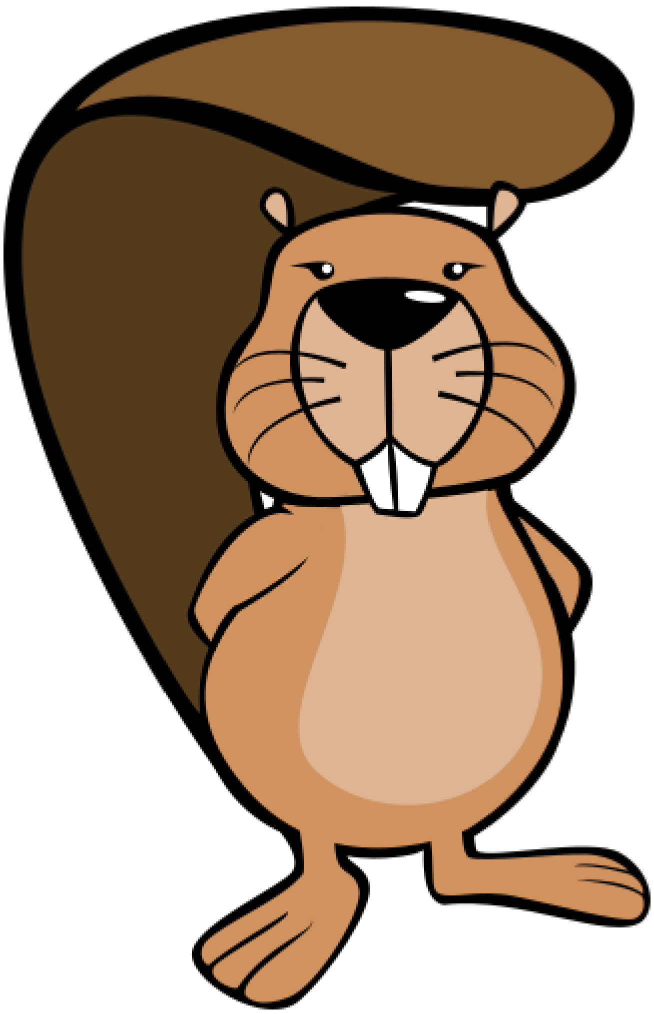
ACM Slovenija

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

Alenka Kavčič (UL FRI)
Nežka Rugelj (OŠ Trzin)
Špela Cerar (UL PEF)

Pomoč pri prevodu:

Roman Bobnarič (Gimnazija Ormož)



Naloge in rešitve

ACM Slovenija

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

Alenka Kavčič (UL FRI)
Nežka Rugelj (OŠ Trzin)
Špela Cerar (UL PEF)

Pomoč pri prevodu:

Roman Bobnarič (Gimnazija Ormož)

Razvoj tekmovalnega sistema:

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

Kazalo nalog

Sosedje

2. in 3. razred

V vasi živijo zajci , rakuni , pujsi  in sove . Spodnja slika prikazuje, v kateri hiši živi katera žival. 3 hiše (na mestih A, B in C) so še prazne.

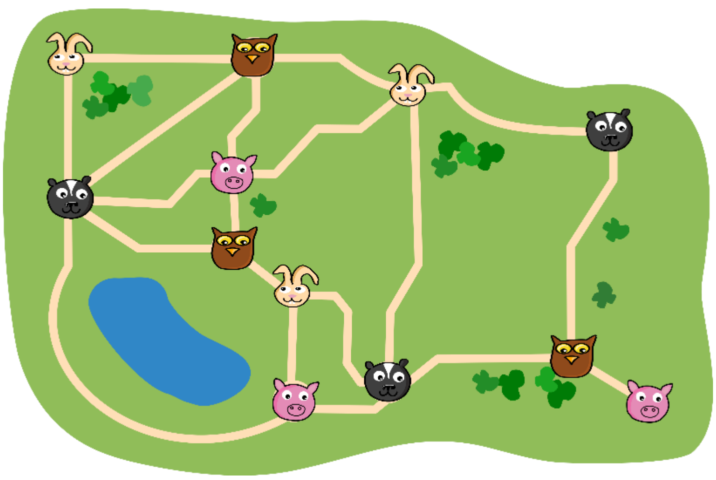
V vas se priselijo pujs, rakun in sova. Vaščani želijo, da v sosednjih hišah, povezanih s potmi, živijo različne živali, na primer noben zajec nima za soseda drugega zajca.

Poveži nove prebivalce s hišami, v katerih bodo živeli.

Rešitev

V hišo na mestu A se vseli sova, saj ima za sosede dva zajca, pujsa in rakuna.

V hiši na mestu B ne more živeti rakun, saj je rakun že v sosednji hiši, zato se vanjo vseli pujs, rakun pa se vseli v hišo na mestu C.



Računalniško ozadje

V nalogi smo srečali graf. Hišice predstavljajo vozlišča, poti med njimi pa povezave. Vrsta živali, ki živi v posamezni hiši, lagko predstavlja barvo vozlišča. Tudi računalničarji in matematiki si z barvanjem vozlišč v grafih pomagamo pri reševanju zahtevnejših problemov.

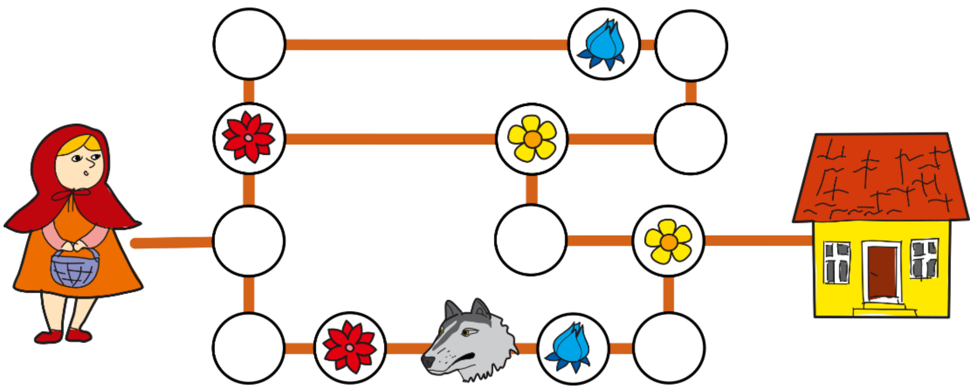
Rože

2. in 3. razred

Rdeča kapica želi na poti k babici nabrati šopek rož. V šopku želi imeti vsaj:

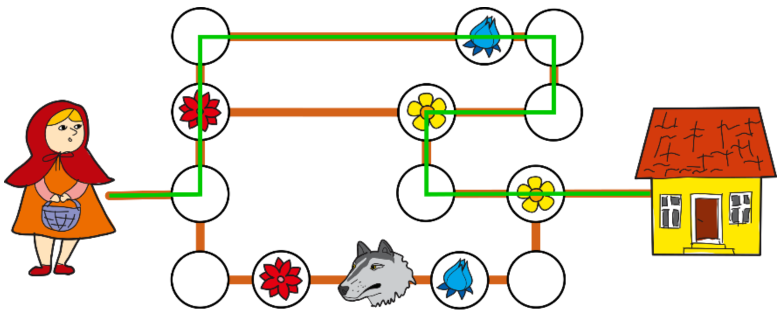
- eno zlatico ,
- eno rdečo vrtnico  in
- en moder tulipan .

Ker se ji mudi, bo šla po isti poti le enkrat. Označi pot, po kateri bo deklica nabrala vse tri vrste rož, a pazi, da NE sreča volka!



Rešitev

Deklica mora na prvem križišču zaviti levo, da se izogne volku. Pri rdeči vrtnici mora nadaljevati naravnost naprej, da lahko nabere še moder tulipan, nato pa pot nadaljuje do babičina hiše, kot je narisano na spodnji sliki.

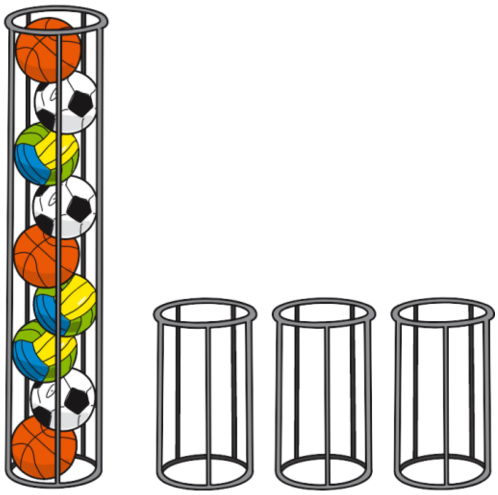


Računalniško ozadje

Pri tej nalogi se je Rdeča kapica sprehodila po grafu do babičine hiške in se pri tem elegantno izognila volku. Tako kot Rdeča kapica tudi računalniki iščejo poti od enega mesta do drugega, pri tem pa upoštevajo omejitve, kot so na primer zaprte ceste in zastoji.

Žoge

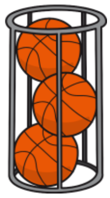
2. in 3. razred



Sara je v stojalo zložila 9 žog za 3 različne športe. Žoge želi razvrstiti v tri manjša stojala, v vsako stojalo želi dati le žoge iste vrste. Ko Sara z vrha velikega stojala vzame žogo, jo odloži v svoje stojalo.

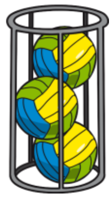
Katero od stojal bo Sara najprej napolnila, če so žoge na začetku zložene, kot kaže slika?

A)



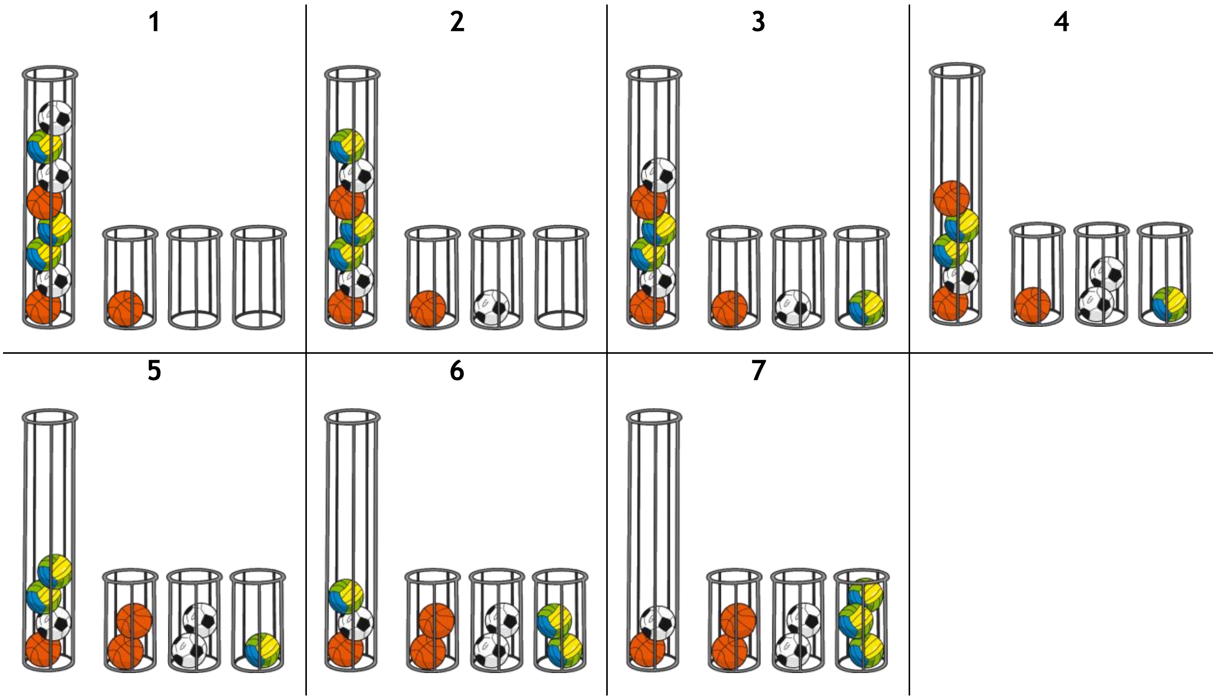
B)

C)



Rešitev

Sara bo najprej napolnila stojalo z žogami za odbojko, to je odgovor C. Sara žoge razvrsti v naslednjem vrstnem redu:



Računalniško ozadje

Stojalo za žoge v naši nalogi predstavlja sklad. Sklad je način shranjevanja podatkov tako, da lahko vedno najprej dostopamo do zadnjega shranjenega podatka.

Dežnik

2. in 3. razred

To je Anin dežnik:

Tudi ena od naslednjih slik prikazuje ta Anin dežnik. Katera?

A)

B)

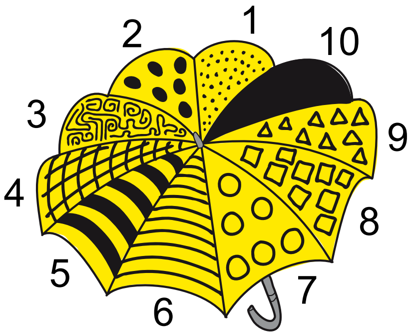
C)

D)

Rešitev

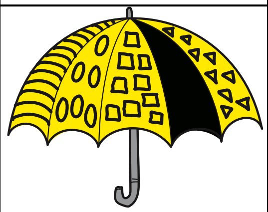
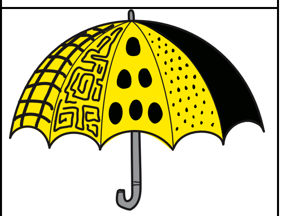
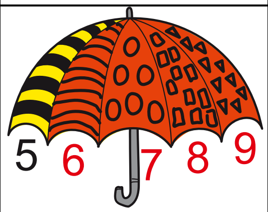
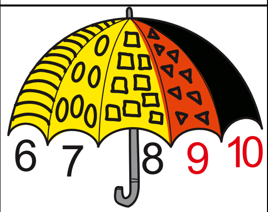
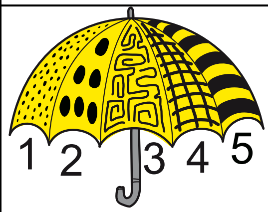
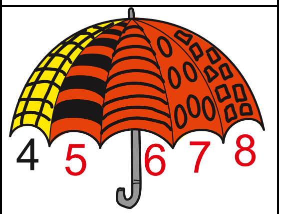
Pravilen odgovor je C. Vsak vzorec se na Aninem dežniku pojavi natančno enkrat, zato lahko slike primerjamo na naslednji način:

1) Pogledamo najbolj levi vzorec na sliki in ga poiščemo na Aninem dežniku.



2) Preverimo, če zaporedje vzorcev ustreza zaporedju vzorcev na Aninem dežniku.

Na vsaki sliki ponujenih odgovorov vidimo zaporedje petih vzorcev, zato lahko preverimo le skladnost polovice zaporedja vzorcev na Aninem dežniku. Ne glede na to, ima le dežnik na sliki C enako zaporedje vzorcev kot Anin dežnik.

	A	B	C	D
Ponujeni odgovori				
Anin dežnik				

Računalniško ozadje

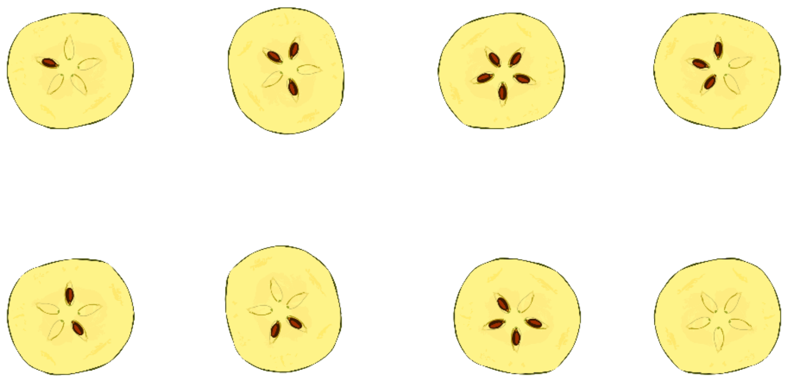
Za rešitev problemov računalničarji pogosto iščemo vzorce in jih izkoristimo, da najemo čim boljšo rešitev.

Jabolka

2. do 4. razred

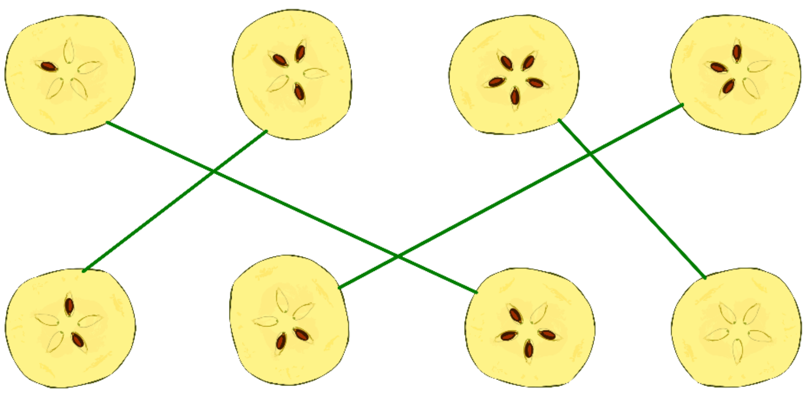
Babica je prerezala 4 jabolka na polovice. Vsako jabolko je imelo 5 pešk. Vsaka peška je ostala samo na eni polovici jabolka.

Poveži vsako polovico jabolka z drugo polovico istega jabolka.



Rešitev

Pravilna rešitev je naslednja:



Pri vsaki polovici jabolka je potrebno pogledati, na katerih mestih imajo oziroma nimajo pešk.

Računalniško ozadje

Da je vsaka peška ostala bodisi na eni bodisi na drugi polovici jabolka v logiki predstavlja logično operacijo ekskluzivni ali.

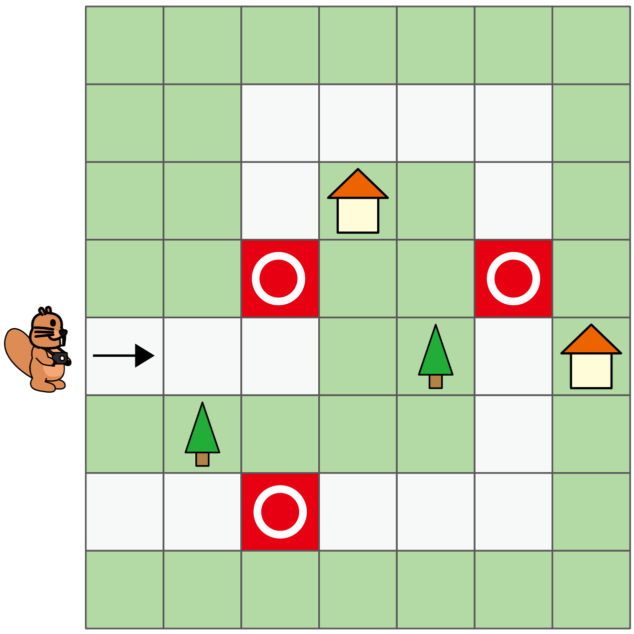
Slike s poti 1

3. in 4. razred

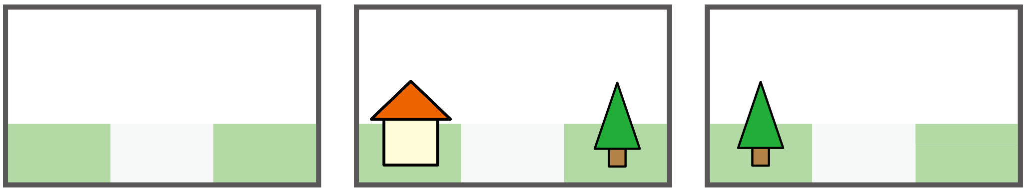
Bober je šel na sprehod in po poti na treh mestih posnel fotografije. Pot sprehoda je prikazana na zemljevidu na desni sliki. Mesta, s katerih je posnel fotografije, so na zemljevidu označena z belim krogom v rdečem kvadratu.

Fotografije prikazujejo, kaj bober z označenih mest na zemljevidu vidi na mestu pred seboj ter levo in desno od tega mesta.

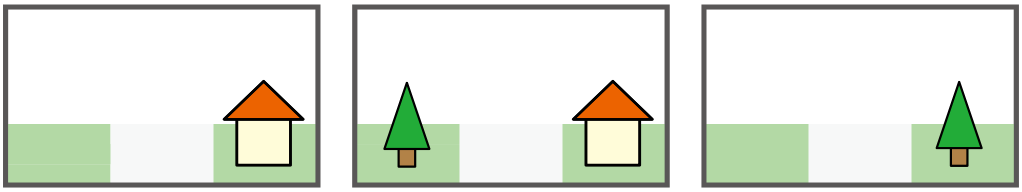
Katere tri fotografije je bober posnel z označenih mest?



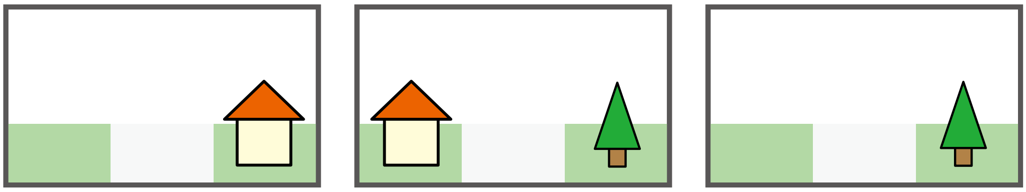
A)



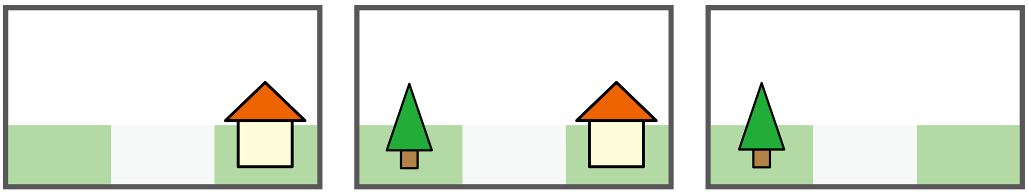
B)



C)



D)



Rešitev

Pravilen odgovor je C.

Na prvi fotografiji je pred bobrom pot, na levi travnik in na desni hiša, zato A ni pravilen odgovor.

Na drugi fotografiji je pred bobrom pot, na levi hiša in na desni drevo, zato odgovora B in D nista pravilna.

Na tretji fotografiji je pred bobrom pot, na levi travnik in na desni drevo, kar prav tako ustreza odgovoru C, ki je pravilen.

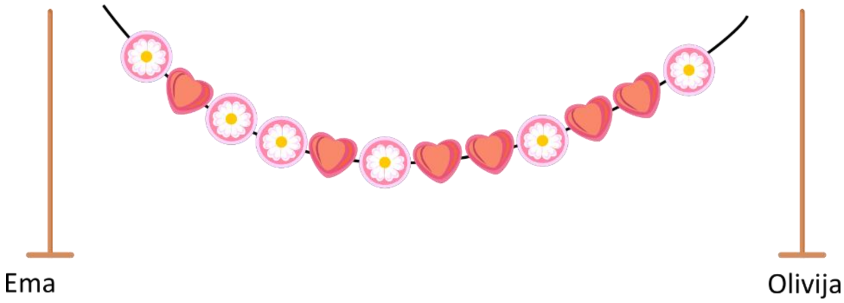
Računalniško ozadje

Računalniški vid je pomemben del računalništva, saj nam omogoča, da se z njegovo pomočjo roboti orientirajo po prostoru.

Ogrlica

3. do 5. razred

Ema in Olivija želita za svojo mamo narediti tako ogrlico:

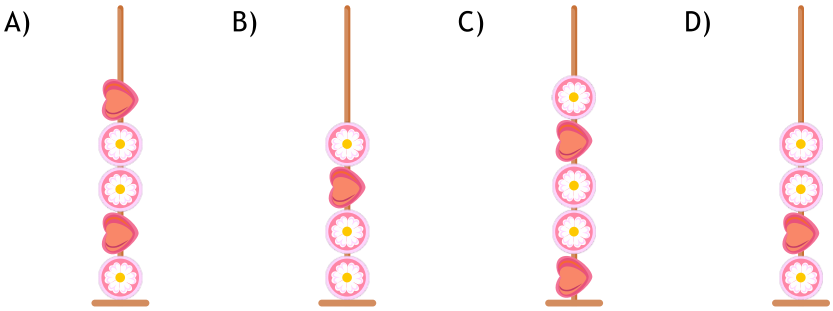


Ogrlico izdelujeta skupaj in obe hkrati. Ema nanjo dodaja perlice z leve strani, Olivija pa z desne.

Pri izdelavi ogrlice si pomagata vsaka s svojim stojalom (na sliki), na katerega natakneta perlice. Nato na vrh obeh stojal pritrdita vrvico in s stojal potisneta perlice na vrvico.

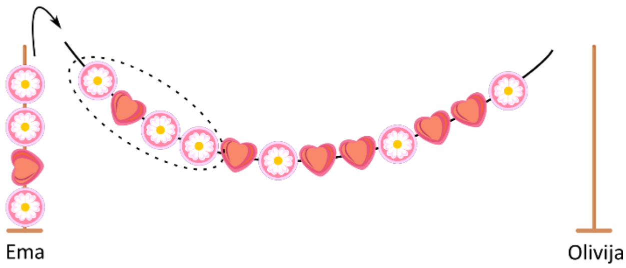
Ema ima z dodajanjem perlic na stojalo več težav, zato se dogovorita, da za vsako perlico, ki jo doda na svoje stojalo Ema, Olivija doda na svoje stojalo dve.

Kako morajo biti na Emino stojalo nanizane perlice, da bo lahko na stojalo pritrdila vrvico in skupaj z Olivijo dokončala ogrlico?



Rešitev

Na končni ogrlici bo 12 perlic. Ema doda 4 perlice, Olivija pa dvakrat več perlic, to je 8 perlic. Zato odgovora A in C nista pravilna, ker je na Eminem stojalu 5 perlic. Ker bo Ema na stojalo pritrdila vrvico in nanjo premaknila perlice, mora te na stojalo nanizati v obratnem vrstnem redu, kot jih bo dala na ogrlico, zato je pravilen odgovor D.



Računalniško ozadje

Pri nizanju perlic na stojalo morata Ema in Olivija uporabiti algoritem zadnji naprej, ki se v računalništvu uporablja, predvsem kadar delamo s tako imenovanim skladom.

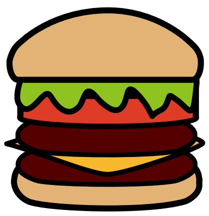
Hamburger

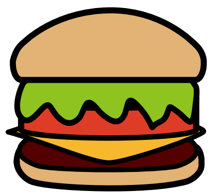
3. do 5. razred

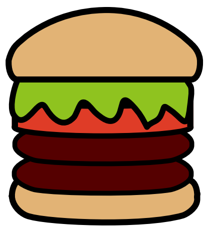
Bor je želel pripraviti okusen hamburger. Jelka ga je skrivaj opazovala pri pripravi in si zapisala postopek, po katerem je Bor dodajal in odstranjeval sestavine. Njeni zapiski izgledajo takole:

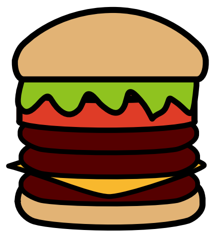
Znak  pomeni, da je Bor odvzel eno sestavino z vrha hamburgerja.

Katerega od naslednjih hamburgerjev je sestavil Bor?

A) 

B) 

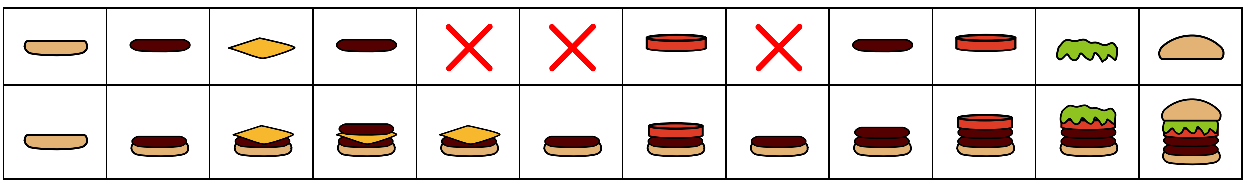
C) 

D) 

Rešitev

Pravilen odgovor je C.

Če sledimo Borovim korakom pri pripravi hamburgerja, dobimo naslednji rezultat:



Bor je dodal 9 sestavin in 3 odvzel, zato mora biti hamburger sestavljen iz 6 sestavin. Hamburger A jih vsebuje 7, hamburger D pa 8, zato to nista pravilna odgovora. Hamburger B vsebuje sir, ki pa ga je Bor vzel iz hamburgerja, zato tudi to ni pravilen odgovor.

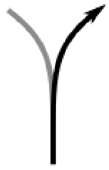
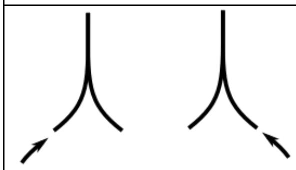
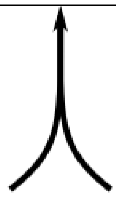
Računalniško ozadje

Zaporedje dodajanja in odvzemanja sestavin iz hamburgerja je pravzaprav program, s pomočjo katerega lahko Jelka sestavi enak hamburger, kot ga je sestavil Bor.

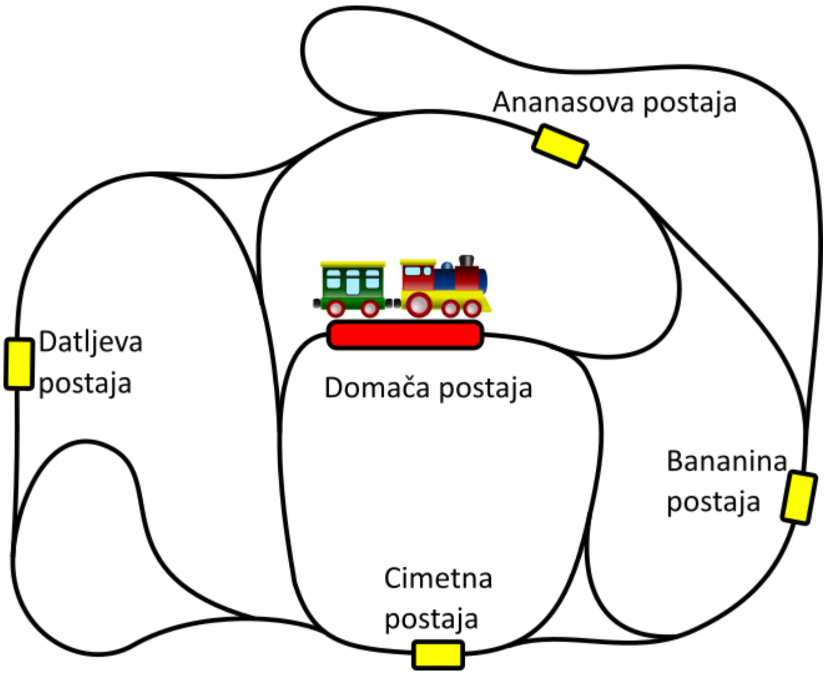
Železnica

4. razred

Tinkara je sestavila železnico z eno domačo in še 4 drugimi postajami. Vlak, ki vozi po tej železnici, lahko upravlja s programom. Poda mu zaporedje ukazov L (levo) in D (desno), ki pomenita, kam naj zavije, ko pride na razcep. Ko izvede vse ukaze, se vlak ustavi na naslednji postaji.

Prihod v razcep	Ukaz	Smer potovanja vlaka po izvedenem ukazu
	L	 Vlak od razcepa potuje po levem tiru.
	D	 Vlak od razcepa potuje po desnem tiru.
	V tem primeru vlak ne izvede nobenega ukaza.	 Vlak nadaljuje naravnost.

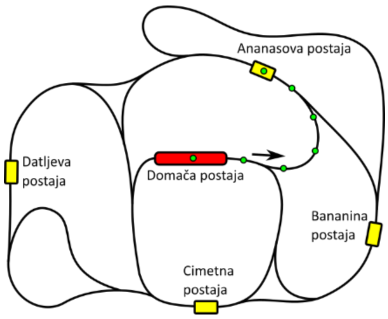
Vlak začne na domači postaji in vozi v smeri, v katero je obrnjen. Na sliki **obkroži postajo**, na kateri se bo vlak ustavil po izvedenem zaporedju ukazov **LLDD**.



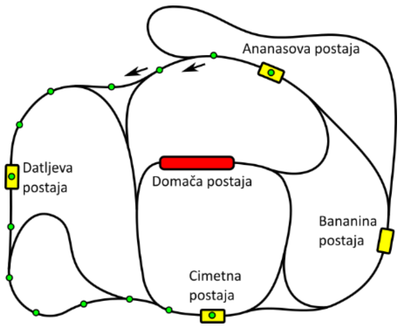
Rešitev

Vlak se bo ustavil na Bananini postaji.

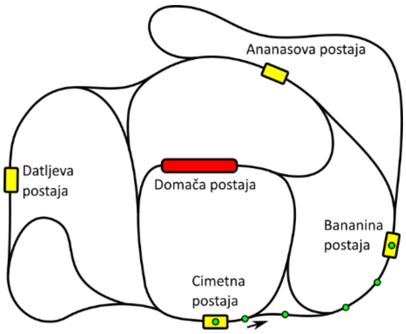
Vlak na prvem razcepu izvede ukaz L (levo). Naslednji razcep je tipa, kjer vlak nadaljuje naravnost, in pride mimo Ananasove postaje (njegovo pot smo označili z zelenimi pikami).



Na naslednjem razcepu izvede ukaz L (levo), takoj na naslednjem pa D (desno), sledi še eno križišče, kjer vlak nadaljuje naravnost. Vlak pripelje do Datljeve postaje in pot nadaljuje mimo nje, saj ima še en ukaz. Ko odpelje z Datljeve postaje, prečka 3 razcepe, kjer nadaljuje pot naravnost, pri tem pride do Cimetove postaje.



Za Cimetovo postajo v razcepu nadaljuje pot po desnem tiru in tako izvede še zadnji ukaz. Pri naslednjem razcepu potuje naravnost in se ustavi na Bananini postaji.



Računalniško ozadje

Vlak v nalogi sprejema preproste ukaze, ki opisujejo njegovo pot. Zaporedje teh ukazov je primer preprostega programa za upravljanjem tega vlaka.

Rikolini

4. razred

V vesolju obstaja ljudstvo Rikolinov. Filip se želi naučiti, kako izgledajo. Ogleda si spodnjih pet fotografij različnih Rikolinov in naredi nekaj zapiskov, ki natančno opisujejo, kaj vidi.

Nato Filip najde še eno fotografijo nekega Rikolina:

Ob tej fotografiji je ugotovil, da je v njegovih zapiskih napaka. Obkroži, katera trditev je zagotovo napačna.

A) Vsak Rikolin ima zobe.
B) Nekateri Rikolini imajo krila.
C) Rikolini imajo ali roge ali troje oči, nikakor pa ne obojega hkrati.
D) Če imajo Rikolini natanko dve roki, imajo tudi natanko dve nogi.

Rešitev

Napaka je bila v trditvi D. Ta trditev glede na novo fotografijo ne velja, saj ima Rikolin dve roki, a nima dveh nog.

Trditev A je lahko resnična, saj imajo Rikolini na vseh šestih fotografijah zobe.

Trditev B je zagotovo resnična, saj ima Rikolin na drugi fotografiji krila.

Tudi trditev C je lahko resnična. Rikolini na prvi, drugi, peti in šesti fotografiji imajo roge in dvoje oči, Rikolina na tretji in četrti fotografiji pa sta brez rogov in imata troje oči.

Računalniško ozadje

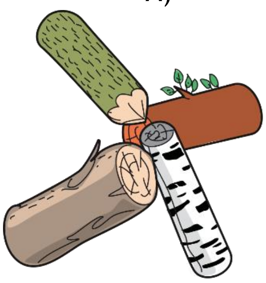
Podobno kot smo mi preverjali veljavnost trditev v ponujenih odgovorih, računalnik pri strojnem učenju v velikih količinah podatkov išče pravila, na podlagi katerih lahko določi ali je na primer na sliki rikolin ali kakšno drugo bitje.

Fotografija

4. in 5. razred

Bober je ravnokar naredil fotografijo. Je ena od spodnjih štirih. Katera?

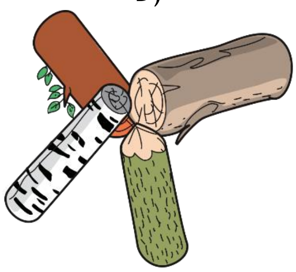
A)



B)

C)

D)



Rešitev

Pravilni odgovor je A.

Da lahko ugotovimo, kako polena izgledajo na fotografiji, moramo ugotoviti, katera polena so levo oz. desno od posameznega polena. Zeleno špičasto poleno (ki izgleda kot svinčnik), je med dvema rjavima polenoma. Zato sliki D in C nista pravi. Opazimo tudi, da je bober, ko je naredil fotografijo, videl zeleno špičasto poleno levo od velikega rjavega polena (brez listov). Torej tudi slika A ni prava. Torej je le slika A tista, ki jo je lahko naredil bober. Preverimo še, ali je vrstni red polen na sliki res pravilen. Levo od špičastega polena mora biti rjavo poleno z listi, levo od njega črno-belo poleno in levo od slednjega veliko rjavo poleno. In tak je vrstni red polen tudi na sliki A.

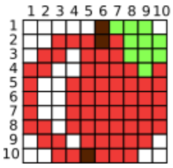
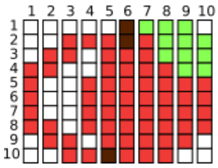
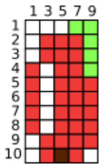
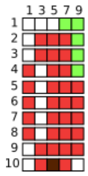
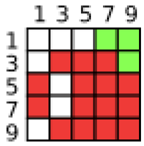
Računalniško ozadje

V tej nalogi nas je zanimal vrstni red polen, ne pa njihove specifične pozicije na sliki. Torej smo upoštevali posebne povezave med poleni: vsako poleno je povezano s polenom, ki je na njegovi desni strani, in s tistim na levi strani. Na podoben način lahko shranjujemo podatke tudi v računalniku: vsaka vrednost (tj. podatek, npr. število ali črka) ima povezavo na njena dva soseda, levega in desnega. Tako strukturo imenujemo dvojno povezan seznam. Tak seznam je zelo fleksibilen: hrani lahko poljubno podatkov, podatke pa lahko zelo hitro dodamo v seznam ali brišemo iz njega.

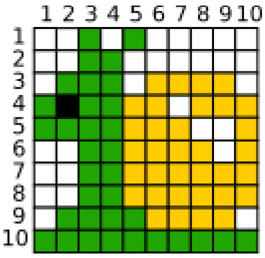
Zmanjšanje fotografij

4. in 5. razred

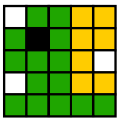
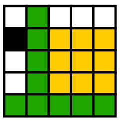
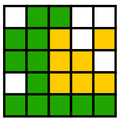
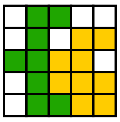
Mojca fotografira jabolko. To fotografijo želi poslati Katarini preko aplikacije Bajber. Da je pošiljanje hitrejše, aplikacija uporablja spodnjo tehniko za zmanjševanje slik.

				
Slika, ki jo pošlje Mojca.	Aplikacija fotografijo razdeli na 10 enako velikih stolpcev.	Iz stolpcev z lihimi oznakami sestavi novo sliko.	Nato to sliko razdeli na 10 enako velikih vrstic.	Iz vrstic z lihimi oznakami naredi novo sliko. To sliko prejme Katarina.

Katarina ji odgovori s sliko polža.



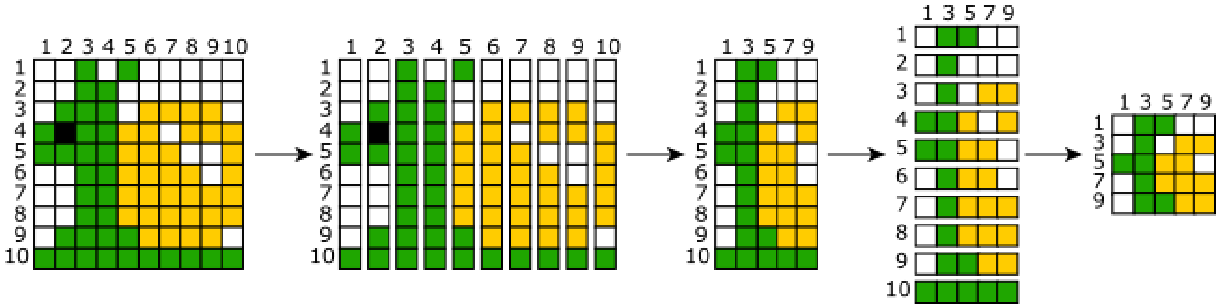
Kakšno fotografijo bo prejela Mojca? Obkroži.

A)  B)  C)  D) 

Rešitev

Pravilni odgovor je D.

Ko sledimo opisanemu postopku, dobimo:



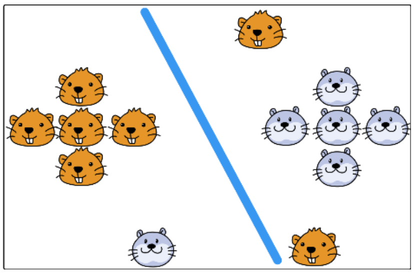
Računalniško ozadje

V nalogi je predstavljen izmišljen algoritem za izgubno stiskanje podatkov, to je količino podatkov o slikovnih točkah smo zmanjšali tako, da smo shranili le podatke o vsaki drugi vrstici in vsakem drugem stolpcu, podatke o preostalih slikovnih točkah pa smo izbrisali in jih izgubili. Kvaliteta slike po uporabi takega algoritma za stiskanje podatkov ne more nikoli več biti enaka kot je bila pred tem.

Bobri in vidre

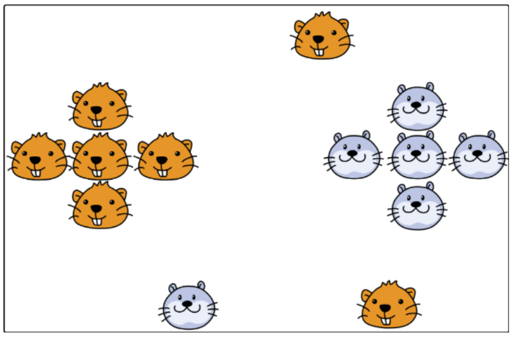
4. in 5. razred

Bobri in vidre so se skregali in zdaj želijo živeti na različnih ozemljih. Ozemlji bodo razdelili z mejo, ki bo potekala po ravni črti. Če pri tem katera žival pristane na napačni strani meje, se mora preseliti. Primer (modra črta je meja):

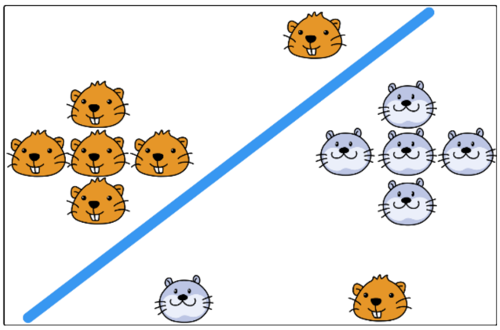


V zgornjem primeru se bodo morale preseliti 3 živali – dva bobra in ena vidra. Ker je selitev za živali zelo stresna, želijo postaviti mejo tako, da se bo selilo čim manj živali.

Z **ravno** črto nariši, kje naj bobri in vidre potegnejo mejno črto.



Rešitev



Pri tej delitvi se bo moral seliti le en bober. Delitve z ravno črto, kjer se ne bi selila nobena žival, ne moremo najti.

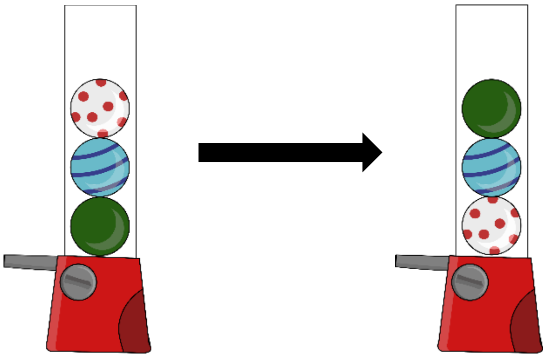
Računalniško ozadje

Tudi osnovnošolci razvrščajo osebe in stvari v različne kategorije, na primer na dekleta in fante, na učitelje in učence, na otroke in odrasle. Takemu razvrščanju v različne kategorije pravimo klasifikacija.

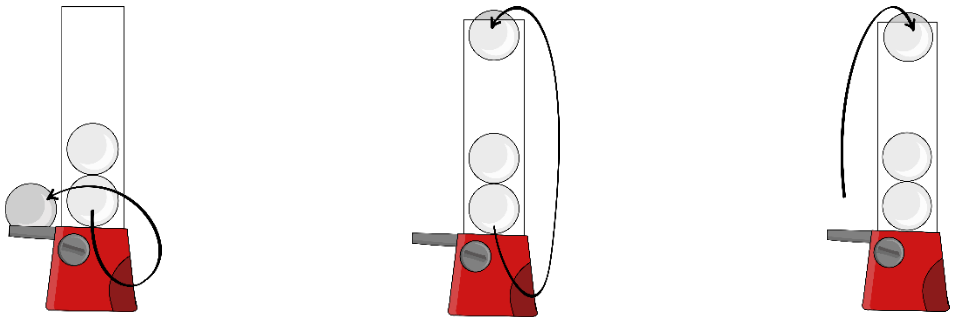
Kroglice

4. in 5. razred

Hinko ima napravo s tremi kroglicami, ki jim želi spremeniti vrstni red, kot kaže slika:



Pri tem lahko uporabi spodnje načine premikanja kroglic:



Spodnjo kroglico postavi na **prazno** poličko.

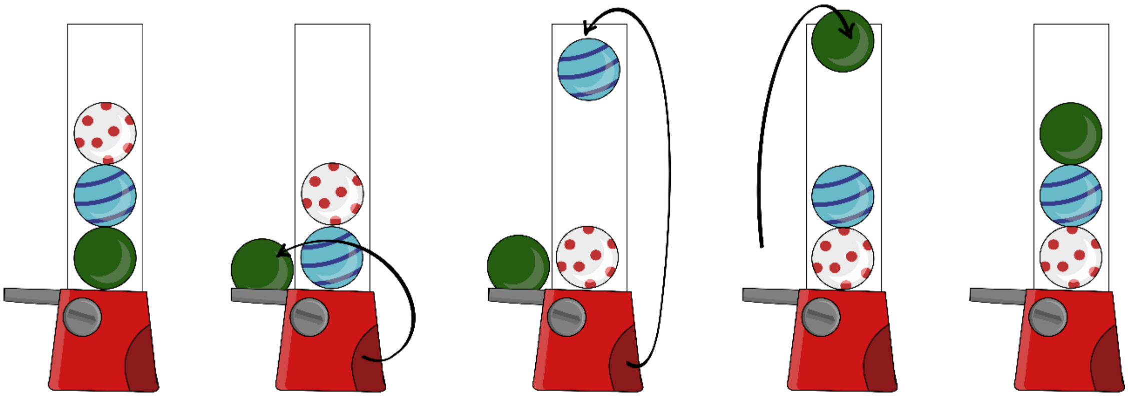
Spodnjo kroglico prestavi na vrh.

Kroglico s poličke postavi na vrh.

Najmanj koliko premikov žogic bo moral narediti?

Rešitev

Narediti bo moral najmanj 3 premike, kot kaže spodnja slika:



Ker lahko kroglice jemlje samo od spodaj, bo zagotovo moral narediti vsaj dve potezi, da modro in zeleno kroglico spravi na vrh. Ker pa imajo v končnem vrstnem redu žogice tudi drugačno zaporedje, kot v prvotnem, mora z eno dodatno potezo zeleno žogico odložiti na poličko. Torej ne moremo dobiti končnega razporeda z manj kot tremi potezami.

Računalniško ozadje

V tej nalogi smo se srečali s tem, kar v računalništvu imenujemo vrsta. Vrsta je namenjena shranjevanju podatkov, pri tem pa uporablja le dva ukaza: dodaj (angleško add), ki shrani podatek na konec vrste, in vzemi (angleško pop), ki vzame podatek z začetka vrste. Vrsta v računalništvu v resnici deluje podobno kot vrsta v trgovini – najprej vrsto zapusti tisti, ki je vanje prišel najprej, ko pride v vrsto nova oseba, pa se postavi na konec vrste.

Magična jablana

4. do 7. razred

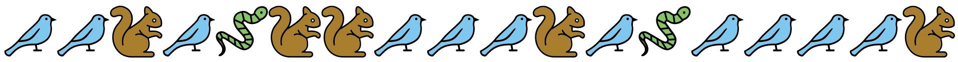
Bober Bine ima na vrtu magično jablano.

Ko na jablano prileti ptica (), na jablani zrasteta dve jabolki.

Ko na jablano spleza veverica (), z jablane pade eno jabolko.

Če pa na jablano spleza kača (), v trenutku z jablane izginejo vsa jabolka.

Neko jutro je Bine na jablani preštel 25 jabolk. Nato je cel dan opazoval in risal, katere živali so prišle na jablano:

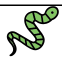
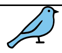
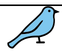
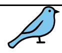
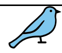


Koliko jabolk je na jablani ob koncu dneva?

Rešitev

Ob koncu dneva je na jablani 7 jabolk.

Ker z jablane izginejo vsa jabolka, ko nanj spleza kača, je dovolj, če pogledamo zaporedje živali po tem, ko na jablano spleza zadnja kača:

Žival						
Kaj se zgodi?	Izginejo vsa jabolka	Zrasteta dve jabolki	Zrasteta dve jabolki	Zrasteta dve jabolki	Zrasteta dve jabolki	Pade eno jabolko
Število jabolk na jablani	0	0 + 2 = 2	2 + 2 = 4	4 + 2 = 6	6 + 2 = 8	8 - 1 = 7

Računalniško ozadje

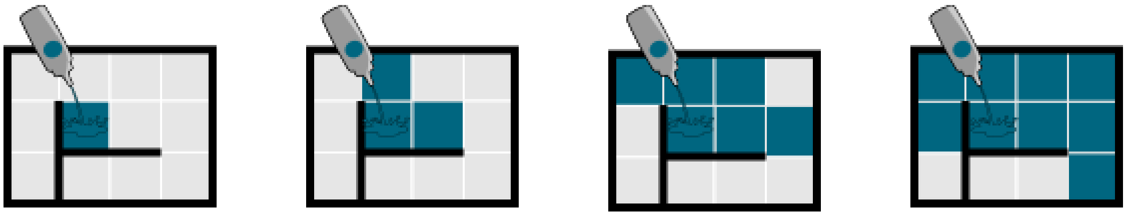
Zapis z živalmi v naši nalogi predstavljajo preprost zapis programa s tremi različnimi ukazi: prištej dve, odštej ena in nastavi vrednost na nič.

Vodenke 1

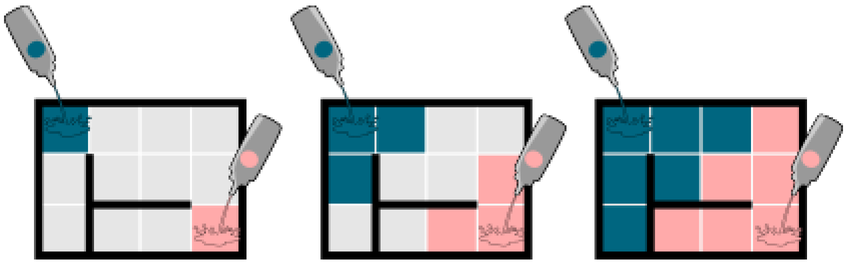
5. razred

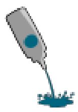
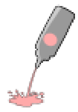
Kot že vemo, so naši bobri zelo vsestranski. Zelo radi tudi ustvarjajo, najraje z vodenkami. Razvili so tehniko ustvarjanja labirintov. Z voščenkami si narišejo labirint, nato pa na eno polje kar iz stekleničke kapnejo vodeno barvo. Ta se potem samodejno razliva po poljih na papirju, vsako sekundo na vsa sosedna polja, ki še niso pobarvana.

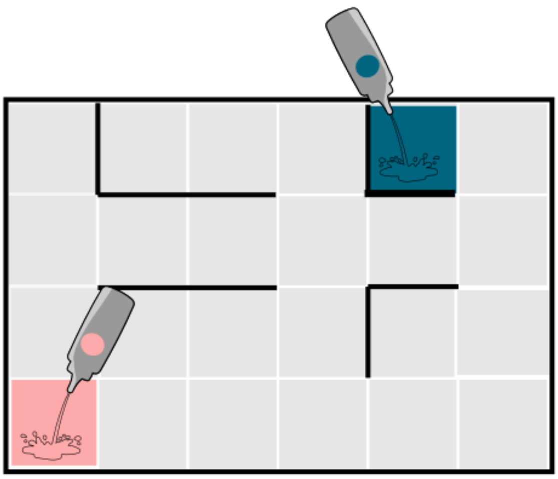
Primer:



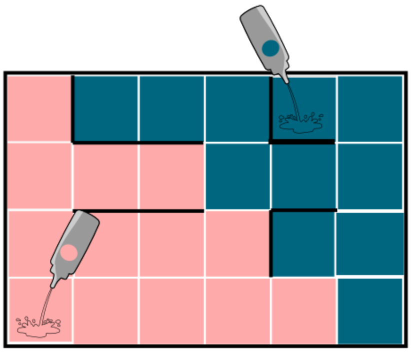
Če bober uporabi več barv, se vsako polje obarva z barvo, ki prva pride do tega polja. Če bi dve barvi **hkrati prišli** do nekega polja, se polje obarva s **temnejšo** barvo.



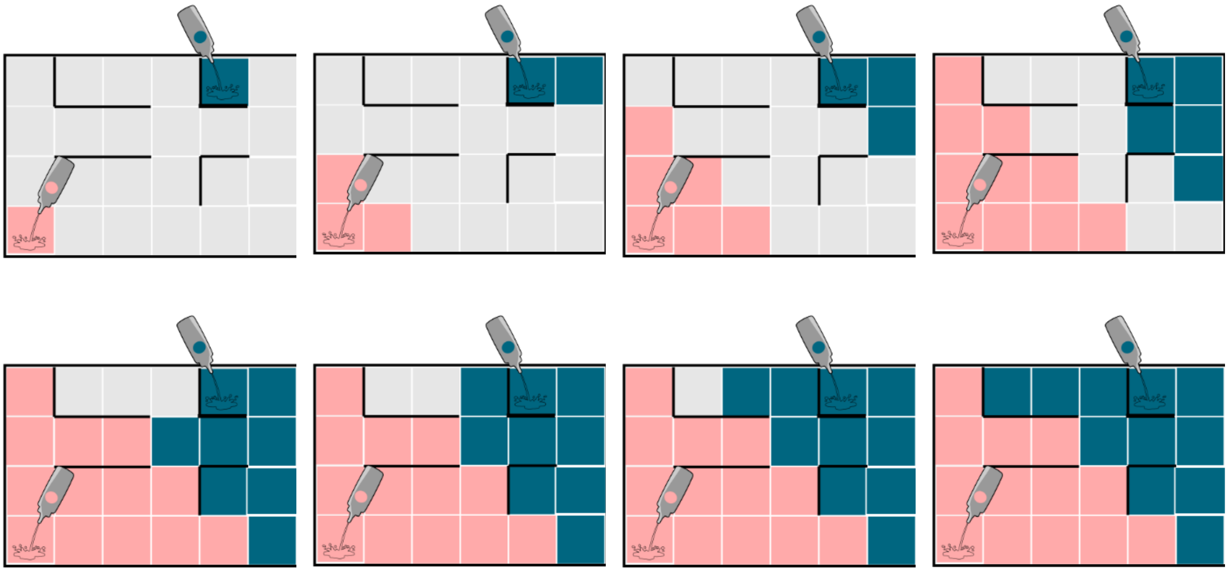
Bober je narisal spodnji labirint in nanj kapnil modro  in roza  barvo, kot kaže slika. Pobarvaj, kako izgleda slika, ko se barve razlijejo čez cel labirint.



Rešitev



To sliko dobimo s postopkom:



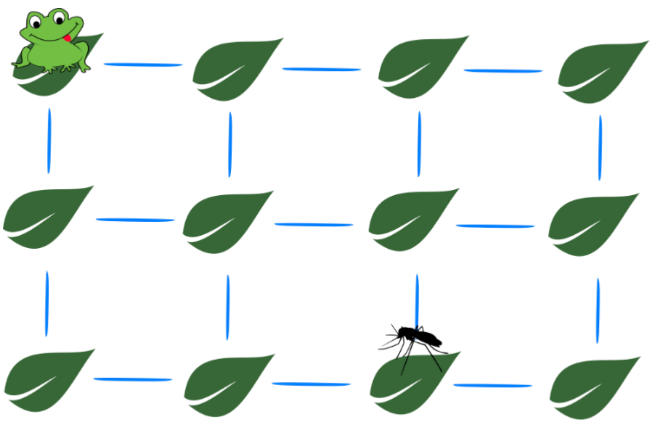
Računalniško ozadje

Pri tej nalogi smo postopoma »poplavljali« polja na mreži z ovirami, pri tem pa šteli, koliko korakov potrebujemo, da posamezna barva doseže posamezno polje. Tista barva, ki je prej dosegla posamezno polje, je na polju tudi ostala.

Žaba in komar

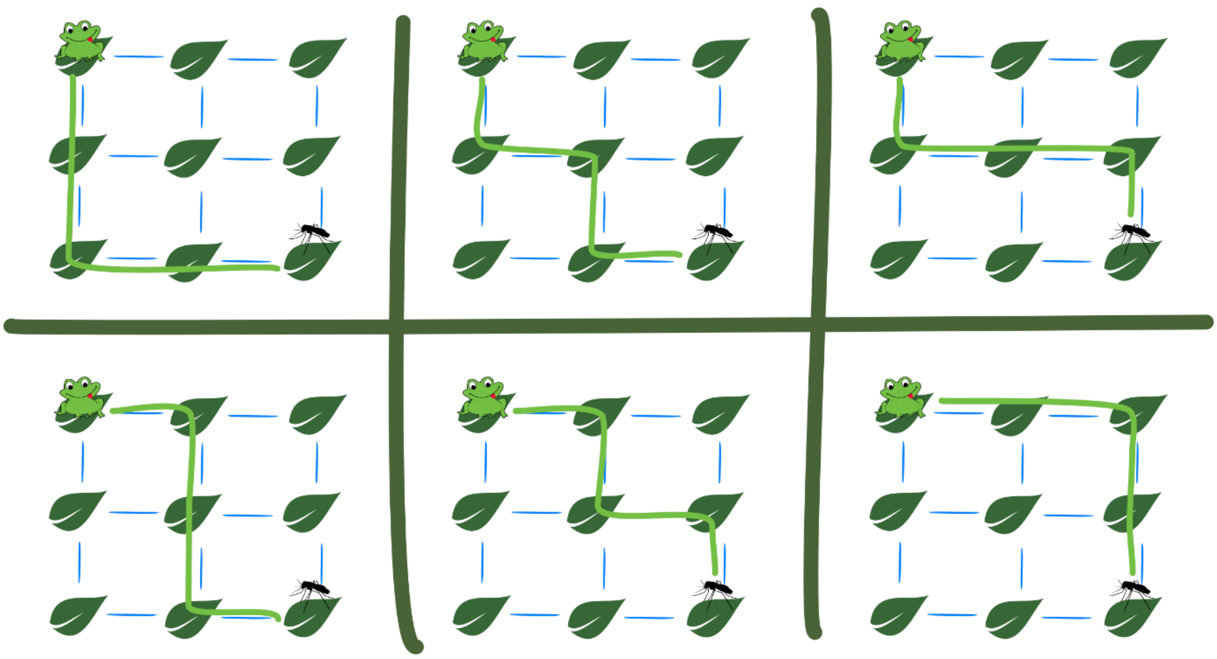
5. do 7. razred

Žaba skače z lokvanja na lokvanj po modrih črtah. Koliko različnih poti jo vodi do komarja, če bo skočila natanko štirikrat?



Rešitev

Možnih je 6 poti.



Računalniško ozadje

Pri tej nalogi smo iskali poti na grafu, ki žabo od začetnega vozlišča do vozlišča s komarjem pripeljejo v točno štirih korakih. Tudi v računalništvu pogosto iščemo poti v grafih.

Božičkovi pomočniki

5. do 7. razred

Bobri letos pomagajo Božičku izdelovati darila. Vsak bober bo izdelal natanko eno igračo v napravi za izdelavo igrač.

Pri tem bobri upoštevajo naslednji časovni razpored:

Bober	Igrača	Trajanje izdelave	Bobrov čas prihoda
Andrej	Avto	5 minut	8.00
Bojan	Medvedek	10 minut	8.00
Cene	Punčka	7 minut	8.04
Dejan	Vlak	12 minut	8.10
Enej	Kocke	9 minut	8.10

Ko bober vstopi v delavnico, gre najprej v čakalnico. Napravo lahko uporablja le en bober hkrati, ostali bobri pa ostanejo v čakalnici. Ko je naprava prosta, jo prične uporabljati bober, ki je že v čakalnici in za izdelavo igrače potrebuje najmanj časa. Ko je bober na vrsti, z napravo izdela igračo od začetka do konca.

V katerem vrstnem redu bodo bobri izdelali svoje igrače?

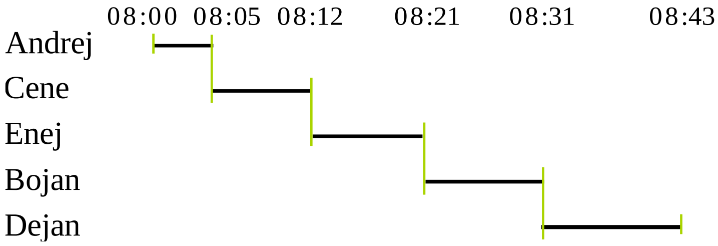
A) Andrej, Bojan, Cene, Dejan, Enej
B) Andrej, Cene, Enej, Bojan, Dejan
C) Andrej, Bojan, Cene, Enej, Dejan
D) Andrej, Cene, Bojan, Enej, Dejan

Rešitev

Pravilen odgovor je B: Andrej, Cene, Enej, Bojan, Dejan.

Ob 8.00 prideta v čakalnico Andrej in Bojan. Ker bo Andrej izdelal svojo igračo hitreje (5 minut) kot Bojan (10 minut), je prvi na vrsti Andrej. Ob 8.04 pride v čakalnico še Cene. Ob 8.05 Andrej konča z izdelavo igrače. Ker izdelava Cenetove igrače traja manj časa (7 minut) kot Bojanove (10 minut), je drugi na vrsti Cene. Ob 8.10 se Bojanu v čakalnici pridružita še Dejan in Enej. Cene ob 8.12 zaključi z izdelavo svoje igrače. Med čakajočimi bo za izdelavo igrače najmanj časa potreboval Enej (9 minut), zato je na vrsti on, Bojan (10 minut) in Dejan (12 minut) pa počakata na vrsto. Ob 8.21 zaključi z izdelovanjem igrače Enej. Na vrsti je Bojan, ki igračo izdela ob 8.31, nato pa še Dejan, ki igračo izdela ob 8.43.

Kdo kdaj izdeluje svojo igračo, je prikazano na naslednji sliki.



Ostali odgovori niso pravilni, saj ne upoštevajo, kateri bober v čakalnici potrebuje najmanj časa za izdelavo svoje igrače.

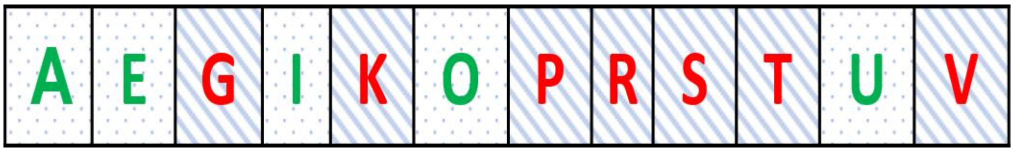
Računalniško ozadje

Naloga ilustrira delovanje algoritma najkrajši naprej, ki se v operacijskih sistemih uporablja za razvrščanje procesov. Za razvrščanje procesov so na voljo tudi drugi algoritmi, računalnik pa v resnici uporablja kombinacijo teh.

Parkirne dovolilnice 1

5. do 7. razred

Bobrsko ministrstvo za promet uporablja posebno abecedo za označevanje parkirnih dovolilnic:



Samoglasniki so postavljeni na pikčastem ozadju (zelene črke), soglasniki so postavljeni na črtastem ozadju (rdeče črke).

Oznake dovolilnic, ki jih tiskajo, ustrezajo sledečim pravilom:

- sestavljajo jih natanko tri črke, ki jim sledi ena števka,
- prva črka je soglasnik,
- tretja črka je samoglasnik,
- črke se na dovolilnici lahko ponavljajo.

Katera oznaka dovolilnice je veljavna?

A) RAKS7
B) PSAT
C) TAA3
D) V5E6
E) CRY4

Rešitev

Pravilen odgovor je C: TAA3.

TAA3 ustreza vsem pravilom: je sestavljena iz natanko treh črk, ki jim sledi številka; prva črka (T) je soglasnik (rdeča črka s črtastim ozadjem); druga črka (A) je iz abecede (druga črka je lahko katera koli črka abecede); tretja črka (A) samoglasnik (zelena črka na pikčastem ozadju); zadnji znak pa je števka (3).

Ker je ponavljanje črk dovoljeno, ni nobenega problema s ponovitvijo črke A v dovolilnici TAA3.

Ostali odgovori so nepravilni, ker:

A) RAKS7 – oznako sestavljajo štiri črke, dovoljene so le tri;

B) PSAT – četrti znak bi morala biti števka, ne črka;

D) V5E6 – števka je lahko le ena na koncu oznake;

E) CRY4 -črki C in Y nista iz abecede, ki jo uporabljajo za oznake dovolilnic.

Računalniško ozadje

V veliko primerih morajo biti podatki sestavljeni po točno določenih pravilih. En primer so registrske oznake vozil (kot v tej nalogi, četudi Bobrsko ministrstvo za promet postavlja povsem drugačna pravila, kot veljajo v Sloveniji), drug primer pa gesla za prijavo v računalnik (dobro geslo mora, na primer, vsebovati majhne in velike črke, številke in posebne znake ter biti dolgo najmanj 12 znakov). Programi, ki obdelujejo vhodne podatke, morajo tudi preverjati, če podatki ustrezajo podanim pravilom, saj lahko napačni vhodni podatki (ki se ne skladajo s podanimi pravili) povzročijo velike težave. Zamisli si nastalo zmedo, če bančni sistem ne bi preverjal vrednosti plačila in bi lahko na račun prejemnika plačali negativni znesek! To bi dejansko pomenilo, da s prejemnikovega računa denar dvignemo in ga prenesemo na svoj račun.

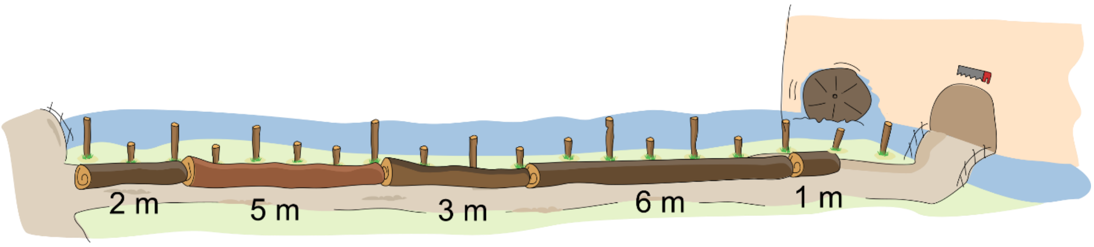
Polena

5. do 9. razred

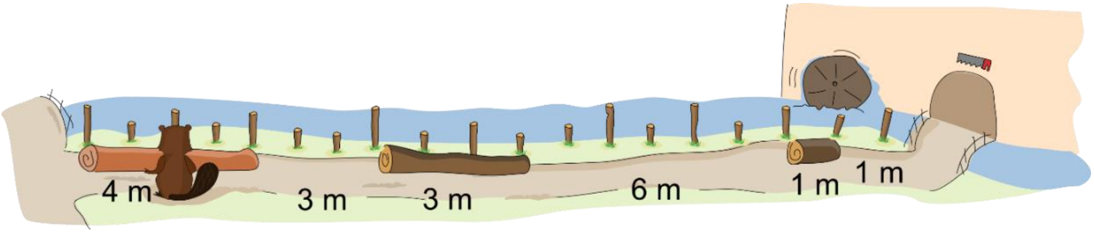
Bober Branko razreže debla na različno dolga polena in jih prodaja ob cesti. Ko odreže novo poleno, ga postavi ob 18-metrsko ozko cesto, kjer so polena zložena eno za drugim, da ne ovirajo prometa.

Vsakič ko Branko pripravi novo poleno, ga vstavi na prvo mesto z leve, na katerem je na voljo dovolj prostora za to poleno. Ko pa proda neko poleno, ga odstrani z mesta in tako se naredi prostor za nova polena.

Branko je ob cesto postavil polena različnih dolžin v naslednjem vrstnem redu: 2 m, 5 m, 3 m, 6 m in 1 m. Njihovo postavitev ob cesti prikazuje slika.

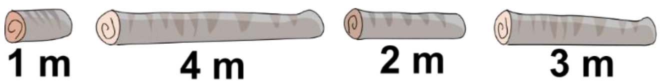


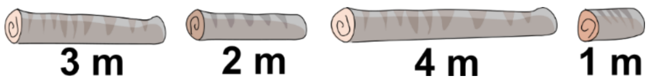
Nato je prodal polena dolžine 6, 2 in 5 metrov. Potem je odrezal novo 4-metrsko poleno in ga postavil skrajno levo. Sedaj cesta izgleda takole:

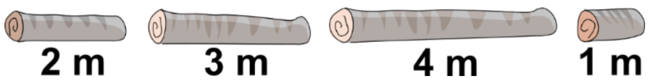


Branko mora odrezati še polena dolžine 1, 2, 3 in 4 metre. Kakšen je vrstni red, v katerem mora odrezati polena, da lahko vsa postavi ob cesto (in pri tem upošteva zgoraj opisana pravila)?

A) 

B) 

C) 

D) 

E) 

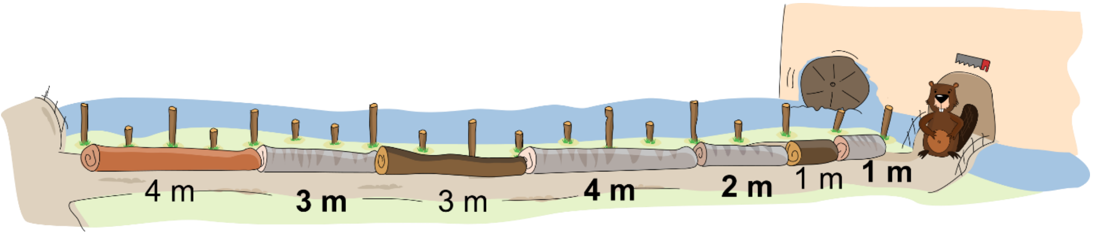
Rešitev

Pravilen odgovor je C. Polena nareže v naslednjem vrstnem redu: 3 m, 2 m, 4 m in 1 m.

Pri vseh ostalih odgovorih Branko nima prostora, da ob cesto postavi zadnje poleno (oziroma pri odgovoru D zmanjka prostor za predzadnje poleno).

Na sliki stanja polen ob cesti, preden Branko razreže nova polena, vidimo, da imamo tri prosta mesta: najbolj levo je dolgo 3 metre, srednje mesto je dolgo 6 metrov in zadnje, na skrajni desni, je dolgo 1 meter.

Odgovor C popolnoma zapolni levo prazno mesto s prvim polenom. Z naslednjima dvema polenoma zapolni v celoti še srednje prosto mesto, zadnje poleno pa gre v skrajno desno prosto mesto. Tako lahko spravimo vsa polena.



Pri odgovoru A popolnoma zapolnimo levo prazno mesto s prvima dvema polenoma, nato pa moramo 3-metrsko poleno postaviti na srednje prazno mesto. S tem smo zmanjšali razpoložljiv prostor srednjega mesta na 3 metre. Zadnje poleno, ki je dolgo 4 metre, tako ne paše niti na to mesto niti na zadnje, skrajno desno prazno mesto.

Odgovor B postavi 1-metrsko poleno na skrajno levo mesto in naslednje, 4-metrsko poleno na srednje mesto. Levo mesto je le delno zapolnjeno in ostane 2-metrski prostor, ki ga popolnoma zapolnimo s tretjim polenom. Vendar za zadnje, 3-metrsko poleno zmanjka prostora (na voljo imamo le 2 metra na srednjem mestu in 1 meter na zadnjem mestu).

Podoben problem imamo tudi pri odgovoru D, le da nanj naletimo že prej, saj nimamo prostora za tretje, 4-metrsko poleno. Zanj sta levo in desno mesto tako ali tako premajhni, srednje mesto pa je delno že zasedeno z drugim, 3-metrskim polenom.

Odgovor E postavi 4-metrsko poleno na srednje mesto (tam ostane še 2 metra prostora), nato s 3-metrskim polenom popolnoma zapolni levo prosto mesto. Tretje, metrsko poleno, postavi na srednje mesto (in tam ostane še meter prostora). Zadnjega polena, ki je dolgo 2 metra, pa ne moremo nikamor postaviti, saj je na edinih prostih mestih, srednjem in zadnjem, le po en meter prostora.

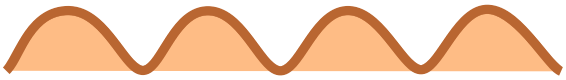
Računalniško ozadje

Cesta, ki jo uporablja Brane, je podobna pomnilniku RAM (angl. *random access memory*) v računalniku, polena pa so kot računalniški procesi, ki za delovanje zahtevajo določeno velikost prostora na pomnilniku. Čeprav je v primeru v nalogi dovolj prostora za vsa polena, je vrstni red dodajanja in odvzemanja lahko tak, da zmanjka prostora za nova polena. Podobno se lahko zgodi tudi v računalniku: vrstni red, v katerem se deli pomnilnika zasedajo in sproščajo, lahko privede do tega, da za proces določene velikosti ni na voljo dovolj velikega strnjenega prostora na pomnilniku, čeprav je skupni prosti del pomnilnika večji od velikosti tega procesa. Temu pravimo fragmentacija pomnilnika. Na podobne probleme naletimo tudi pri zapisu datotek na disk.

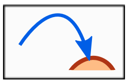
Sejanje korenja

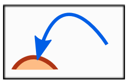
5. do 9. razred

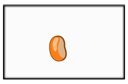
Robotski zajec seje korenje na štirih gomilah.



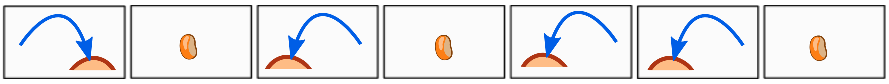
Zajec pozna naslednje ukaze:

 Skoči na desno gomilo.

 Skoči na levo gomilo.

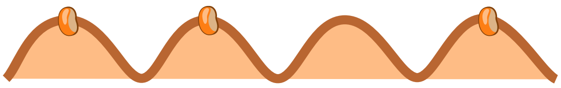
 Posej seme korenja na gomili, na kateri si.

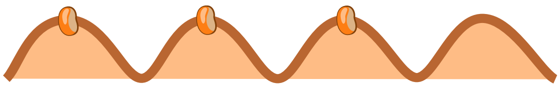
Zajcu smo pripravili naslednje zaporedje ukazov:

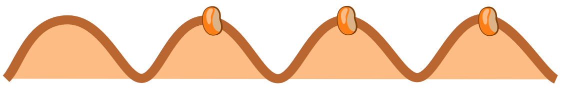


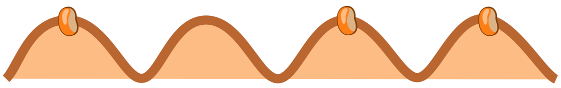
Ne vemo, na kateri gomili je bil zajec na začetku, vemo pa, da je na koncu posejal semena korenja na treh različnih gomilah.

Katera slika pravilno prikazuje, na katerih gomilah je zajec posejal semena korenja?

A) 

B) 

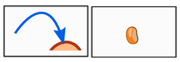
C) 

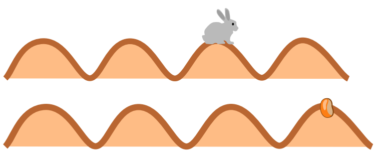
D) 

Rešitev

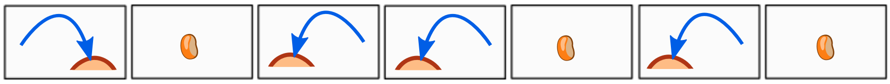
Pravilen odgovor je D.

Zajec mora začeti na tretji gomili, saj bi drugače moral skočiti izven slike.

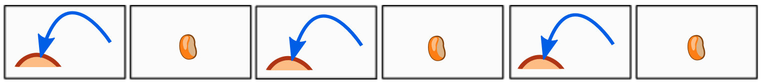




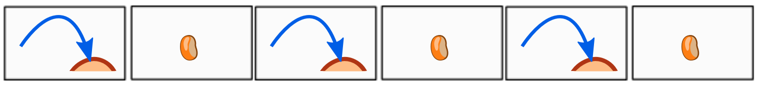
Da bi dobili odgovor A, bi lahko zajec na začetku stal na tretji gomili in izvedel naslednje zaporedje ukazov:



Da bi dobili odgovor B, bi lahko zajec na začetku stal na četrti gomili in izvedel naslednje zaporedje ukazov:



Da bi dobili odgovor C, bi zajec lahko stal na prvi gomili in izvedel naslednje zaporedje ukazov:



Računalniško ozadje

Robote upravljamo s programi. Program je zaporedje ukazov, ki jih robot razume. Pomemben del pisanja programov je tudi razhroščevanje, to je iskanje napak v programski kodi.

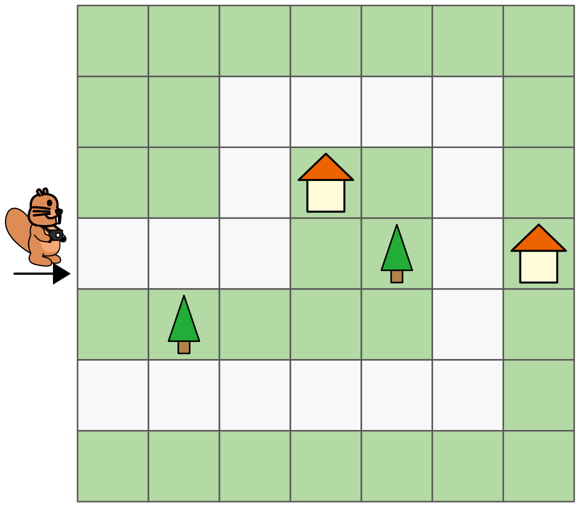
Slike s poti 2

5. do 9. razred, srednja šola

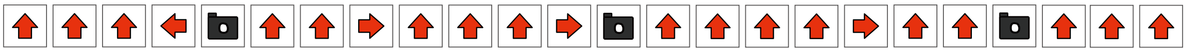
Bober je šel na sprehod in po poti posnel tri fotografije v smeri hoje.

Za opis poti sprehoda je bober uporabil naslednje znake:

Premakni se naprej	
Obrni se levo	
Obrni se desno	
Posnemi fotografijo	

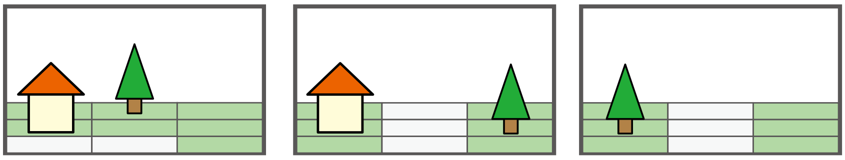


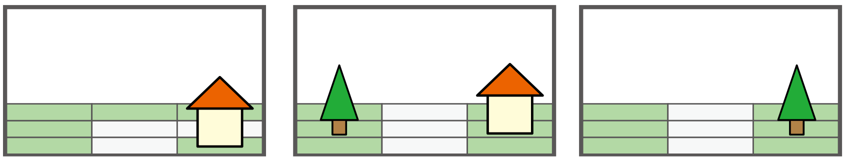
Svojo pot je opisal tako:

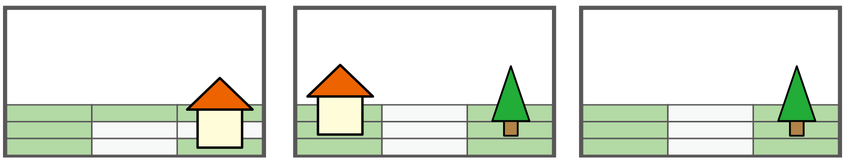


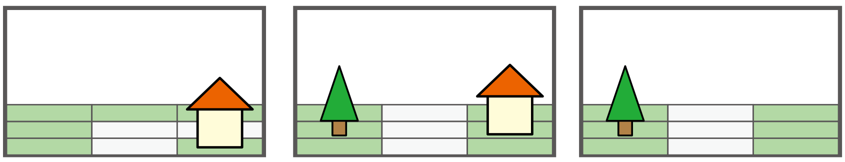
Vsaka fotografija prikazuje, kar bober vidi pred seboj na naslednjih 3 poljih in kar je levo in desno od njih.

Katere tri fotografije je bober posnel na sprehodu?

A) 

B) 

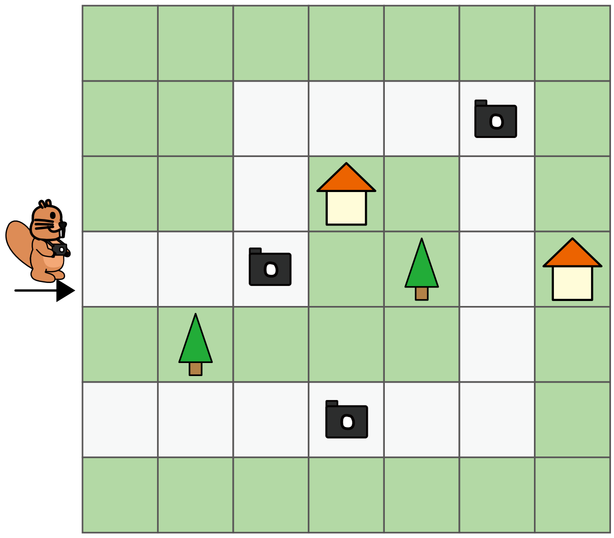
C) 

D) 

Rešitev

Pravilen odgovor je C.

Bober je svoje fotografije posnel z naslednjih treh mest:



Fotografij pod A in D ni mogel posneti, saj je na zadnji fotografiji drevo na levi strani ceste, moralo pa bi biti na desni.

Fotografij pod B ni mogel posneti, saj sta na drugi fotografiji drevo in hiša na napačni strani glede na mesto, s katerega je bober fotografijo posnel. (Enaka, napačna fotografija je tudi druga fotografija pod D.)

Torej je posnel fotografije pod C. Preverimo še, ali je prava tudi prva fotografija pod C: na desni je hiša, na sredini cesta in na levi travnik, na zadnjih poljih pa cesta zavije in vsa tri polja zaseda travnik.

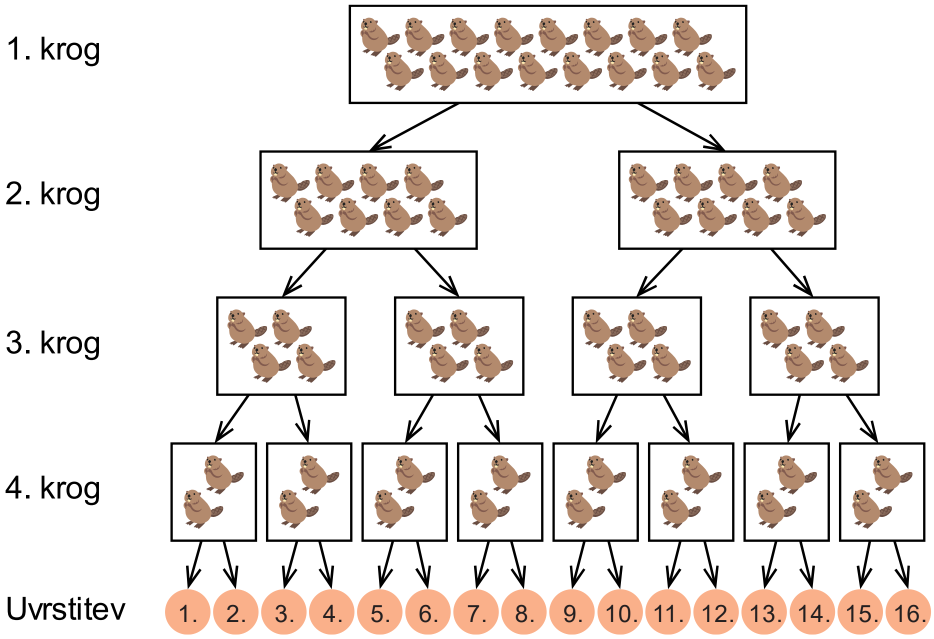
Računalniško ozadje

Računalniški vid je pomemben del računalništva, saj nam omogoča, da se z njegovo pomočjo roboti orientirajo po prostoru.

Squash turnir

6. in 7. razred

Danes imajo bobri vsakoletni squash turnir. Na tekmovanje se je uvrstilo 16 tekmovalcev. Tekmovali bodo v štirih krogih, da bi ugotovili, kdo je letos najboljši igralec squasha. Igralci bodo po odigranih štirih krogih razvrščeni od 1. do 16. mesta. Vseh 16 igralcev skupaj igra v prvem krogu, potem pa se po vsakem krogu razdelijo na dva dela. Zmagovalci nadaljujejo v eni skupini (na sliki označena s puščico levo), poraženci pa v drugi skupini (na sliki označena s puščico desno) v naslednji krog tekmovanja.



Na primer, igralec, ki zmaga v 1. in 2. krogu, ampak izgubi v 3. in 4. krogu, se bo uvrstil na 4. mesto.

Noro je igral na turnirju in izgubil samo enkrat. Na katero mesto se je lahko uvrstil?

Rešitev

Noro je lahko zasedel 2., 3., 5. ali 9. mesto.

Če je Noro izgubil v 1. krogu, je nato zmagal v 2., 3. in 4. krogu, torej je zasedel 9. mesto.

Če je izgubil v drugem krogu in zmagal v 1., 3. in 4., je zasedel 5. mesto.

Če je zmagal v 1., 2. in 4., izgubil pa v 3. krogu, je zasedel 3. mesto.

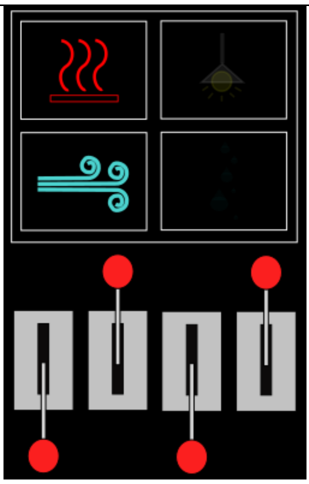
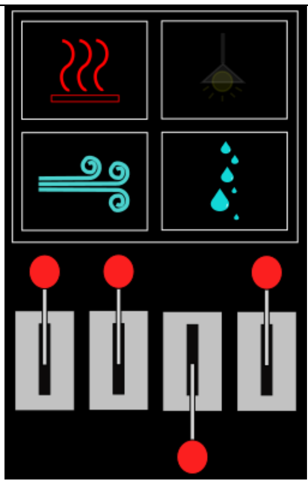
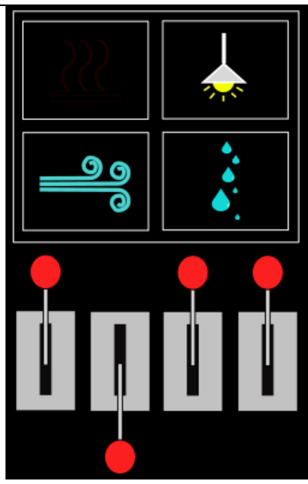
Če pa je zmagoval vse do 4. kroga in v njem izgubil, pa je zasedel 2. mesto.

Stikala

6. in 7. razred

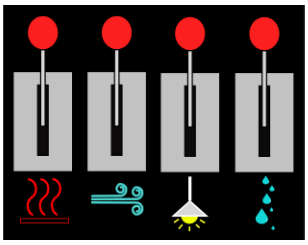
Na neki vesoljski postaji ima vsako območje enako nadzorno ploščo s 4 stikali za nadzor sistemov za ogrevanje (), prezračevanje (), osvetlitev () in vlažnost(). Vsa stikala na vsaki plošči delujejo na enak način in so razporejena v istem vrstnem redu. Ker je postaja zelo stara, so se z leti oznake na teh stikalih žal izbrisale.

Na srečo v 3 območjih sistemi še vedno delujejo. Na spodnji sliki so prikazane plošče v teh območjih. Nad stikali so razporejene sličice, ki kažejo, kateri sistemi so v posameznem primeru vklopljeni.

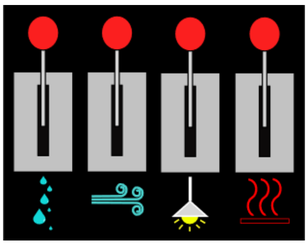
1	2	3
		

Glede na zgornje kontrolne plošče ugotovi, kateri sistem upravlja katero stikalo (stikala so postavljena v enakem vrstnem redu kot na zgornjih nadzornih ploščah).

A)

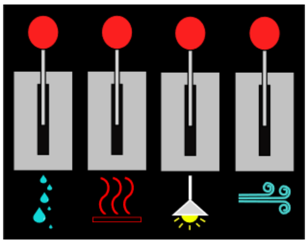


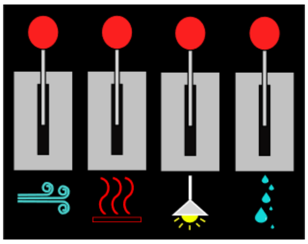
B)



C)

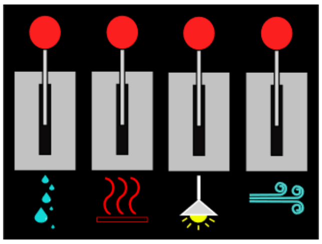
D)





Rešitev

Pravilna rešitev je C:



Oglejmo si stikala.

Ker je na vseh treh kontrolnih ploščah vklopljeno prezračevanje, četrto stikalo pa je edino, ki je vklopljeno na vseh treh kontrolnih ploščah, mora biti četrto stikalo za prezračevanje. To upošteva le stikalo v odgovoru C.

Lahko se prepričamo tudi drugače. Prvo stikalo je na plošči 1 obrnjeno navzdol, na 2 in 3 pa navzgor. Edini sistem, ki ima v prvem primeru različno stanje od drugih dveh, je vlaga. Torej je prvo stikalo za vlažnost.

Zato sta drugo in četrto stikalo zagotovo namenjena ogrevanju in prezračevanju (saj sta na kontrolni plošči 1 vključeni), tretje pa torej pripada osvetljavi. Na kontrolni plošči 3 je drugo stikalo izklopljeno, prav tako pa tudi ogrevanje. Zato vemo, da z drugim stikalom upravljamo ogrevanje.

Računalniško ozadje

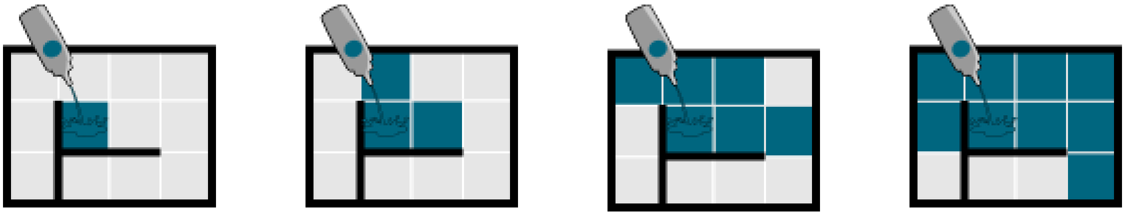
Vsako stikalo na kontrolni plošči je lahko prižgano ali ugasnjeno, tako kot naprava, ki je vezana na to stikalo. Iz slik, ki prikazujejo, katera naprava trenutno deluje, moramo logično sklepati na to, katero napravo upravljamo s posameznim stikalom.

Vodenke 2

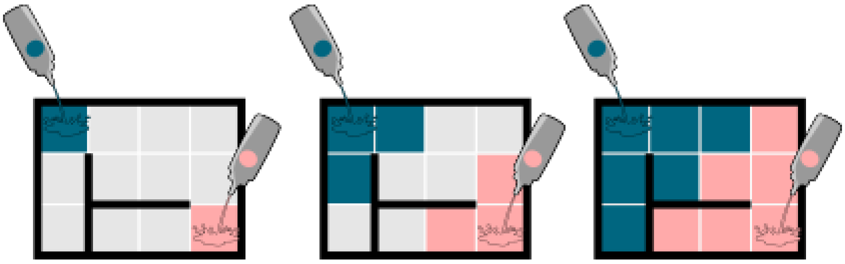
6. do 9. razred

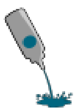
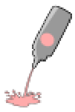
Kot že vemo, so naši bobri zelo vsestranski. Zelo radi tudi ustvarjajo, najraje z vodenkami. Razvili so tehniko ustvarjanja labirintov. Z voščenkami si narišejo labirint, nato pa na eno polje kar iz stekleničke kapnejo vodeno barvo. Ta se potem samodejno razliva po poljih na papirju, **vsako sekundo** na vsa sosedna polja, ki še niso pobarvana.

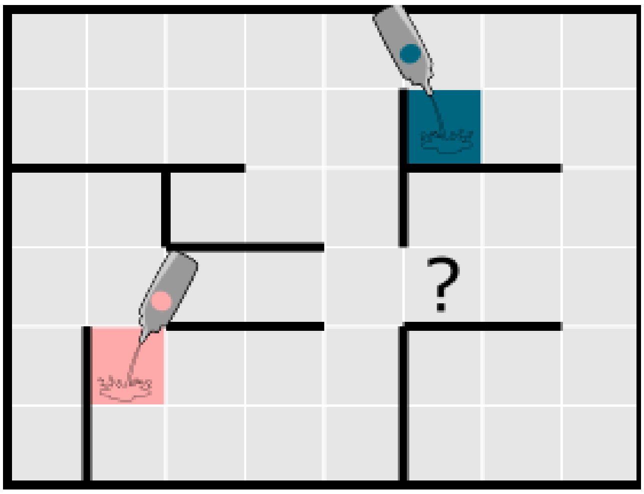
Primer:



Če bober uporabi več barv, se vsako polje obarva z barvo, ki prva pride do tega polja. Če bi dve barvi **hkrati** prišli do nekega polja, se polje obarva s **temnejšo** barvo.



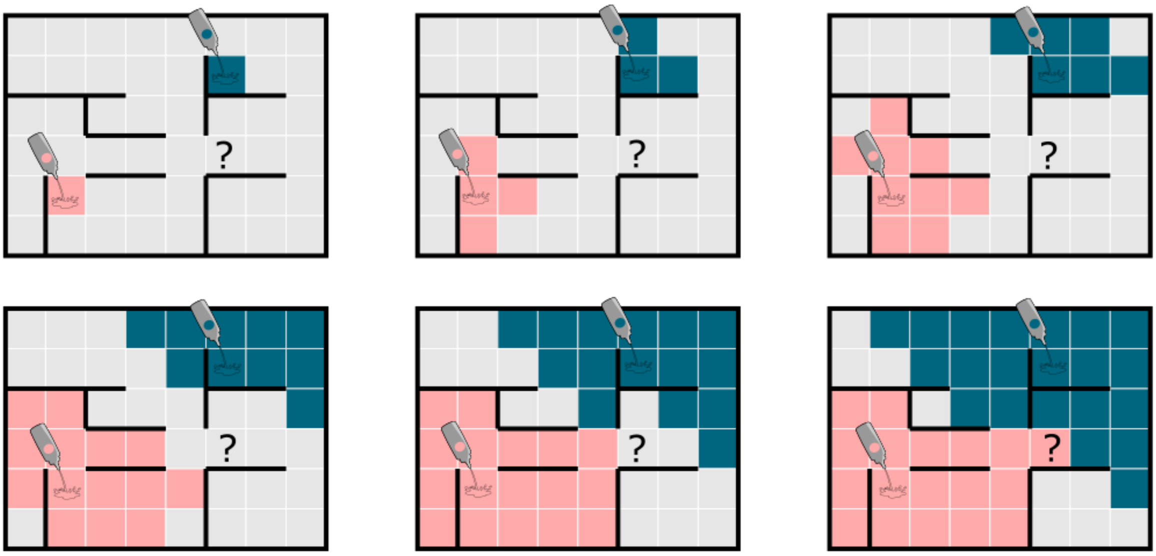
Bober je narisal spodnji labirint in nanj kapnil modro  in roza  barvo, kot kaže slika. S katero barvo se obarva označeno polje?



Rešitev

Označeno polje je na koncu roza barve.

Dovolj je, da si ogledamo potek barvanja do točke, ko se obarva označeno polje:



Računalniško ozadje

Pri tej nalogi smo postopoma »poplavljali« polja na mreži z ovirami, pri tem pa šteli, koliko korakov potrebujemo, da posamezna barva doseže posamezno polje. Tista barva, ki je prej dosegla posamezno polje, je na polju tudi ostala.

Sporočila po reki

6. do 9. razred

Bobri iz vasi ob reki pošiljajo sporočila bobrom v vasi nižje ob reki tako, da na lesene ploščke zapišejo enomestna števila in jih spustijo po reki do druge vasi.

Včasih se zgodi, da rečni tok obrne katerega od ploščkov, pri čemer se številka na njem izbriše, a vrstni red ploščkov se vedno ohrani. Zato na koncu sporočila dodajo še en plošček, ki vsebuje tako imenovano »kontrolno števko«. Ta se izračuna tako, da vsako drugo vrednost pomnožijo s 3, začenši s prvim ploščkom, nato pa seštejejo vse vrednosti. Na koncu vzamejo zadnjo števko rezultata in jo kot kontrolno števko zapišejo na zadnji plošček.

Za sporočilo na sliki (8 4 2 5) bi tako naračunali kontrolno števko na naslednji način (števila na ploščkih so zapisana krepko):

Plošček 1	Plošček 2	Plošček 3	Plošček 4	Vsota
(**8**×3)	+ **4**	+ (**2**×3)	+ **5**	= 39

Vrednosti na prvem in tretjem lesenem ploščku smo pomnožili s 3, rezultata sešteli in prišteli še vrednosti na drugem in četrtem ploščku. Vsota vseh vrednosti je 39, vzamemo le zadnjo števko, to je 9, in jo kot kontrolno števko zapišemo na zadnji plošček.

Ko so bobri poslali spodnje sporočilo po reki, se je tretji leseni plošček obrnil. Katera številka je zapisana na njem?

Rešitev

Na tretjem lesenem ploščku je zapisano število 6.

Vemo, da je kontrolna števka enaka 8. Torej mora biti vsota $4 \times 3 + 1 + x \times 3 + 7$ enaka številu, katerega enic je 8 (z x smo označili iskano število na tretjem ploščku). To število lahko poiščemo tako, da najprej izračunamo $4 \times 3 + 1 + 7 = 20$. Ker ima ta vsota zadnjo števko (enice) 0, mora biti $x \times 3$ enako nekemu številu, katerega zadnja števka (enice) je 8. Z drugimi besedami, iščemo enomestno število, katerega trikratnik ima zadnjo števko enako 8. Temu pogoju ustreza le število 6. To lahko preverimo tudi s pomočjo tabele trikratnikov vseh števil od 0 do 9 (podčrtana je zadnja števka, to so enice):

$0 \times 3 = \underline{0}$	$1 \times 3 = \underline{3}$	$2 \times 3 = \underline{6}$	$3 \times 3 = \underline{9}$	$4 \times 3 = 1\underline{2}$
$5 \times 3 = 1\underline{5}$	$6 \times 3 = 1\underline{8}$	$7 \times 3 = 2\underline{1}$	$8 \times 3 = 2\underline{4}$	$9 \times 3 = 2\underline{7}$

Torej so bobri poslali sporočilo 41678. Preverimo lahko še izračun kontrolne števke: $4 \times 3 + 1 + 6 \times 3 + 7 = 3\underline{8}$.

Računalniško ozadje

Kontrolne števke (oziroma nekoliko bolj zapletene *kontrolne vsote*) se pogosto uporabljajo za ugotavljanje napak pri prenosu podatkov. To je zelo pomembno, saj se podatki večinoma prenašajo na velike razdalje (npr. ko odprete spletno stran v tuji državi) ali z nezanesljivo kakovostjo prenosa (npr. brezžično preko zraka).

Najbolj enostavna metoda je, da enostavno seštejemo vse številke. Vendar s to metodo ne moremo ugotoviti zamenjave dveh sosednjih števil, kar je pogosta napaka pri prenosu. Z enostavnim dodatkom, izmeničnim množenjem z dvema različnima številoma (v našem primeru 3 in 1), lahko v večini primerov ugotovimo tudi tako napako pri prenosu (le razlika obeh izbranih števil ne sme biti 5).

Tekmovalci

6. do 9. razred

Hrastova in Kostanjeva ekipa se bosta pomerili na prijateljski tekmi. V obeh ekipah je po 5 članov, za tekmo pa morajo izbrati po 4 igralce. Da bo igra poštena, bodo izbrali igralce tako, da bo vsota starosti vseh igralcev enaka v obeh ekipah.

HRASTOVA EKIPA:

STAROST 15 21 7 8 12

KOSTANJEVA EKIPA:

STAROST 20 9 6 18 11

Kateri član Kostanjeve ekipe ne bo igral na tekmi?

A)

20

B)

9

C)

6

D)

18

E)

11

Rešitev

Pravilen odgovor je B. Na tekmi ne bo igral 9-letni član ekipe.

Edina možna rešitev je, da izberemo naslednje igralce:

HRASTOVA EKIPA:

15 21 7

12

KOSTANJEVA EKIPA:

20

6 18 11

Tako imata obe ekipi enako vsoto let igralcev: 15 + 21 + 7 + 12 = 20 + 6 + 18 + 11 = 55.

En način, kako pridemo do rešitve, je s poskušanjem. Lahko pa rešitev poiščemo tudi bolj pametno. Za vsako ekipo izračunamo vsoto let vseh članov ekipe: dobimo 15 + 21 + 7 + 8 + 12 = 63 za Hrastovo ekipo ter 20 + 9 + 6 + 18 + 11 = 64 za Kostanjevo ekipo. Vidimo, da sta vsoti starosti precej podobni, razlikujeta se le za 1. Sedaj poiščemo par igralcev, prvi iz Hrastove, drugi iz Kostanjeve ekipe, pri katerem je prvi igralec eno leto mlajši od drugega igralca v paru. Hitro lahko vidimo, da edino 8-letni član Hrastove ekipe in 9-letni član Kostanjeve ekipe zadoščata temu pogoju. Če ta dva ne bosta igrala, imata obe ekipi enako vsoto starosti igralcev: 63 – 8 = 55 (Hrastova ekipa) in 64 – 9 = 55 (Kostanjeva ekipa).

Vendar pozor, na tak način ne moremo rešiti poljubnega problema vsote podmnožic. Rešitev poljubnega problema lahko zagotovo poiščemo, če preverimo vse možnosti. V naši nalogi bi za vsako ekipo sistematično sestavili vse možne kombinacije igralcev in izračunali vsoto njihovih let, kot prikazuje tabela:

Št.	*Ekipa*	*Kombinacija*	*Vsota let*
1	Hrastova	{15, 21, 7, 8}	51
2	**Hrastova**	**{15, 21, 7, 12}**	**55**
3	Hrastova	{15, 21, 8, 12}	56
4	Hrastova	{15, 7, 8, 12}	42
5	Hrastova	{21,7, 8, 12}	48
6	Kostanjeva	{20, 9, 6, 18}	53
7	Kostanjeva	{20, 9, 6, 11}	46
8	Kostanjeva	{20, 9, 18, 11}	58
9	**Kostanjeva**	**{20, 6, 18, 11}**	**55**
10	Kostanjeva	{6, 9, 18, 11}	44

Iz tabele hitro razberemo, da je vsota let v obeh ekipah enaka le pri kombinaciji {15, 21, 7, 12} za Hrastovo ekipo in {20, 6, 18, 11} za Kostanjevo ekipo.

Računalniško ozadje

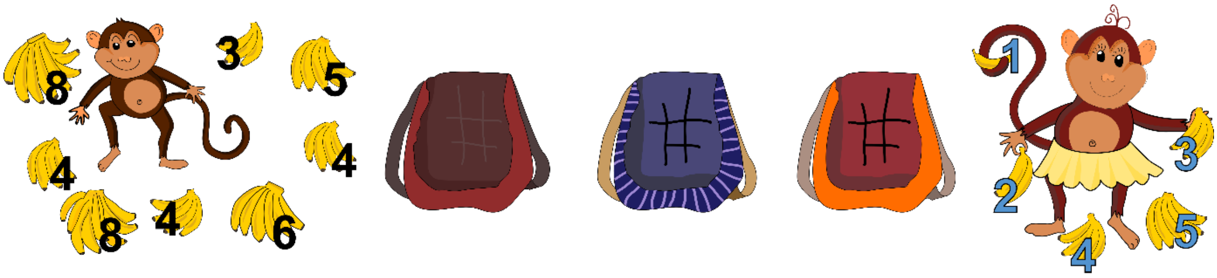
Naloga zahteva, da v dveh množicah (ekipi v naši nalogi) poiščemo podmnožici z enako vsoto. To je poenostavljena različica *problema vsote podmnožic*: za podano množico moramo preveriti, ali obstaja taka podmnožica, da je vsota števil te podmnožice enaka neki vnaprej določeni vrednosti. Problem vsote podmnožic sicer lahko zelo enostavno opišemo, veliko težje pa ga je učinkovito rešiti.

Ena metoda za rešitev tega problema je, da preverimo vse možne podmnožice. Če je problem majhen (kot v naši nalogi), lahko računalnik hitro preveri vse podmnožice in najde rešitev. V naši nalogi lahko skupino 4 igralcev sestavimo na 5 različnih načinov. Ker imamo dve ekipi, moramo skupaj narediti in preveriti 10 kombinacij. Te smo zapisali v tabelo pri rešitvi, računalnik pa bi to nalogo opravil zelo hitro. Vendar z večanjem števila igralcev hitro narašča tudi čas, potreben za iskanje rešitve. Zato potrebujemo bolj učinkovito rešitev: uporabimo dinamično programiranje.

Banane

6. do 9. razred, srednja šola

Opica Mimi se odpravlja na izlet z dvema prijateljema. S seboj želi vzeti banane za malico. Ima več šopov banan, ki jih mora enakomerno razdeliti v tri nahrbtnike, pri tem pa ne sme odtrgati nobene banane iz šopa (šopi banan morajo ostati celi). Če je potrebno, lahko Mimi uporabi tudi katerega od petih šopov banan, ki jih ima mama opica (desno na sliki).



Ali Mimi lahko razdeli šope banan v tri nahrbtnike tako, da bo v vsakem nahrbtniku enako število banan (in pri tem ne razdre nobenega šopa banan)?

A) Ne, četudi mama pomaga Mimi z dodatnimi bananami.
B) Da in pri tem ne potrebuje dodatnih banan od mame.
C) Da, če mama da Mimi eno banano.
D) Da, če mama da Mimi dve banani.
E) Da, če mama da Mimi tri banane.
F) Da, če mama da Mimi štiri banane.
G) Da, če mama da Mimi pet banan.

Rešitev

Pravilen odgovor je E: Mimi lahko razporedi šope banan v tri nahrbtnike, če ji mama da šop treh banan.

Mama opica mora dati šop 3 banan, da lahko Mimi enakomerno razdeli po 15 banan v vsakega od 3 nahrbtnikov. Primer rešitve prikazuje spodnja slika.

Da pridemo do rešitve naloge, moramo najprej poiskati vsoto vseh banan v šopih in ta vsota mora biti deljiva s 3 (ker imamo 3 nahrbtnike).

V vseh šopih, ki jih ima Mimi, je skupaj 3 + 4 + 4 + 4 + 5 + 6 + 8 + 8 = 42 banan. Teh 42 banan lahko razdelimo na 3 dele, in sicer po 14 banan. Sedaj poskusimo sestaviti šope banan tako, da bodo imeli skupaj po 14 banan. Ker je 14 sodo število, bomo morali šope z lihim številom banan sestaviti skupaj. Torej mora biti šop 3 banan v istem nahrbtniku kot šop 5 banan. Če to upoštevamo, moramo v 3 nahrbtnike razdeliti naslednje šope banan: (5 + 3), 8, 8, 4, 4, 4 in 6. En nahrbtnik lahko napolnimo s 14 bananami, če uporabimo šopa z 8 in 6 bananami (ali tri šope 4 + 4 + 6 ali 3 + 5 + 6), vendar pa ostalih dveh nahrbtnikov nikakor ne moremo napolniti, ne da bi razdrli en šop banan.

Torej je edina možnost, da vzamemo še en šop banan od mame opice. Ker pa mora biti število deljivo s 3, lahko vzamemo le šop 3 banan. Tako imamo skupaj 45 banan in v vsak nahrbtnik jih moramo spraviti 15. Najprej spravimo v vsakega od nahrbtnikov največje šope banan, saj je manjše šope lažje porazdeliti: v prvi nahrbtnik damo šop 8 banan, v drugega tudi šop 8 banan, v tretjega pa šop 6 banan. V tretji nahrbtnik dodamo še šopa po 4 in 5 banan, v prvi in v drugi nahrbtnik pa dodamo šopa po 3 in 4 banane.

Računalniško ozadje

Naloga obravnava *problem razdelitve*, pri katerem moramo razdeliti seznam celih števil (šopov banan) na tri podmnožice (nahrbtniki), ki imajo enake vsote. Če je seznam števil dovolj kratek, lahko enostavno ugotovimo, ali rešitev obstaja (in seveda tudi poiščemo rešitev, če obstaja). Ko pa seznam števil narašča, postaja problem vedno težji za hitro rešitev. Zaenkrat za problem razdelitve še ne obstaja učinkovit algoritem, ki bi našel rešitev ali pa le preveril, če rešitev sploh obstaja. Problem spada med *NP-polne probleme*, za katere sicer zelo enostavno preverimo, če je neka rešitev pravilna, a zelo težko poiščemo pravilno rešitev.

Robota

6. do 9. razred, srednja šola

Imamo dva robota,  in , ki se lahko premikata na mreži z enega na drug kvadratek. Za oba robota imamo en sam daljinski upravljalnik, zato na vsak ukaz z daljinca oba robota izvedeta povsem enako akcijo.

Daljinec ima tri ukaze:

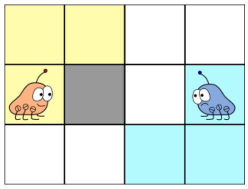
 Premakni se naprej na naslednji kvadratek (v smeri, v katero je robot trenutno obrnjen).

 Obrni se za 90 stopinj v levo (robot pri tem ostane na istem kvadratku).

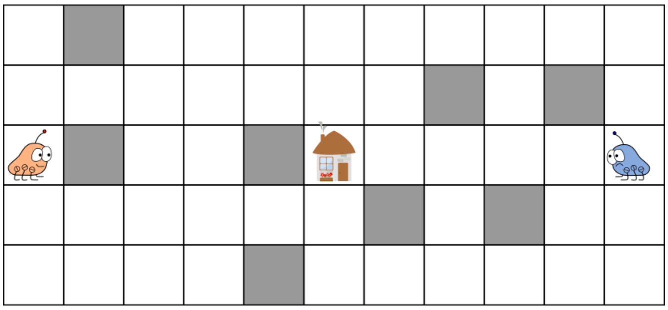
 Obrni se za 90 stopinj v desno (robot pri tem ostane na istem kvadratku).

Na primer, če sta robota postavljena na mrežo kot na desni sliki, se ob zaporedju ukazov

premakneta po obarvanih kvadratkih.



Imamo naslednjo mrežo in na njej postavljena robota:



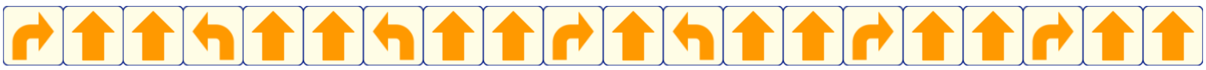
Robotoma podamo zaporedje ukazov, ki oba robota premaknejo do hiše na sredini. Pri tem seveda pazimo, da se robota ne zaletita v oviro (sivi kvadratek) ali se premakneta izven mreže.

Katero je najmanjše število ukazov, ki jih moramo podati robotoma?

Rešitev

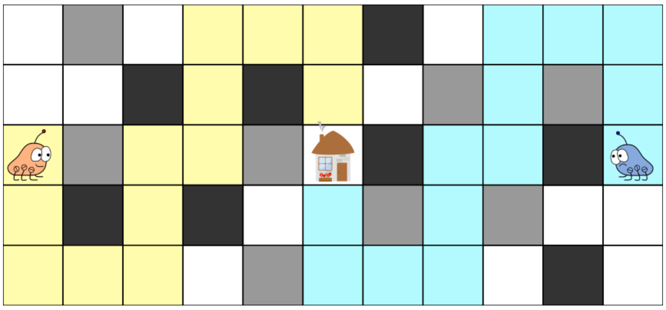
Robotoma podamo zaporedje 20 ukazov.

Da oba robota uspešno prideta do hiše na sredini, jima moramo podati naslednje ukaze:



Rešitev je unikatna (če izločimo redundantne ukaze in upoštevamo le najkrajšo rešitev po številu podanih ukazov).

Kako lahko poiščemo rešitev? Ker se oba robota premikata na enak način, najprej prezrcalimo vse ovire na mreži glede na oba relativna zorna kota, enega za vsakega robota. Prezrcaljene ovire so na mreži označene s črnimi kvadratki.



S tem je pot robota do hiše postala očitna, saj se lahko proti hiši premakne le na en način z izogibanjem oviram. Pot je na sliki pobarvana za vsakega robota. Ko poznamo pot robota, zlahka zapišemo tudi zaporedje ukazov, ki jih mora prejeti robot, in preštejemo te ukaze.

Računalniško ozadje

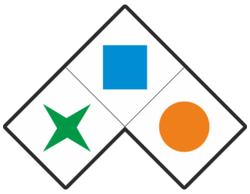
Če več robotov izvaja enako nalogo in zato uporablja enak program, morajo biti ti programi bolj pazljivo napisani, da lahko robot izvede nalogo brez nesreče. To seveda zahteva bolj zapletene programe, ki robota navadno ne peljejo po vnaprej določeni poti (kot v naši nalogi), ampak robot sproti prepoznava ovire in se jim izogiba. Pri tem morajo roboti paziti, da niso drug drugemu v napoto.

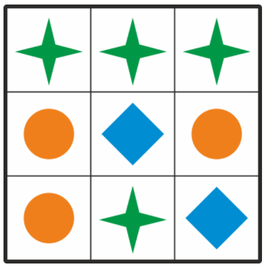
Vzorci

6. do 9. razred, srednja šola

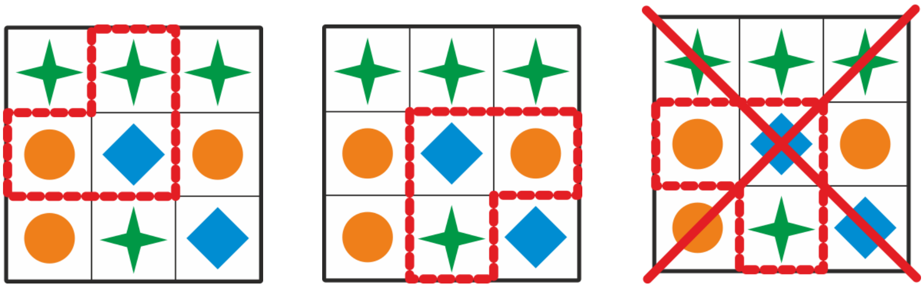
Vrtna miza na desni je okrašena z vzorčastimi ploščicami.

Spodnji vzorec lahko na tej mizi najdemo dvakrat.

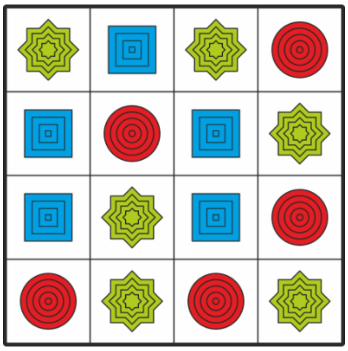




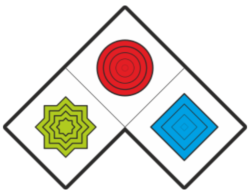
Pri tem lahko vzorec rotiramo, ne smemo pa ga zrcaliti:



Imamo tudi večjo mizo z drugačnim vzorcem:



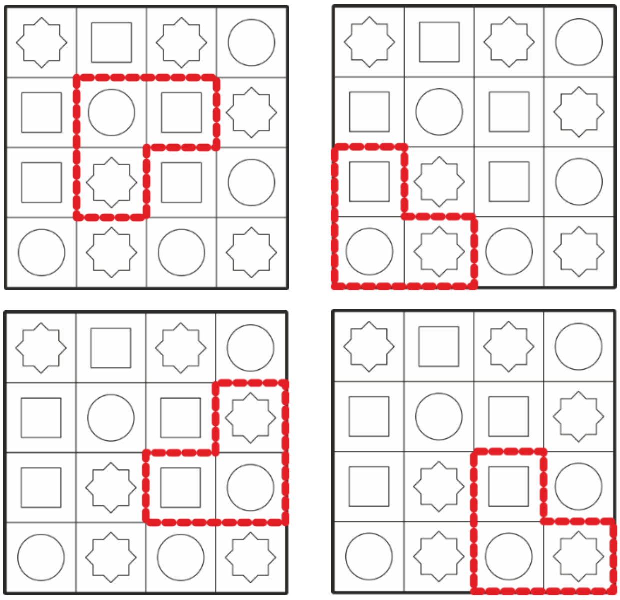
Kolikokrat se na tej mizi pojavi vzorec na desni?



Rešitev

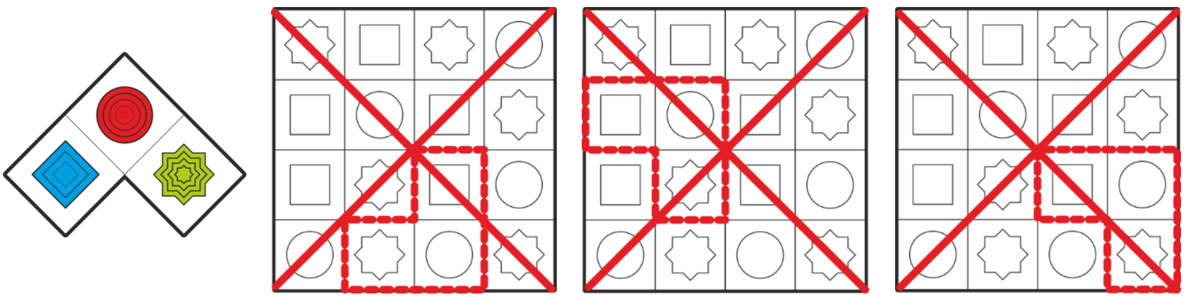
Podan vzorec se pojavi 4-krat.

Spodnje slike prikazujejo, kje se pojavi vzorec:



In kako poiščemo te rešitve? Takoj lahko vidimo, da ima iskan vzorec na sredini rdeč krog, zato lahko na mizi iščemo rdeče kroge. Na vsak rdeči krog na mizi v mislih položimo vzorec in ga obračamo ter sproti preverjamo, ali se ujema z vzorcem na mizi.

Paziti moramo, da se iskani vzorec popolnoma ujema z vzorcem na mizi, saj ga ne smemo zrcaliti. Na mizi sicer najdemo tudi podobne vzorce, le da imajo zamenjano zvezdo in kvadrat (leva spodnja slika). Tak vzorec bi videli v ogledalu, torej je zrcaljen. Zrcaljen vzorec na mizi najdemo trikrat, a te rešitve ne štejejo.



Računalniško ozadje

Iskanje ponavljajočih se vzorcev v podatkih je v računalništvu zelo pomembna naloga, ki jo imenujemo *razpoznavanje vzorcev*. Ta naloga je ključnega pomena za umetno inteligenco in strojno učenje, saj omogoča, da se računalnik uči iz podatkov ter dela napovedi ali odločitve. Tak primer je razpoznavanje slik, kjer računalnik z iskanjem vzorcev na sliki identificira objekte, jih razvrsti v kategorije in lahko tudi prepozna obraze. Pri obdelavi naravnega jezika razpoznavanje vzorcev omogoča računalniku, da razume strukturo stavka, iz besedila izlušči teme ali pa ustvari novo besedilo (ki ne moremo ločiti od besedila, ki bi ga ustvaril človek).

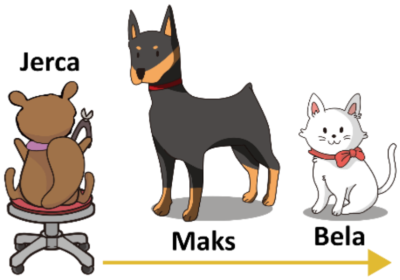
Podoben pristop lahko uporabimo tudi pri programiranju, ko poiščemo ponavljajoče se podnaloge in jih rešimo le enkrat. S takim modularnim pristopom k rešitvi zmanjšamo ponavljanje kode in naredimo program krajši in bolj razumljiv, s tem pa tudi olajšamo razhroščevanje in vzdrževanje programa.

Stilist za ljubljenčke

8. in 9. razred

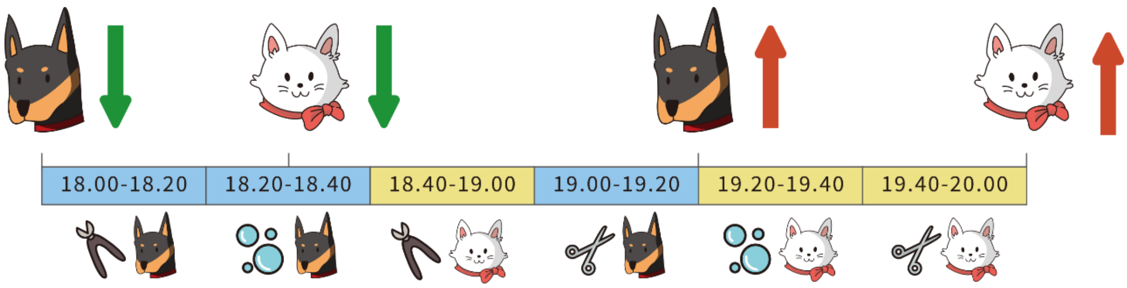
Jerca, stilistka za hišne ljubljenčke, potrebuje 60 minut za ureditev enega. Da lažje načrtuje svoj dnevni urnik, Jerca vsako urejanje razdeli na tri opravila po 20 minut: urejanje krempljev (), kopanje () in striženje dlake (). Za urejanje vsakega hišnega ljubljenčka mora opraviti vsa tri opravila, v tem vrstnem redu. Vsako opravilo mora dokončati, preden začne z naslednjim opravilom.

Jerca se vedno drži tudi naslednjih pravil, ki ji zagotavljajo, da je vsak hišni ljubljenček na koncu v celoti urejen:



1. Ko hišni ljubljenček pride v salon, ga postavi na konec vrste.
2. Ko za posameznega hišnega ljubljenčka zaključi vsa tri opravila, ta zapusti salon (in ni več v vrsti).
3. Urejanje začne pri prvem hišnem ljubljenčku v vrsti, ureja pa le enega naenkrat.
4. Ko konča trenutno opravilo, Jerca vzame v urejanje naslednjega hišnega ljubljenčka v vrsti.
5. Ko Jerca začne z opravilom, dela tako dolgo, da ga dokonča.
6. Ko Jerca konča opravilo na zadnjem hišnem ljubljenčku v vrsti, se premakne nazaj na začetek vrste.

Na primer, dva hišna ljubljenčka, Bela in Maks, sta prišla v salon včeraj. Maks je prišel ob 18.00 in Bela ob 18.30. Spodnja slika prikazuje potek dela v salonu za vsakega od njiju.



Danes so v salon prišli trije hišni ljubljenčki:

ob 10.00

ob 10.30

ob 11.20

Mimi Koder Medo

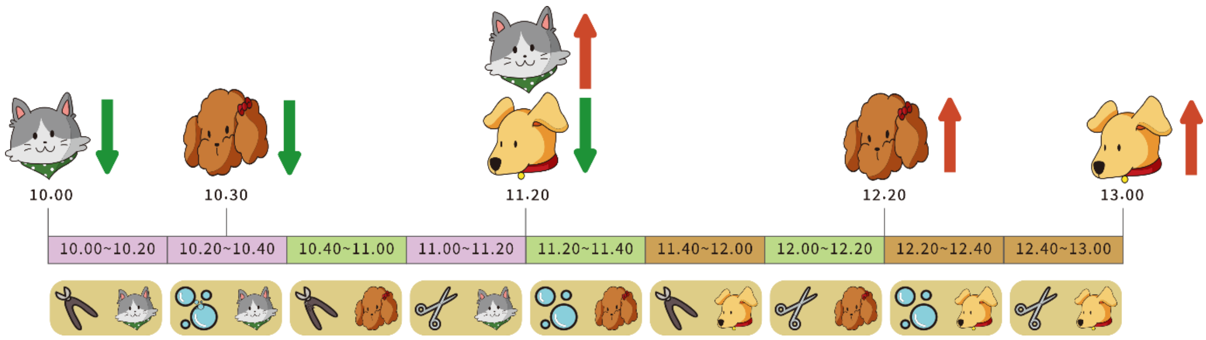
Katera izjava je pravilna?

A) Mimi zapusti salon ob 11.40.
B) Koder zapusti salon ob 12.20.
C) Medo zapusti salon ob 12.40.
D) Nobena izjava ni pravilna.

Rešitev

Pravilen je odgovor B: Koder zapusti salon ob 12.20.

Na podlagi časov prihodov se Jerca posveti hišnim ljubljenčkom, kot prikazuje spodnja slika:



Ob 11.20 Jerca konča z delom na Mimi, ko pride Medo. Pravila pravijo, da mora Jerca urejati naslednjega v vrsti, ki je Koder, in Medo pristane na koncu vrste. Torej, Jerca začne z urejanjem Meda, ko je Koder okopan. Po urejanju Medovih krempljev pa postriže še Kodra (po odhodu Mimi je Koder prvi v vrsti in ker je Jerca z Medom prišla že do konca vrste, se vrne na začetek, torej je naslednji na vrsti Koder) in s tem zaključi njegovo urejanje. Koder lahko zapusti salon ob 12.20.

Računalniško ozadje

Čeprav ima računalnik le en procesor, ki lahko izvaja le en ukaz naenkrat, lahko računalnik »hkrati« izvaja tudi več opravil. Za razdeljevanje procesorske moči med različnimi opravili skrbi *razvrščevalnik*, ki je del operacijskega sistema. Razvrščanje opravil izvaja na različne načine glede na uporabljen *razvrščevalni algoritem*, ki vpliva na to, kako učinkovito (in pravično) bo med opravila razdeljena procesorska moč.

V naši nalogi je procesor Jerca, ki mora izvesti več opravil na hišnih ljubljenčkih. Pravila, ki jih je postavila Jerca za urejanje hišnih ljubljenčkov, predstavljajo razvrščevalni algoritem. Način, kako Jerca ureja enega hišnega ljubljenčka 20 minut in se nato premakne na naslednjega, je podoben *krožnemu izvajanju* (angl. *Round Robin*): opravila so v vrsti, razvrščevalnik določi vsakemu opravilu enak časovni interval in med njimi kroži. Ko se neko opravilo konča v celoti, se odstrani iz vrste (sicer pa ostane v vrsti, dokler ni v celoti opravljeno).

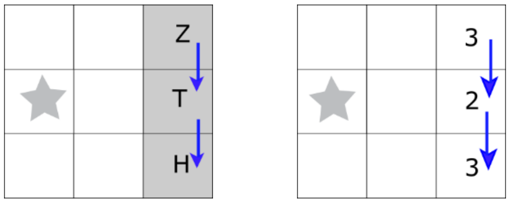
Poleg kroženja obstaja več različnih razvrščevalnih algoritmov, kot so »prvi pride, prvi postrežen« ali FIFO, »najkrajši najprej« in »po prioriteti«. Posebnost algoritma krožnega izvajanja je, da se trudi zagotoviti, da procesor na posamezni nalogi na izostane predolgo, da se opravila začnejo čimprej, da se čaka čim manj in da so čakalni časi enaki za vse. V naši nalogi čaka Koder le 10 minut, čeprav je prišel v salon za Mimi. Če bi Jerca uporabljala algoritem »prvi pride, prvi postrežen« in bi ljubljenčke urejala v celoti (urejanje kremljev, kopanje in striženje dlake) po vrsti, bi moral Koder počakati, da Jerca dokonča urejanje Mimi, torej bi čakal kar 30 minut.

Toplo - hladno

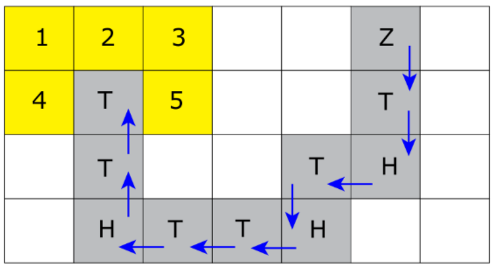
8. in 9. razred

Danijel igra igro iskanja zaklada, ki je zakopan nekje na kvadratni mreži. Iskanje začne z začetnega polja, označenega z **Z**, in se lahko premika po **en korak naenkrat** le vodoravno ali navpično na sosednja polja. Po vsakem koraku Danijel prejme sporočilo, ali je bližje zakladu (toplo, T) ali pa se je od zaklada oddaljil (hladno, H). **Razdalja** do zaklada je najmanjše število korakov do zaklada.

Na primer, na mreži 3 x 3, ki je prikazana spodaj, je zaklad zakopan pod z zvezdico označenim poljem. Danijel gre najprej dva koraka naprej, kot kažejo puščice na levi mreži. Pri tem Danijel dobi sporočili “T” in “H”. Števila na poljih desne mreže prikazujejo razdaljo do zaklada na tem polju.



V novi igri dobi Danijel mrežo velikosti 4 x 7. Njegova pot je označena s puščicami, v poljih pa so vpisana tudi prejeta sporočila (T ali H). Danijel dobi še namig, da je zaklad pod enim od rumenih polj, ki so označena s številkami od 1 do 5.

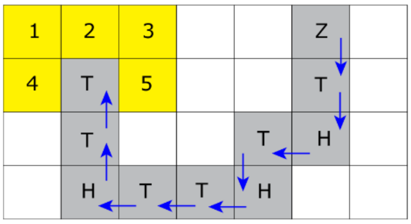


Pod katerim oštevilčenim poljem je zakopan zaklad?

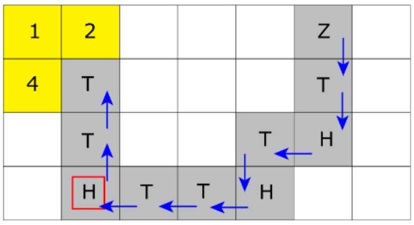
Rešitev

Pravilni odgovor je polje 5.

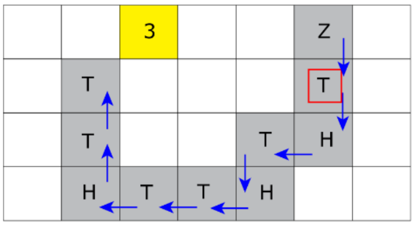
Vidimo, da se oznake T in H na Danijelovi poti lahko spreminjajo glede na premike v smeri puščic. Če se pri premiku na naslednje polje bolj približa zakladu, je na polju oznaka T, če pa se od zaklada oddalji, je na polju zapisan H. Začne na polju Z.



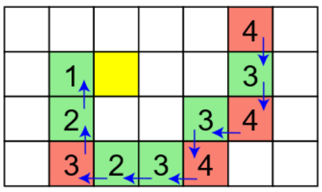
Poglejmo spremembo oznake po 7. premiku (premik s 3. na 2. stolpec v mreži); na spodnji sliki je označena z rdečim kvadratom. Ko se je Danijel premaknil v levo s 4. na 3. stolpec, se je premaknil bližje zakladu, saj je dobil sporočilo T. Ko pa se je premaknil levo s 3. na 2. stolpec, je dobil sporočilo H, kar pomeni, da se je oddaljil od zaklada. To pomeni, da mora biti zaklad v 3. stolpcu. Zato zaklad ne more biti pod polji 1, 2 in 4.

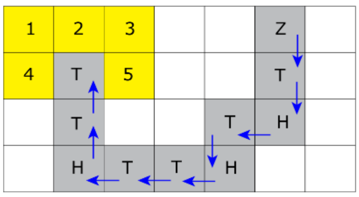


Sedaj pa poglejmo še Danijelove premike v navpični smeri. Z začetnega mesta v 1. vrstici (Z) se je premaknil v 2. vrstico in pri tem dobil sporočilo T, kar pomeni, da se je premaknil bližje zakladu. Če bi bil zaklad pod poljem 3, bi moral dobiti sporočilo H, saj bi premik pomenil oddaljevanje od zaklada. To pomeni, da tudi pod poljem 3 ne more biti zaklada.



Ostane nam le še polje 5. Da se prepričamo, da to polje res skriva zaklad, preverimo, če so oznake T in H na mreži pravilno zapisane. V mrežo na spodnji levi sliki smo za vsako polje na Danijelovi poti zapisali razdaljo do polja 5 (na sliki označeno rumeno). Če se je razdalja pri premiku zmanjšala, smo polje pobarvali zeleno, če pa se je razdalja povečala, je polje rdeče. Tudi začetno polje je rdeče. Torej bi na rdečih poljih morale biti oznake H, na zelenih pa T. Vidimo, da se slika Danijelove poti (na desni) ujema z našo sliko na levi.





Tako smo se prepričali, da je zaklad res zakopan pod poljem 5.

Računalniško ozadje

Spodbujevalno učenje (angl. *reinforcement learning*) je tehnika strojnega učenja, pri kateri na podlagi povratne informacije prek nagrajevanja (oziroma kaznovanja) dosežemo priučitev vedenja (oziroma optimizacijo vedenja). Uporablja se v računalniških sistemih za optimizacijo enostavnih vzorcev vedenja in postopno priučitev kompleksnejšega obnašanja.

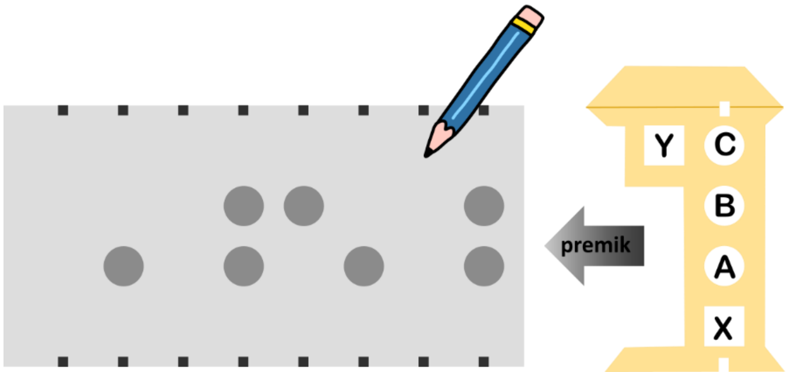
V naši nalogi se Danijel uči poiskati optimalno pot do zaklada. Premika se po mreži, ob vsakem premiku pa prejme povratno informacijo glede na razdaljo do zaklada: T za pozitiven signal in H za negativen signal. Danijel se glede na prejete signale sproti optimalno odloča za naslednji korak.

Avtonomna vožnja je eden od primerov uporabe zgornje metode. Da lahko uspešno deluje v nepredvidljivem okolju, mora samovozeči avtomobil opravljati številne meritve, načrtovanja in predvidevanja, kot sta na primer načrtovanje poti in napoved gibanja.

Pametno ravnilo

8. in 9. razred, srednja šola

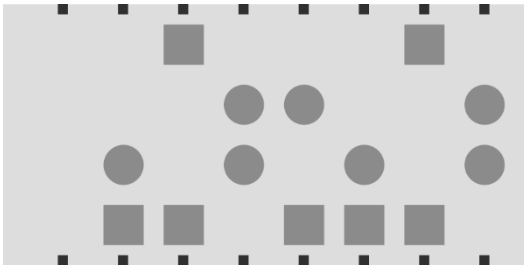
Pametno ravnilo ima 5 lukenj (A, B, C, X, Y), s svinčnikom pa s pomočjo ravnila (lukenj X in Y) rišemo sive kvadratke na listu papirja.



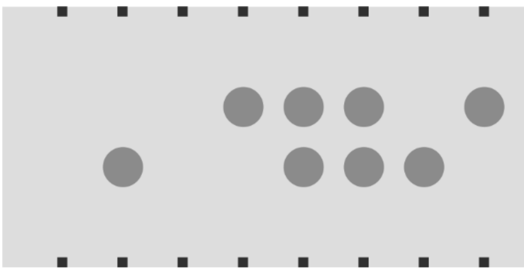
Postopek je naslednji:

- Začnemo na prvem stolpcu na desni.
- Če skozi luknje A, B in C v ravnilu vidimo dve ali več sivih pik, narišemo siv kvadratek na Y. (Ta siv kvadratek bomo videli kot sivo piko, ko ga bomo pogledali skozi luknjo C.)
- Če je število sivih pik, ki jih vidimo skozi A, B in C, liho, narišemo siv kvadratek na X.
- Premaknemo se za en stolpec na levo in ponovimo postopek, dokler ne pridemo do skrajno levega stolpca.

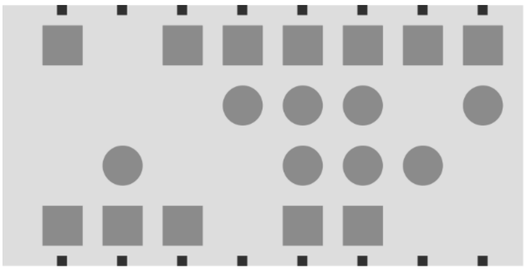
Če pametno ravnilo uporabimo na zgornjem primeru, dobimo naslednji rezultat:



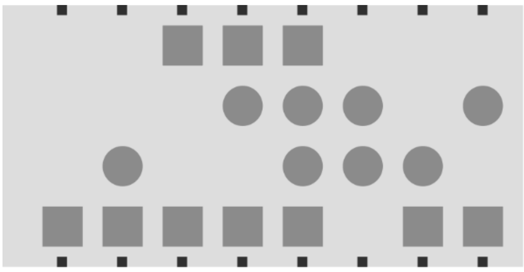
Kakšen rezultat bi dobili, če bi pametno ravnilo uporabili na primeru na spodnji sliki?



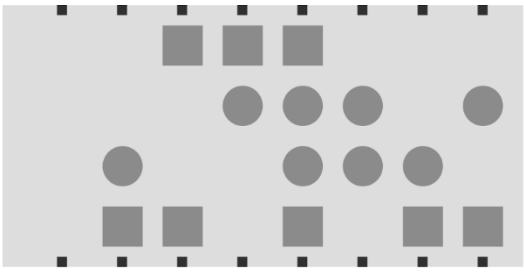
A)



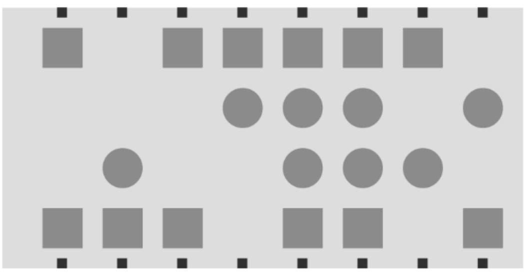
B)



C)



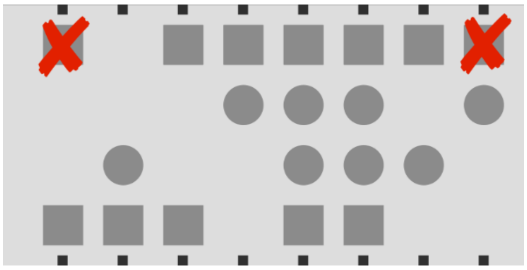
D)



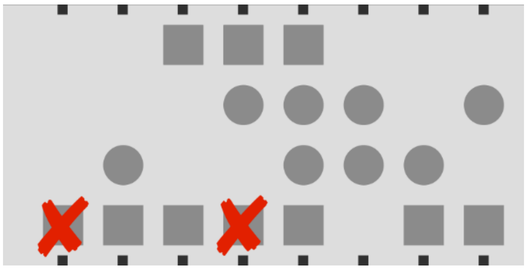
Rešitev

Pravilen odgovor je C.

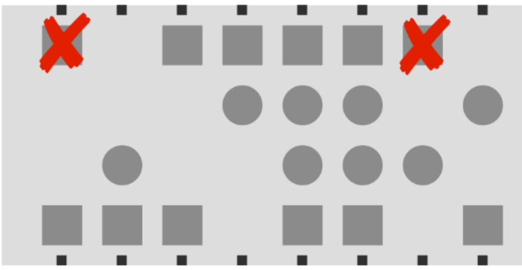
Odgovor A ni pravilen, ker v prvem stolpcu nismo mogli narisati sivega kvadratka na mestu Y, saj je Y v stolpcu levo od A, B in C, skozi katere gledamo pike v prvem stolpcu. Poleg tega je napaka tudi v predzadnjem stolpcu, kjer smo skozi luknje videli le eno sivo piko (A), a smo vseeno posivili kvadratek na Y.



Odgovor B ni pravilen, ker v petem stolpcu z desne nismo upoštevali pravila postopka: skozi ravnilo smo videli dve sivi piki (pri C in pri B), kar je sodo število, a smo vseeno narisali siv kvadratek na X. Poleg tega smo kvadratek na X narisali tudi v zadnjem stolpcu, čeprav skozi luknje nismo videli nobene sive pike.



Odgovor D ni pravilen, ker smo v drugem stolpcu skozi luknje videli le eno sivo piko (B), a smo vseeno posivili kvadratek na Y. Poleg tega je napaka tudi v predzadnjem stolpcu, kjer smo skozi luknje videli le eno sivo piko (A), a smo vseeno posivili kvadratek na Y.



Ostane le še odgovor C, pri katerem zlahka preverimo, da smo upoštevali vsa pravila postopka uporabe pametnega ravnila.

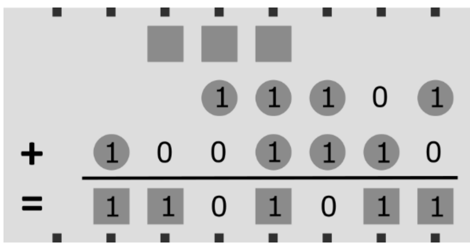
Računalniško ozadje

Če pozorno analiziramo delovanje ravnila v tej nalogi, hitro ugotovimo, da ravnilo pravzaprav izvede dvojiško seštevanje: vzorca v drugi in tretji vrstici predstavljata dvojiški števili (krogec pomeni 1, prazen prostor pa 0), vzorec v četrti vrstici pa je vsota teh dveh števil (glede na tretje pravilo). V prvi vrstici označimo, če je pri seštevanju prišlo do prenosa bita in ga zato upoštevamo v naslednjem stolpcu (drugo pravilo).

Poglejmo primer seštevanja:

$(11101)_2 + (1001110)_2 = (1101011)_2$

ali desetiško 29 + 78 = 107



Algoritem je tako preprost, da ga lahko izvedemo z leseno napravo in uporabo frnikol, kot je na primer *Marble Binary Adding machine* (https://makezine.com/projects/marble-adding-machine/).

Parkirne dovolilnice 2 8. in 9. razred, srednja šola

Bobrsko ministrstvo za promet izda omejeno število parkirnih dovolilnic. Vsaka dovolilnica ima svojo oznako, ki ustreza naslednjim pravilom:

- sestavljena je iz natanko treh velikih tiskanih črk, ki jim sledi ena števka,
- prva črka je soglasnik,
- tretja črka je samoglasnik,
- črke se lahko ponavljajo.

Pri tem uporabljajo abecedo:

Samoglasniki so postavljeni na pikčastem ozadju (zelene črke),
soglasniki so postavljeni na črtastem ozadju (rdeče črke).

Koliko takšnih parkirnih dovolilnic lahko izdajo?

A) 10 B) 46 C) 270 D) 4 200 E) 17 280 F) neomejeno

Rešitev

Na prvem mestu je soglasnik, torej imamo 7 različnih možnosti.

Na drugem mestu je katerakoli črka bobrske abecede, zato imamo 12 možnosti.

Na tretjem mestu stoji samoglasnik, za kar imamo 5 možnosti.

Na zadnjem mestu stoji števka, torej imamo 10 možnosti.

Skupaj: $7 \cdot 12 \cdot 5 \cdot 10 = 4\,200$ možnosti.

Računalniško ozadje

V veliko primerih morajo biti podatki sestavljeni po točno določenih pravilih. En primer so registrske oznake vozil (kot v tej nalogi, četudi Bobrsko ministrstvo za promet postavlja povsem drugačna pravila, kot veljajo v Sloveniji), drug primer pa gesla za prijavo v računalnik (dobro geslo mora, na primer, vsebovati majhne in velike črke, številke in posebne znake ter biti dolgo najmanj 12 znakov). Programi, ki obdelujejo vhodne podatke, morajo tudi preverjati, če podatki ustrezajo podanim pravilom, saj lahko napačni vhodni podatki (ki se ne skladajo s podanimi pravili) povzročijo velike težave. Zamisli si nastalo zmedo, če bančni sistem ne bi preverjal vrednosti plačila in bi lahko na račun prejemnika plačali negativni znesek! To bi dejansko pomenilo, da s prejemnikovega računa denar dvignemo in ga prenesemo na svoj račun.

Odpri

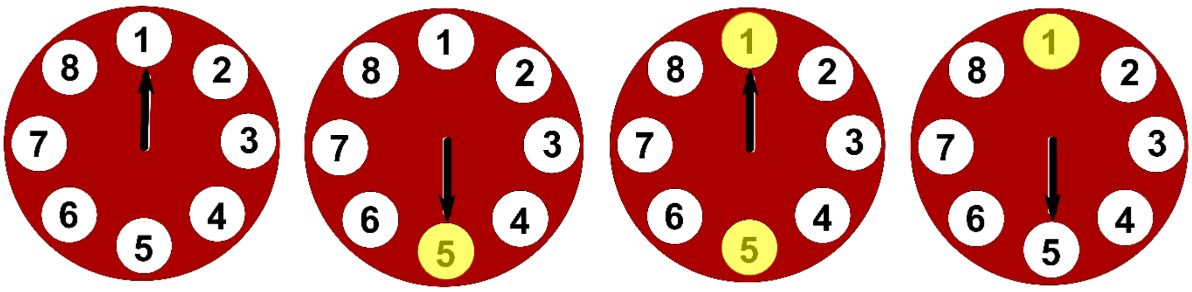
8. in 9. razred, srednja šola

Bober lahko odklene trezor, če vključi določeno kombinacijo luči.

Ob vsaki potezi lahko obrne puščico za 3 ali 4 mesta v smeri urinega kazalca. Puščica spremeni vključenost luči na mestu, kjer se ustavi: če je luč vklopljena, jo izklopi; če je izklopljena, jo vklopi.

Na primer, to se zgodi, ko bober izvede 3 poteze, kjer se puščica vsakič obrne za 4 mesta:

Začetni položaj | Po 1. potezi (4 mesta) | Po 2. potezi (4 mesta) | Po 3. potezi (4 mesta)



Za odpiranje trezorja mora bober osvetliti samo luči na mestih 7 in 8 (na ostalih mestih morajo biti luči izključene).

Najmanj koliko potez mora zaporedno opraviti bober, da vključi samo luči na mestih 7 in 8, če začne z začetnega položaja zgoraj?

Rešitev

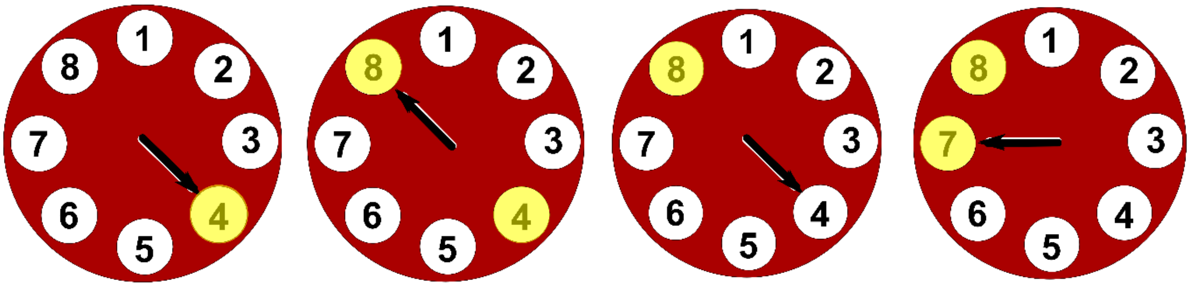
Bober mora narediti najmanj 4 poteze.

Eden od načinov za reševanje tega problema je, da malo razmislimo in ugotovimo, da lahko 7 in 8 vklopimo s po dvema premikoma iz začetnega položaja. Da pridemo do 7, moramo opraviti dve potezi, in sicer dva premika za 3 mesta. Da dosežemo 8, pa se moramo premakniti enkrat za 4 mesta in enkrat za 3 mesta.

Razmislimo naprej. Če puščica kaže na 8, lahko 7 dosežemo v dveh potezah, v obratno smer to ne velja: da dosežemo 8 z mesta 7, je potrebnih več potez in zaplete rešitev. To nam nakazuje, da je obetavna rešitev ravno ta, da najprej pridemo do 8, nato pa do 7, ki bi lahko morda bila izvedljiva v 4 potezah. Ampak s 4 potezami seveda spremenimo tudi vmesna mesta, na katerih pristane puščica. Tako moramo, da bi ohranili vklopljeni le uči na mestih 7 in 8 in ne ostalih luči, zagotoviti, da vsakič pristanemo na istih številih – v obeh primerih, ko gremo na 8 in nato še ko gremo do 7.

To deluje, če se najprej premikamo tako: najprej za 3 mesta (vklopimo 4), nato za 4 mesta (vklopimo 8), nato spet za 4 mesta (izklopimo 4) in na koncu za 3 mesta, da vklopimo 7.

1. poteza 3 mesta	2. poteza 4 mesta	3. poteza 4 mesta	4. poteza 3 mesta



Da so 4 poteze najboljša rešitev, lahko preverimo tudi tako, da sistematično pregledamo vsa možna zaporedja potez.

Če bober naredi le 1 potezo:

1. poteza	Vključene številke
3 mesta	4
4 mesta	5

Z eno potezo lahko bober vklopi le eno luč.

Če bober opravi dve potezi:

1. poteza	2. poteza	Številka mesta, na katerem pristane	Vključene številke
3 mesta	3 mesta	4, 7	4, 7
3 mesta	4 mesta	4, 8	4, 8
4 mesta	3 mesta	5, 8	5, 8
4 mesta	4 mesta	5, 1	5, 1

Z uporabo 2 potez lahko bober vključi luči na dveh mestih, ampak ti mesti nista 7 in 8.

Če opravi 3 poteze:

1. poteza	2. poteza	3. poteza	Številke	Vključene številke
3 mesta	3 mesta	3 mesta	4, 7, 2	4, 7, 2
3 mesta	3 mesta	4 mesta	4, 7, 3	4, 7, 3
3 mesta	4 mesta	3 mesta	4, 8, 3	4, 8, 3
3 mesta	4 mesta	4 mesta	4, 8, 4	8
4 mesta	3 mesta	3 mesta	5, 8, 3	5, 8, 3

4 mesta	3 mesta	4 mesta	5, 8, 4	5, 8, 4
4 mesta	4 mesta	3 mesta	5, 1, 4	5, 1, 4
4 mesta	4 mesta	4 mesta	5, 1, 5	1

Z uporabo 3 potez lahko bober vključi eno ali tri številke. Na tak način ne more vključiti 7 in 8.

Če bober opravi 4 poteze:

1. poteza	2. poteza	3. poteza	4. poteza	Številke	Vključene številke
3 mesta	3 mesta	3 mesta	3 mesta	4, 7, 2, 5	4, 7, 2, 5
3 mesta	3 mesta	3 mesta	4 mesta	4, 7, 2, 6	4, 7, 2, 6
3 mesta	3 mesta	4 mesta	3 mesta	4, 7, 3, 6	4, 7, 3, 6
3 mesta	3 mesta	4 mesta	4 mesta	4, 7, 3, 7	4, 3
3 mesta	4 mesta	3 mesta	3 mesta	4, 8, 3, 6	4, 8, 3, 6
3 mesta	4 mesta	3 mesta	4 mesta	4, 8, 3, 7	4, 8, 3, 7
3 mesta	4 mesta	4 mesta	3 mesta	4, 8, 4, 7	8, 7
3 mesta	4 mesta	4 mesta	4 mesta	4, 8, 4, 8	-
4 mesta	3 mesta	3 mesta	3 mesta	5, 8, 3, 6	5, 8, 3, 6
4 mesta	3 mesta	3 mesta	4 mesta	5, 8, 3, 7	5, 8, 3, 7
4 mesta	3 mesta	4 mesta	3 mesta	5, 8, 4, 7	5, 8, 4, 7
4 mesta	3 mesta	4 mesta	4 mesta	5, 8, 4, 8	5, 4
4 mesta	4 mesta	3 mesta	3 mesta	5, 1, 4, 7	5, 1, 4, 7
4 mesta	4 mesta	3 mesta	4 mesta	5, 1, 4, 8	5, 1, 4, 8
4 mesta	4 mesta	4 mesta	3 mesta	5, 1, 5, 8	1, 8
4 mesta	4 mesta	4 mesta	4 mesta	5, 1, 5, 1	-

S 4 potezami je možno vključiti le luči na mestih 7 in 8, in sicer le na en način (ki smo ga našli že z razmislekom). Torej je najmanjše število potez, da sta vključeni le luči na mestih 7 in 8, 4 poteze.

Računalniško ozadje

V tej nalogi dobimo pravila (navodila), ki jih moramo ponoviti večkrat. Ne vemo pa, koliko krat moramo posamezno previlo uporabiti, da dosežemo cilj.

En način reševanja problema je, da sestavimo program, ki izpiše vse možne kombinacije potez (v tem primeru 4) in preverimo rezultat v vsakem primeru. Računalničarji temu rečejo bruteforce search – iskanje z grobo silo. Kakorkoli že, ta metoda postane zelo neprimerna, ko ne poznamo števila ponovitev ali če je količina možnosti prevelika za spomin.

V tem primeru je podobno kot v metodi "poskusi z napako", kjer poskušamo najti rešitev z večkratnim sledenjem pravilom. Pomembno je, da vemo, da prva rešitev, ki jo dobimo na takšen način, ni nujno tudi najboljša, razen če začnemo z najboljšimi začetnimi navodili. V našem primeru moramo začeti z le 1 potezo in nato večamo število potez.

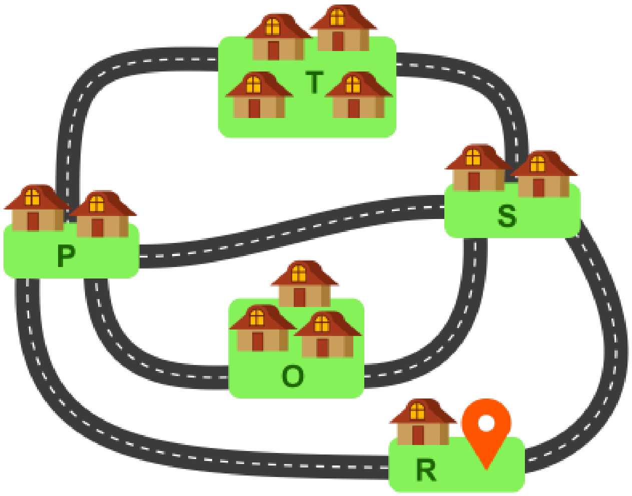
V informatiki predstavljajo številski sistemi ključno vlogo pri predstavitvi in delu s podatki. Krožna postavitev števil v nalogi, podobno kot pri uri, predstavlja specifični številski sistem. Računalniki za predstavitev in obdelavo podatkov uporabljajo različne številske sisteme, kot so dvojiški oziroma binarni (osnova je 2), decimalni (osnova je 10) ali šestnajstiški (osnova je 16). Številke v nalogi so kot števila na uri, a je namesto od 1 do 12 je njihov razpon od 1 do 8.

Razumevanje številskih sistemov in njihovega prikaza je v informatiki osnovnega pomena in pomaga računalnikom učinkovito shranjevati, procesirati in prenašati podatke.

Snemanje cest

8. in 9. razred, srednja šola

Podjetje, ki digitalizira zemljevide, želi z enim avtom posneti slike vseh cest, ki povezujejo vasice na spodnjem zemljevidu. Zaradi časovne omejitve imajo za snemanje vseh poti na voljo samo sedem ur. Izberejo lahko katerokoli pot iz posamezne vasice, vendar potrebujejo za vsako pot med dvema vasicama natanko eno uro.



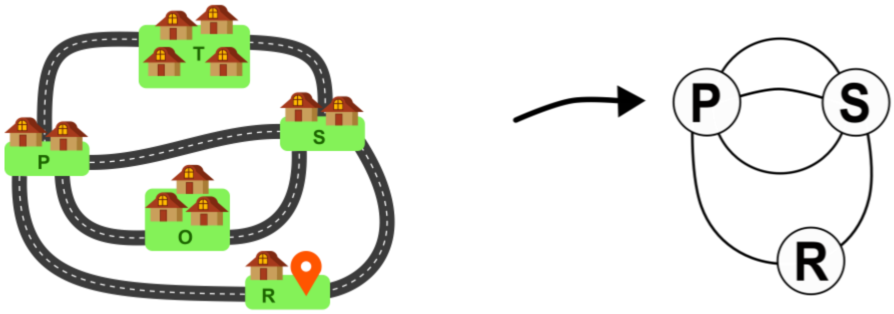
Koliko različnih zaporedij poti lahko izberejo, če začnejo v vasici R, da svoje delo dokončajo v 7 urah?

Rešitev

Iz vasice R lahko prevozijo vse ceste po 12 različnih poteh.

Ker morajo posneti vožnjo po sedmih cestah v sedmih urah, mora avto prevoziti vsako cesto natanko enkrat.

Avto lahko med različnimi potmi izbira le v določenih vaseh: R, kjer začne in konča svojo pot (lahko začne z vožnjo levo ali desno), ter v P in S, ki ju povezuje več poti. Problem si lahko poenostavimo tako, da si narišemo abstraktni model povezav med vasmi in v njem zanemarimo vasici O in T, kjer je smer potovanja določena s smerjo vožnje do vasi (ne moremo izbirati poti).



Ker moramo začeti iz R in moramo prevoziti vsako pot natanko enkrat, se moramo vrniti in končati pot v R.

Najprej preglejmo, kaj se zgodi, če gremo iz R na desno proti S. Edina možnost izbire poti na naši poti je izbira med tremi potmi, ki vodijo iz vasice S v P. Ker imamo na voljo tri poti in ker moramo prevoziti vse tri poti med S in P, je možnosti šest.

Da bi lažje razložili ta korak, te tri poti poimenujmo zgornja, srednja, spodnja. Začnemo lahko s katerokoli od teh treh, ampak se po odločitvi lahko vrnemo samo po eni od preostalih dveh; in ko se odločimo za drugo pot, s tem določimo tudi tretjo pot - to je tista, ki je še nismo uporabili. Tako je število možnosti: 3 x 2 x 1 = 6, kot prikazuje razpored:

zgornja, srednja, spodnja	srednja, zgornja, spodnja	spodnja, zgornja, srednja
zgornja, spodnja, srednja	srednja, spodnja, zgornja	spodnja, srednja, zgornja

Tako imamo, če začnemo iz R na desno, na voljo šest različnih kombinacij poti med vasicama P in S.

Kaj pa, če gremo iz R na levo proti P? Tudi tukaj imamo na voljo 6 različnih kombinacij poti med vasicama P in S, ker imamo enake izbire kot prej, le da gremo z leve na desno – torej v obratni smeri kot prej.

Skupno imamo tako 12 možnih kombinacij poti, ki natanko enkrat obiščejo vsako pot med vasicami.

Tukaj je seznam vseh 12 možnih kombinacij poti med vasicami:

R → S → T → P → S → O → P → R	R → P → O → S → P → T → S → R
R → S → T → P → O → S → P → R	R → P → O → S → T → P → S → R
R → S → P → T → S → O → P → R	R → P → S → O → P → T → S → R
R → S → P → O → S → T → P → R	R → P → T → S → O → P → S → R
R → S → O → P → S → T → P → R	R → P → T → S → P → O → S → R
R → S → O → P → T → S → P → R	R → P → S → T → P → O → S → R

Računalniško ozadje

Graf je sestavljen iz množice vozlišč (običajno predstavljeni kot točke ali majhni krožci) in množice povezav. Vsaka povezava povezuje dve vozlišči. Zaporedje povezanih vozlišč se v grafu imenuje pot. V našem problem so vasice vozlišča, ceste pa povezave.

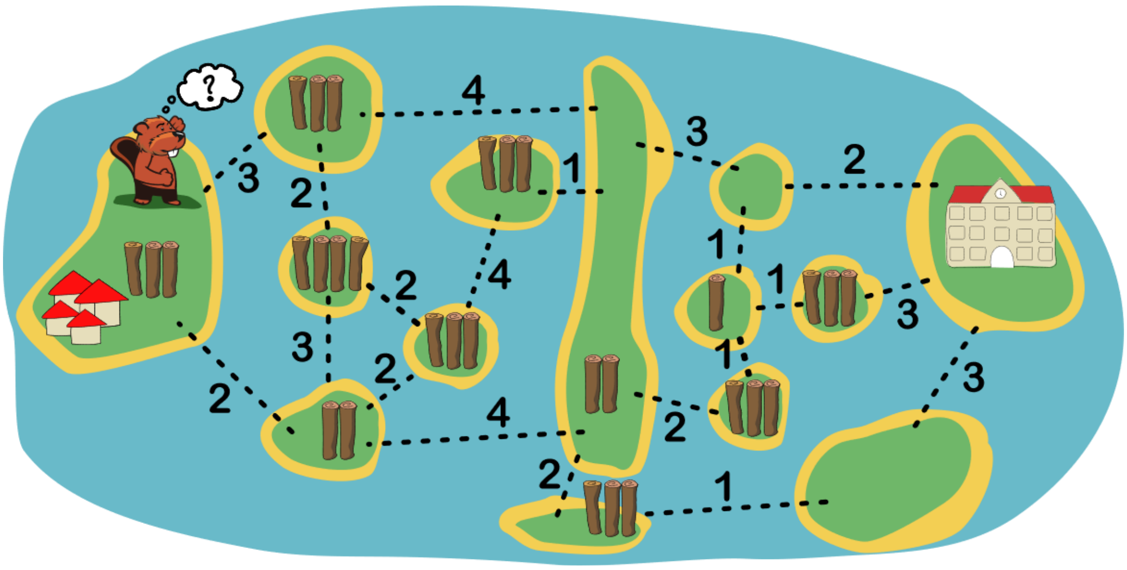
Graf v tej nalogi je poseben, ker je povezan (obstaja pot med katerimakoli dvema vozliščema) in nima vozlišč z lihim številom povezav. Za grafe, ki imajo ti dve lastnosti obstaja pot, po kateri lahko vse povezave prepotujemo natanko enkrat in se vrnemo na začetno vozlišče – imenujemo jih Eulerjev obhod. Prvi jih je matematično raziskoval Leonhard Euler, ko je reševal znameniti problem Sedem mostov Köniugsberga iz leta 1736. Eulerjeve obhode lahko najdemo z uporabo Fleuryjevega algoritma ali Hierholzerjevega algoritma.

Mostovi

8. in 9. razred, srednja šola

Božo gradi mostove. Njegovo naslednje opravilo je gradnja mostov med vasjo, v kateri živijo učenci, in novo šolo na otoku.

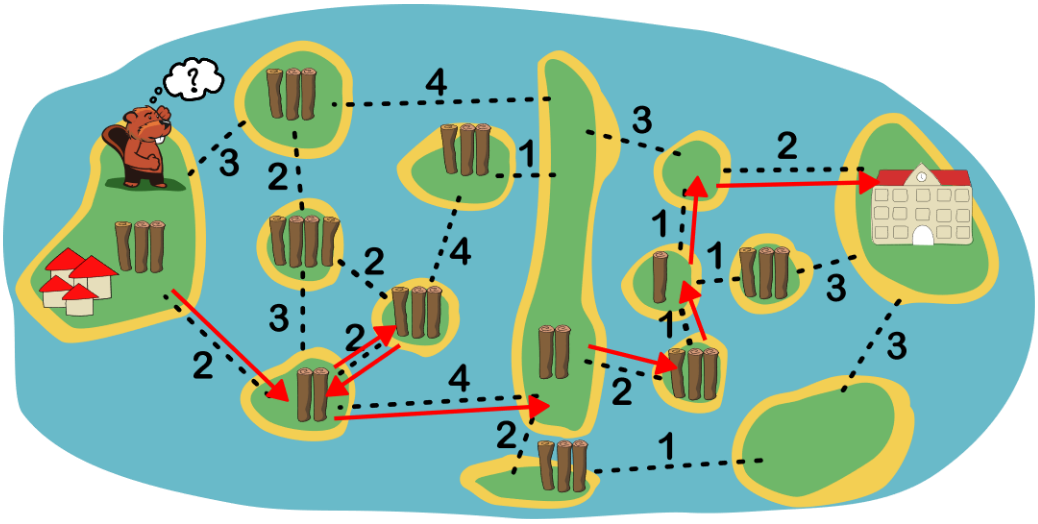
- Ker Božo ne zna plavati, mora najprej zgraditi most, ki ga lahko prečka, kolikorkrat je to potrebno.
- Za gradnjo vsakega mostu potrebuje dovolj debel. V svoji vasi lahko vzame le 3 debla, da lahko začne z gradnjo. Potuje lahko med vsemi otoki, ki so že povezani z mostom in si na njih nabere dodatna debla za gradnjo. Spodnji zemljevid prikazuje, koliko debel potrebuje za gradnjo posameznega mostu med otokoma in koliko debel je na razpolago na posameznem otoku.
- Vsako deblo se lahko uporabi samo enkrat, ni pa potrebno uporabiti vseh.



Najmanj koliko debel potrebuje Božo, da zgradi mostove med otoki, da bodo lahko učenci šli po suhem v novo šolo?

Rešitev

Božo potrebuje najmanj 14 debel.



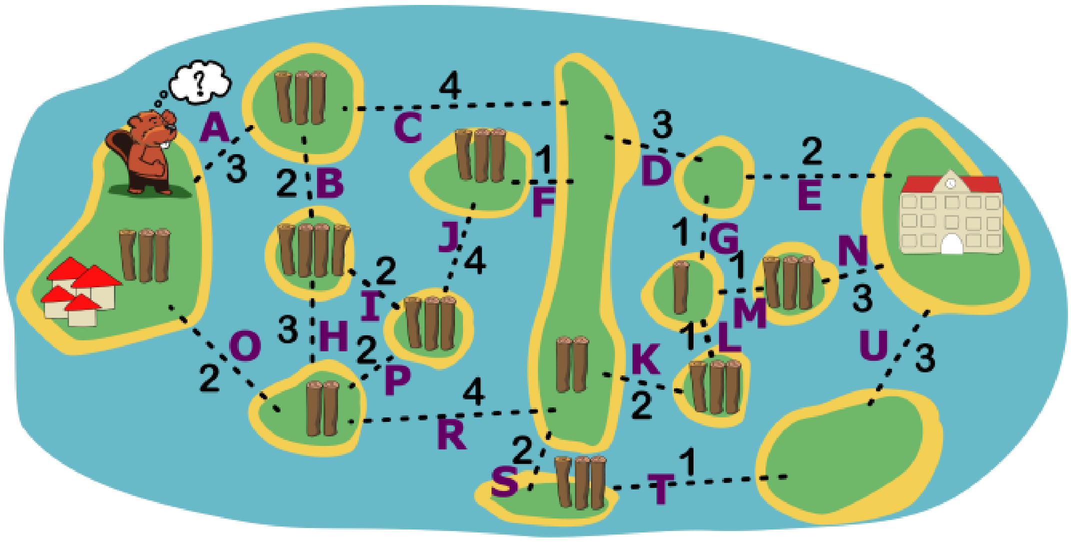
Rešitev je prikazana na zgornji sliki. Da je prava pot izvedljiva (to je, da na vsakem koraku velja, da je število pobranih debel večje ali enako številu uporabljenih debel), lahko pokažemo z zapisovanjem vsote pobranih debel in vsote uporabljanih debel na vsakem koraku: (3,2)(5,4)(8,4)(8,8)(10,10)(13,11)(14,12)(14,14).

Da bi dokazali, da je ta pot najugodnejša, moramo preveriti, če obstaja še kakšna pot, ki povezuje vas in šolo in je zanjo potrebno uporabiti manj kot 14 debel. In seveda, da je taka pot izvedljiva (to je, da imamo vedno dovolj debel za gradnjo mostov).

Najprej ugotovimo, da vse poti vodijo preko sredinskega, velikega otoka. Zato lahko naš problem razdelimo na dva dela: pot od vasi do tega otoka (leva stran) in pot od otoka do šole (desna stran). Ugotovimo lahko, da za pot iz vasi do otoka na sredini potrebujemo vsaj 6 debel in za pot od sredinskega otoka do šole še najmanj dodatnih 5 debel (to imenujemo spodnja meja).

Najprej rešimo levo stran z uporabo desne spodnje meje. To pomeni, da poiščemo vse možne poti na levi strani, ki porabijo 9 debel ali manj (ker bi katerakoli rešitev za levo stran z 10 debli ali več, prinesla rešitev večjo od 14 debel, saj za desno stran potrebujemo najmanj 5 debel).

Na sliki označimo poti s črkami abecede.



Na levi strani imamo naslednje poti, ki uporabijo 9 debel ali manj (prikazano s parom število vzetih debel, uporabljenih debel):

{A,C}=(6,7), {A,B,C}=(10,9), {O,R}=(5,6), {O,H,R}=(9,9), {O,P,R}=(8,8), {O,P,J,F}=(11,9)

Če odstranimo neizvedljive rešitve (tiste, kjer Božo pobere manj debel, kot jih uporabi), nam ostane:

{A,B,C}=(10,9), {O,H,R}=(9,9), {O,P,R}=(8,8), {O,P,J,F}=(11,9)

Vidimo, da je spodnja meja za levo stran enaka 8 debel (v primeru poti {O,P,R}).

Nato rešimo še desno stran z uporabo spodnje meje za levo stran in iščemo poti, kjer uporabimo 6 debel ali manj:

{D,E}=(2,5), {D,G,E}=(3,6), {K,L,G,E}=(6,6), {S,T,U}=(5,6)

Tudi tu odstranimo neizvedljive rešitve; ostane nam le {K,L,G,E}=(6,6).

Na koncu kombiniramo rešitve leve in desne strani, da najdemo izvedljive pare, ki imajo 14 ali manj skupno porabljenih debel. Najdemo le eno rešitev: {O,P,R}=(8,8) + {K,L,G,E}=(6,6) → (14,14), ker vse druge rešitve za levo stran zahtevajo 9 debel, kar bi skupaj prineslo porabo 15 debel (kar pa ni najmanjše potrebno število debel).

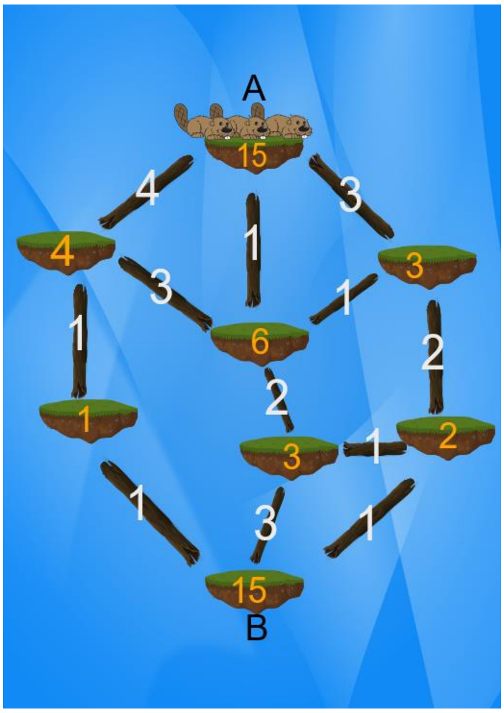
Računalniško ozadje

Otoke in mostove lahko predstavimo kot vozlišča in poti v grafu. Število debel, pobranih na otoku in uporabljenih pri gradnji, lahko povežemo z otoki in mostovi. Pot, ki jo prepotuje Božo, lahko predstavimo s seznamom poti, ki jih zgradi Božo, pri čemer je vrstni red v seznamu enak vrstnemu redu gradnje mostov. V naši rešitvi smo problem rešili z metodo deli in vladaj.

Premiki po otokih

srednja šola

Na otoku A je 15 bobrov, ki želijo iti na zabavo na otok B.



Vsi ostali otoki so prazni. Ker so se bobri za na zabavo pražnje oblekli, se ne želijo zmočiti in so se odločili do otoka B oditi peš. Bobri hodijo med otoki po hlodih, pri tem pa upoštevajo naslednja pravila:

- Številka na hlodu nam pove, največ koliko bobrov je lahko sočasno na tem hlodu.
- Številka na otoku nam pove, največ koliko bobrov je lahko hkrati na tem otoku.
- Na otoku A in B je lahko hkrati največ 15 bobrov.
- Skupine bobrov potujejo med otokoma A in B v več korakih. Na primer, v prvem koraku grejo lahko z otoka A na levi otok 4 bobri, en bober na sredinski otok in trije bobri na desni otok.
- Šele ko so vsi bobri, ki se premikajo v prvem koraku, na naslednjem otoku, se lahko začne drugi korak. Lahko se seveda odločijo, da se nekaj ali pa tudi vsi bobri v nekem koraku ne premaknejo nikamor in ostanejo na otoku.

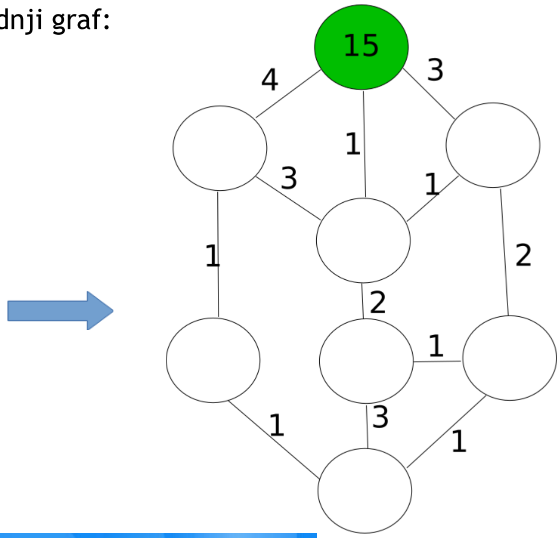
Kakšno je največje število bobrov, ki so lahko po petem koraku na otoku B?

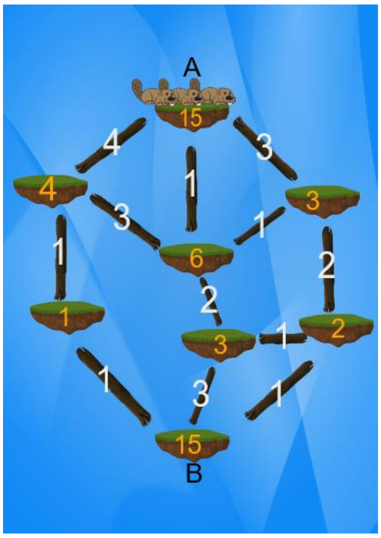
Rešitev

Po petem koraku je lahko na otoku B največ 13 bobrov.

Za lažjo rešitev naloge je smiselno sliko najprej nekoliko poenostaviti. Otoke na sliki predstavljajo krogi, v katerih je zapisano, koliko bobrov je na njih, povezave med njimi pa hlode, na katerih je zapisana njihova največja nosilnost. Za lažjo razlago bomo »nosilnost« otokov iz slike izpustili, upoštevali pa jo bomo pri reševanju.

Začetno sliko bi tako prerisali v naslednji graf:



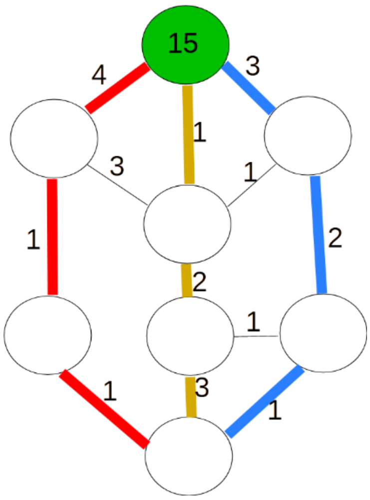


Vidimo lahko, da so za pot od otoka A do otoka B potrebni vsaj 3 koraki. To lahko bobri naredijo po treh različnih poteh:

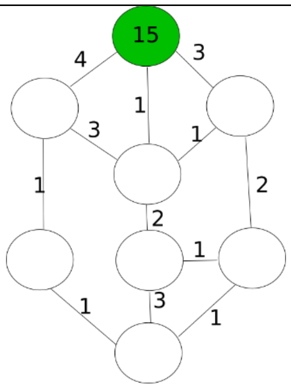
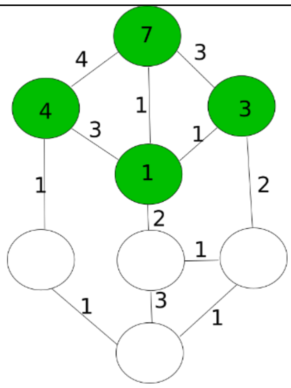
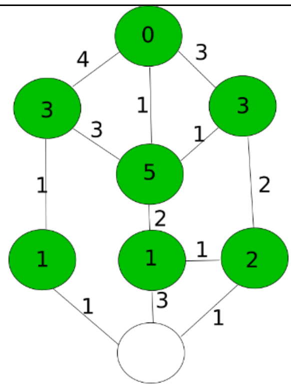
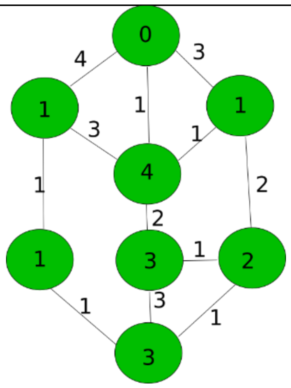
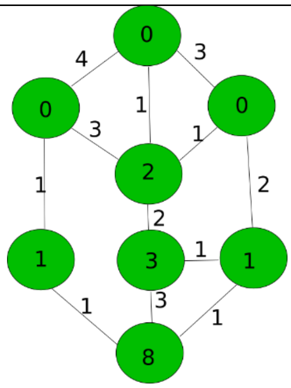
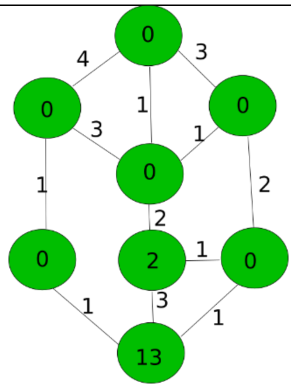
- Če bobri sledijo levi (rdeči) poti, grejo lahko štirje na prvi otok, eden gre na drugi otok in eden na otok B.
- Če bobri sledijo sredinski (rumeni) poti, gre lahko eden na prvi otok, nato na drugi in na otok B.
- Če bobri sledijo desni (modri) poti, grejo lahko trije na prvi otok, dva na drugi otok in en na otok B.

V treh korakih lahko tako otok B dosežejo največ trije bobri.

Ker hlodi, ki so povezani z otokom B lahko nosijo hkrati 1+3+1=5 bobrov, je največje število bobrov, ki lahko pridejo v enem koraku na otok B enako 5. Iz tega razloga je nemogoče, da bi lahko na otok po petih korakih prišlo več kot 13 bobrov (3+5+5).



Zanima nas, ali je mogoče, da jih na otok B v petih korakih res pride 13 ali je ta številka še nižja. Pa poglejmo:

Začetek	Po 1. koraku	Po 2. koraku
		
Po 3. koraku	Po 4. koraku	Po 5. koraku
		

Iz postopka reševanja vidimo, da res lahko na otok B v petih korakih pride 13 bobrov.

Računalniško ozadje

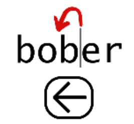
V nalogi je predstavljen graf z utežmi tako na vozliščih kot tudi na povezavah med njimi. Potrebno je bilo poiskati najbolj optimalen pretok (bobrov) skozi graf. Za iskanje maksimalnega pretoka skozi graf lahko na podobnih primerih uporabimo Ford–Fulkersonov algoritem.

Urejevalnik besedil

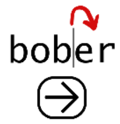
srednja šola

Enostaven urejevalnik besedil ob pritisku na tipko omogoča vstavljanje znaka levo od kazalca (na slikah je prikazan kot pokončna črta) in uporabo naslednjih šestih posebnih tipk:

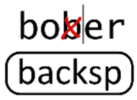
tipka »puščica levo« premakne kazalec za en znak v levo



tipka »puščica desno« premakne kazalec za en znak v desno

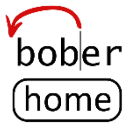


tipka »backsp« izbriše znak levo od kazalca

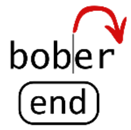


tipka »del« izbriše znak desno od kazalca

tipka »home« prestavi kazalec na začetek besedila



tipka »end« prestavi kazalec na konec besedila



V urejevalniku imamo zapisano besedilo `abcde*ghij***` in bi želeli vse znake * zamenjati s črko, da dobimo besedilo `abcdefghijklm`. Kazalec se nahaja na koncu besedila.

`abcde*ghij***|` ⟶ `abcdefghijklm`

Najmanj koliko pritiskov na tipke moraš narediti za tak popravek?

Rešitev

Potrebujemo najmanj 13 pritiskov na tipke.

Zamenjati moramo znake pri štirih zvezdicah. Vsako zvezdico moramo pobrisati (ena tipka) in namesto nje zapisati ustrezno črko (ena tipka). To zahteva 8 pritiskov na tipke. Ker pa se ena zvezdica ne nahaja pri kazalcu, se moramo še premakniti do te zvezdice, kar zahteva premik preko 7 znakov, torej še 7 pritiskov na tipke (to je v primeru, ko najprej zamenjamo tri končne zvezdice in se nato premaknemo na zvezdico v sredini). Skupaj torej uporabimo 15 tipk. Ker pa imamo na voljo tudi tipki »home« in »end«, lahko število potrebnih pritiskov na tipke še zmanjšamo, če najprej zbrišemo vse končne zvezdice, nato se premaknemo do srednje zvezdice in jo zamenjamo, potem pa naredimo premik na konec z eno samo tipko in dodamo še tri manjkajoče črke. Ta postopek zahteva le 13 pritiskov na tipke.

Ali bi lahko zamenjavo izvedli le z 12 pritiski na tipke? Osem tipk moramo pritisniti, da vse štiri zvezdice zamenjamo s črkami. Ker do zvezdice v sredini ne moremo priti z drugimi tipkami kot puščica levo oz. desno, je najmanjše število tipk za premik do te zvezdice štiri, saj so med obema skupinama zvezdic štiri črke, ki jih moramo preskočiti. Torej skupaj 12 pritiskov na tipke. Vendar pa se moramo med obema skupinama zvezdic premakniti dvakrat (končne zvezdice najprej zbrišemo, nato pa se moramo še enkrat vrniti na konec, da dopišemo ustrezne črke), zato potrebujemo še najmanj eno tipko.

Zaporedje tipk je torej naslednje: backsp, backsp, backsp, levo, levo, levo, levo, backsp, f, end, k, l, m.

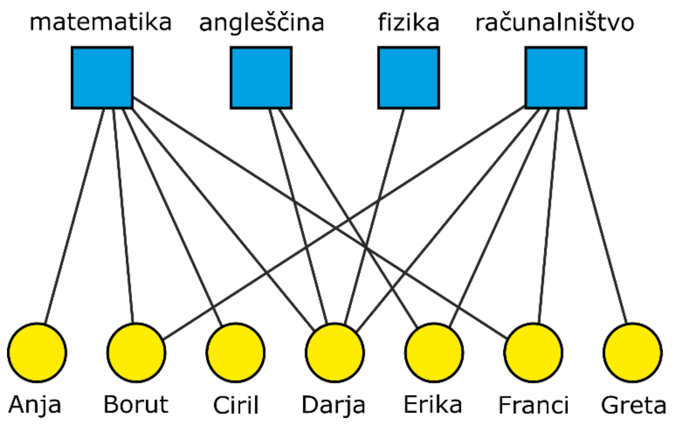
Računalniško ozadje

Tudi pri vsakodnevnih enostavnih nalogah, kot je uporaba urejevalnika besedil, se srečamo z različnimi računalniškimi koncepti. Zaporedje korakov, ki smo jih morali narediti v nalogi, smo zapisali kot zaporedje pritiskov na tipke, ki izgleda podobno kot programska koda v nekem enostavnem programskem jeziku. Rešitev v nalogi predstavljenega problema je veliko, mi pa smo morali poiskati optimalno rešitev, ki zahteva kar najmanj korakov (oziroma pritiskov na tipke).

Novi učitelji

srednja šola

Anja, Borut, Ciril, Darja, Erika, Franci in Greta so se prijavili na razpis za nove učitelje na šoli. Šola išče učitelje matematike, fizike, računalništva in angleščine. Slika prikazuje, kdo od prijavljenih izpolnjuje pogoje za poučevanje posameznih predmetov.



Šola bo za vsak predmet najela le eno osebo, vsaka oseba pa lahko na šoli poučuje le en predmet.

Katera od trditev je zagotovo resnična?

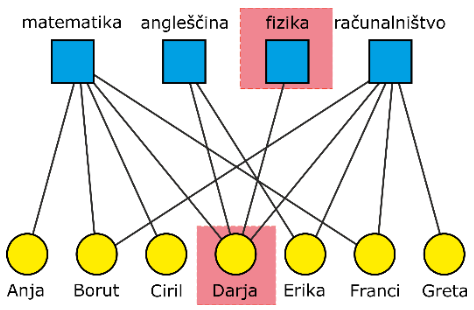
A) Greta bo poučevala računalništvo.
B) Erika bo poučevala angleščino.
C) Darja bo poučevala računalništvo.
D) Če bo Borut poučeval računalništvo, bo Franci poučeval matematiko.
E) Ciril bo poučeval matematiko.

Rešitev

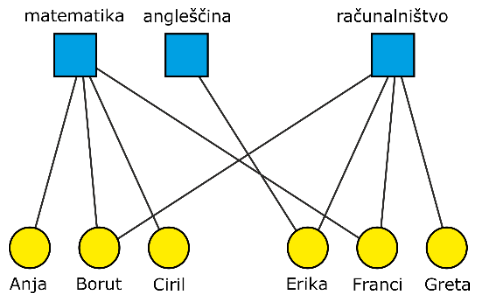
Pravilen odgovor je B: Erika bo poučevala angleščino.

Če za poučevanje nekega predmeta izpolnjuje pogoje le ena prijavljena oseba, bo šola najela to osebo za ta predmet, četudi ta oseba izpolnjuje pogoje tudi za poučevanje drugih predmetov.

Tako lahko hitro vidimo, da je za poučevanje fizike ustrezna le Darja. Torej je odgovor C napačen.

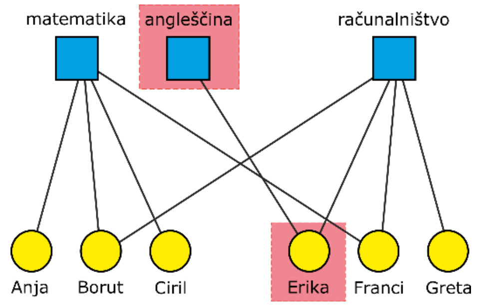


Če Darja dobi fiziko, lahko tako Darjo kot fiziko odstranimo iz grafa, saj je mesto učitelja fizike zapolnjeno in Darja lahko na šoli uči le en predmet.

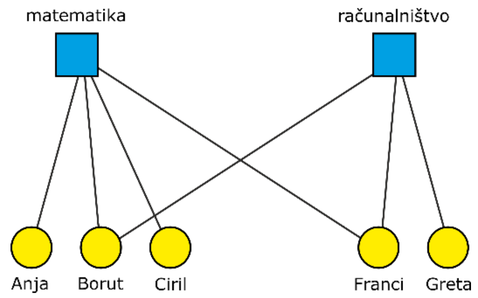


Podobno sklepamo še naprej: za poučevanje angleščine je sedaj primerna le ena oseba, to je Erika. Torej je trditev B pravilna.

Vendar, ali je lahko pravilna še katera druga trditev?

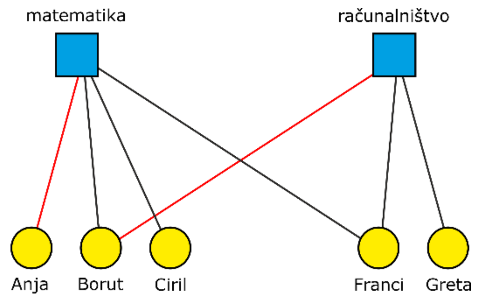


Ker bo Erika učila angleščino, lahko tudi Eriko in predmet angleščina odstranimo iz grafa. Za preostala dva predmeta, matematiko in računalništvo, imamo tako več primernih kandidatov, zato ne moremo z gotovostjo vedeti, katere kandidate bo izbrala šola.



Hitro lahko najdemo rešitev, ki ni skladna z odgovori A, D in E. Na primer, Anja je izbrana za poučevanje matematike, Borut pa za poučevanje računalništva (označeno rdeče v grafu na desni).

Torej je edina zagotovo resnična trditev le B.



Računalniško ozadje

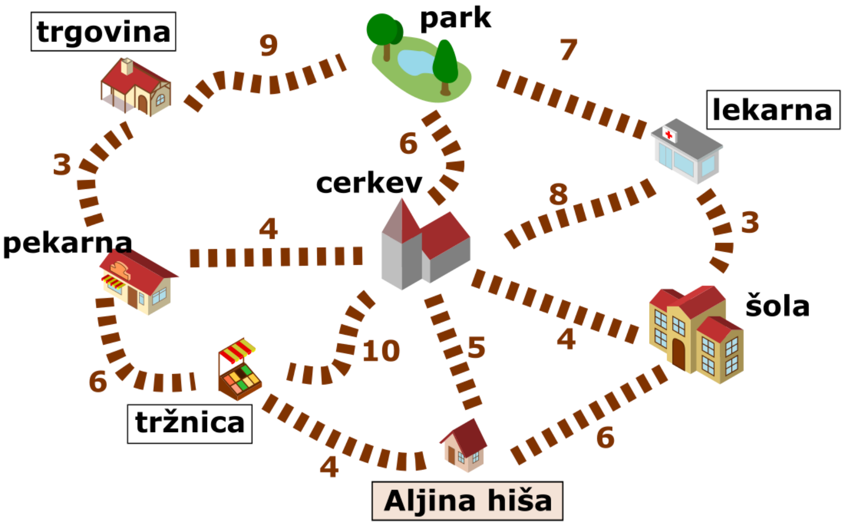
Pri reševanju tega problema smo morali upoštevati podane omejitve (kdo lahko poučuje nek predmet) in poiskati tako rešitev, ki zadošča vsem omejitvam. Ko smo pri reševanju problema neki spremenljivki določili vrednost (na primer Darji dodelili fiziko), smo hkrati tudi zožili druge možnosti za rešitev. S podobnimi problemi se srečamo pri dodeljevanju virov (na primer pri sestavi urnika) ali pa pri ugankah (na primer Sudoku).

Pri rešitvi smo si pomagali z *dvodelnim grafom*, s katerim smo predstavili podatke o predmetih in prijavljenih učiteljih. Dvodelni graf je vrsta grafa, kjer imamo dve skupini vozlišč, vsaka povezava pa povezuje vozlišči iz različnih skupin.

Aljini opravki

srednja šola

Zemljevid prikazuje Aljino vas. Številke ob poteh povejo, koliko časa (v minutah) potrebuje Alja za pot med dvema krajema.



Danes ima Alja tri opravke: v trgovini mora kupiti sok, v lekarno mora po zdravila ter na tržnico po sadje. Vrstni red opravkov ni pomemben (lahko jih opravi v katerem koli vrstnem redu), za vsak opravek pa potrebuje eno minuto. Alja se odpravi iz svoje hiše in se po končanih opravkih vrne nazaj v hišo.

Alja za vse opravke potrebuje 41 minut, če se odpravi po opravkih po naslednji poti:

> Aljina hiša → tržnica (sadje) → pekarna → trgovina (sok) → park → lekarna (zdravila) → šola → Aljina hiša

Za samo pot porabi 38 minut (4 + 6 + 3 + 9 + 7 + 3 + 6) in še 3 minute za vse tri opravke.

Kakšen je najkrajši čas, v katerem lahko Alja opravi vse opravke (in se tudi vrne domov)?

Rešitev

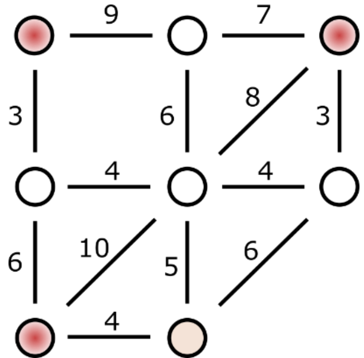
Najkrajši čas za vse tri opravke je 39 minut.

Pot, ki terja najmanj časa je naslednja:

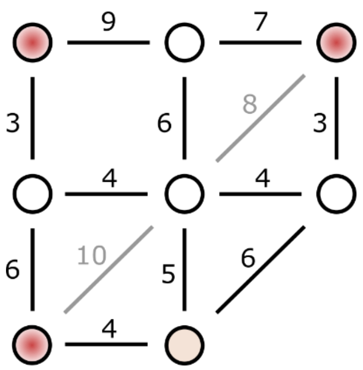
> Aljina hiša → šola → lekarna (zdravila) → šola → cerkev → pekarna → trgovina (sok) → pekarna → tržnica (sadje) → Aljina hiša

Za to pot potrebuje 6 + 3 + 3 + 4 + 4 + 3 + 3 + 6 + 4 = 36 minut ter še 3 minute za vse tri opravke, torej skupaj 39 minut. To pot lahko opravi tudi v nasprotni smeri in zanjo porabi enako časa.

Rešitev lahko poiščemo s pomočjo grafa in tudi pokažemo, da ne obstaja krajša pot. Krogi v grafu (vozlišča) predstavljajo

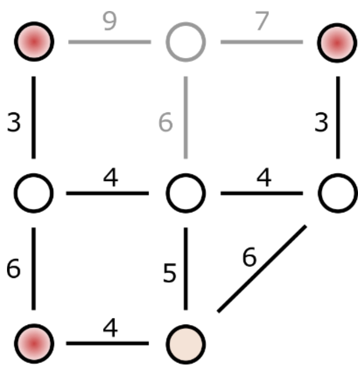


kraje v Aljini vasi, črte (povezave) pa poti med temi kraji. Tudi tu so povezavam pripisane številke, ki določajo čas, potreben za to pot. Z rdečo smo označili kraje oz. vozlišča, ki jih mora Alja obiskati, označena pa je tudi Aljina hiša kot začetek in konec poti.

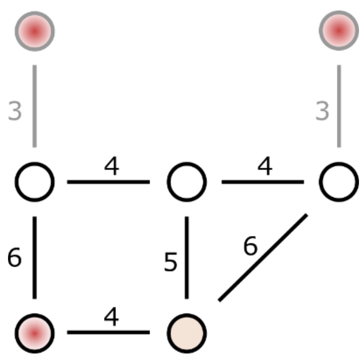


Povezavi, ki sta sivi (glej levo sliko), lahko zanemarimo, saj terjata preveč časa in lahko med vozliščema, ki ju taka povezava povezuje, pridemo hitreje naokoli preko vmesnega vozlišča.

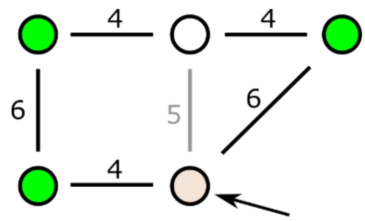
Zato ti dve povezavi odstranimo iz grafa.



Prav tako lahko odstranimo tudi vozlišče oz. kraj »park«, saj Alja tam nima opravkov, za vsako pot, ki gre skozi park, pa obstaja hitrejša alternativa.



Zgornji dve vozlišči (trgovina in lekarna) morata ostati na Aljini poti. Vsakega od njiju lahko dosežemo iz vozlišča tik pod njim in za to porabimo 3 minute tja in 3 minute nazaj (to velja za obe vozlišči). Zato lahko iz grafa odstranimo tudi ti dve vozlišči, a moramo končni rešitvi prišteti še 12 minut za obisk obeh vozlišč (po 6 minut za vsakega od njiju).



Tako na koncu dobimo graf, ki ga prikazuje slika na desni. Pot moramo začeti in končati v spodnjem desnem vozlišču (na katerega kaže puščica) ter obiskati vsa tri zeleno obarvana vozlišča. Najkrajša pot gre skozi vseh pet vozlišč, pri tem pa izpustimo sivo pobarvano povezavo.

Za to pot potrebujemo 4 + 6 + 4 + 4 + 6 = 24 minut. Temu prištejemo še 12 minut (za obisk trgovine in lekarne; glej razlago pri predhodnem grafu) in 3 minute za vse tri opravke, skupaj torej 39 minut.

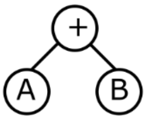
Računalniško ozadje

Pri reševanju problema smo zemljevid Aljine vasi poenostavili in zapisali v obliki *uteženega grafa*. V računalništvu se veliko problemov lahko reši tako, da jih predstavimo z grafom in poiščemo določene lastnosti takega grafa. V naši nalogi smo morali poiskati najkrajši cikel, ki vključuje nekaj vnaprej določenih vozlišč. S podobnimi problemi se ukvarja posebna matematična in računalniška disciplina, ki jo imenujemo *teorija grafov*.

Matematični izrazi

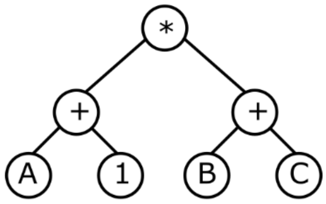
srednja šola

Matematične izraze z operacijami seštevanje (+), odštevanje (-), množenje (*) in deljenje (/) lahko zapišemo tudi na nekoliko drugačen način in predstavimo grafično na naslednji način (A, B in C so spremenljivke, ki nastopajo v izrazih):



Leva slika prikazuje izraz **A + B** , ki ga lahko zapišemo kot:

A B +

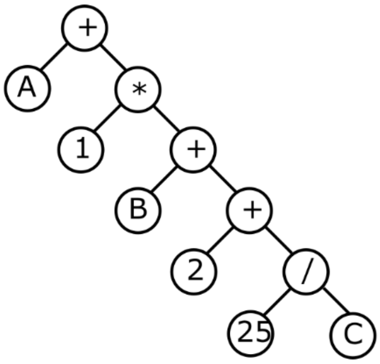


Izraz **(A + 1) * (B + C)** na desni sliki pa zapišemo kot:

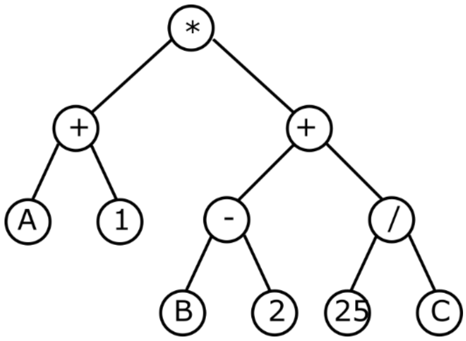
A 1 + B C + *

Katera slika prikazuje izraz, zapisan kot **A 1 + B 2 + * 25 C / +** ?

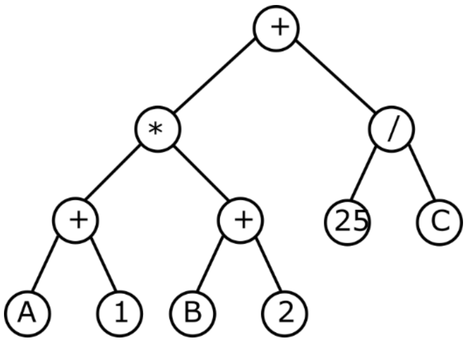
A)



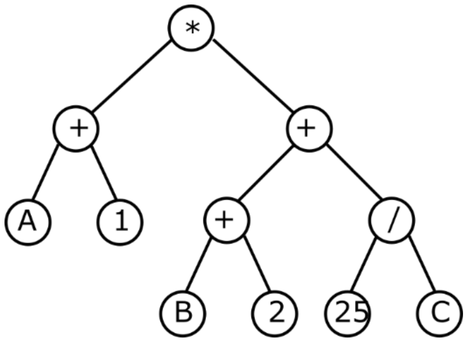
B)



C)



D)

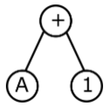


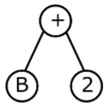
Rešitev

Pravilen odgovor je C.

Sliko izraza lahko sestavimo po korakih.

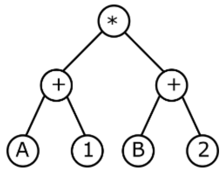
Zapis **A 1 +** grafično prikažemo kot:



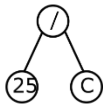


Zapis **B 2 +** predstavimo z:

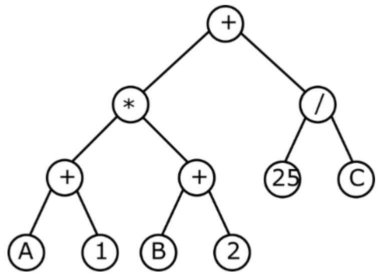
Če oba izraza pomnožimo, dobimo izraz **A 1 + B 2 + ***, ki ga grafično prikažemo z grafom na desni.



Zapis **25 C /** grafično predstavimo z:



Če zadnja dva izraza seštejemo, dobimo zapis **A 1 + B 2 + * 25 C / +**, ki je grafično predstavljen s sliko na desni.



Odgovor A ni pravilen, saj ustreza izrazu **A 1 B 2 25 C / + + * +**.

Odgovor B tudi ni pravilen, ker vsebuje operacijo odštevanja (-), ki pa se ne pojavi v iskanem izrazu.

Odgovor D pa je napačen, ker ustreza izrazu **A 1 + B 2 + 25 C / + ***.

Računalniško ozadje

Obrnjeni poljski zapis ali *postfiksni zapis* (angleško *reverse Polish notation* ali krajše RPN) je način zapisa aritmetičnih izrazov, ki ga uporabljajo prevajalniki (ti pretvorijo kodo programskega jezika, ki jo razumejo ljudje, v strojno kodo, ki jo razumejo računalniki). Uporablja se za vrednotenje izrazov v programu. Pri obrnjenem poljskem zapisu operator sledi operandom, s tem pa odpravimo tudi potrebo po oklepajih. Izraze lahko predstavimo z binarnim drevesom, kot smo to naredili tudi v naši nalogi.

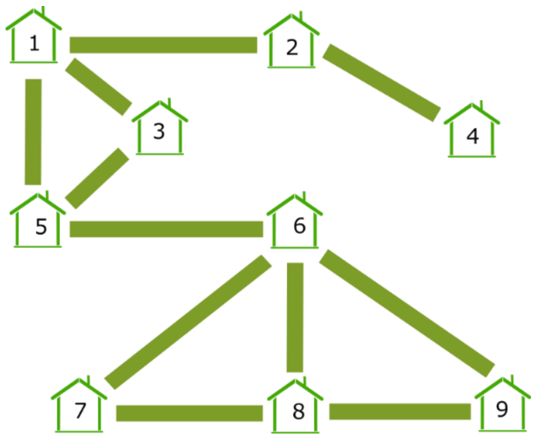
Pisane hiške

V Bobrovi vasi so zgradili 9 novih hišk, ki jih bodo razdelili med navijače treh nogometnih klubov. Hiške bodo dobili po trije navijači iz vsakega kluba. Vsak od njih bo hiško pobarval v barvah svojega kluba: rumeno, modro ali rdeče.

Da bi se v vasi izognili pretiranemu druženju navijačev istega kluba (in morebitnim izgredom), bodo hiške razdelili tako, da dva navijača iz istega kluba ne bosta dobila hišk, ki sta neposredno povezani s cesto.

Slika na desni prikazuje lokacije hišk in ceste, ki jih povezujejo.

Katere hiške morajo biti zagotovo enake barve?



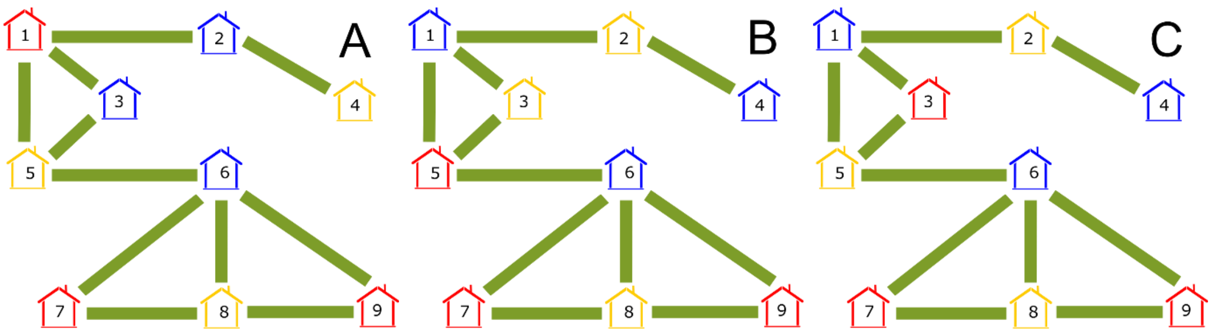
Rešitev

Pravilen odgovor je hiški 7 in 9.

Hiški 6 in 8 sta povezani, zato morata biti različnih barv. Hiški 7 in 9 pa sta povezani z obema hiškama (6 in 8), zato morata biti obe tretje barve, ki še preostane. Torej sta zagotovo enake barve.

Sedaj pa še pokažimo, da imajo vsi ostali pari hišk lahko različne barve. Za hiške 6, 7, 8 in 9 smo že ugotovili, kako jih moramo pobarvati s tremi barvami: par hišk (7, 9) mora biti enake barve, za preostali dve hiški pa uporabimo preostali dve barvi. Torej noben par hišk (razen 7 in 9) nima enake barve.

Recimo, da je hiška 6 modra, 8 rumena ter 7 in 9 rdeči. Preostale hiške pobarvamo na tri različne načine, kot prikazujejo slike A, B in C.



Najprej se osredotočimo le na hiške 1 do 5, saj smo za hiške 6 do 9 že določili barve. Pri barvanju A imamo iste barve le para hišk (2, 3) in (4, 5); oba para hišk pa sta pobarvana v različnih barvah pri barvanju C.

Sedaj poglejmo še pare, kjer je ena hiška iz skupine 1 do 5, druga pa iz skupine 6 do 9. Vse hiške 1 do 5 imajo pri barvanju B različne barve kot pri barvanju A. Hiške 6 do 9 pa so pri obeh barvanjih enake barve. To pomeni, da bodo pari teh hišk imeli različne barve ali pri barvanju A ali pa pri B. Tako smo pokazali, da so z izjemo para (7, 9), ki je tudi naša rešitev, vsi drugi pari hišk različnih barv vsaj pri enem načinu barvanja. Torej ne obstaja noben drug par hišk, ki bi zagotovo moral biti enake barve.

Računalniško ozadje

Tudi pri tej nalogi smo uporabili graf, s katerim smo prikazali poenostavljen zemljevid Bobrove vasi. Graf je množica vozlišč (v našem primeru hiške), ki so povezana s povezavami (v našem primeru ceste med hiškami). Problem, ki smo ga reševali v nalogi, je različica problema barvanja grafov: kako pobarvati vozlišča grafa tako, da nobeni dve sosednji vozlišči nista enake barve. V splošnem je to za računalnik težek problem, saj zanj ne obstaja učinkovit algoritem.

BoberGPT

srednja šola

BoberGPT je klepetalni robot, ki zna tvoriti stavke s tremi besedami. Jezikovni model izbira besede eno za drugo. Spodnji tabeli prikazujeta verjetnosti, da se jezikovni model odloči za določeno naslednjo besedo glede na predhodne besede.

Verjetnosti za drugo besedo ob podani prvi besedi:

Prva beseda	radi	neradi
Kenguruji	0,7	0,3
Bobri	0,6	0,4

Verjetnosti za tretjo besedo ob podanih prvih dveh besedah:

Prvi dve besedi	plavajo	skačejo
Kenguruji radi	0,2	0,8
Kenguruji neradi	0,9	0,1
Bobri radi	0,7	0,3
Bobri neradi	0,1	0,9

Na primer, če se stavek začne z besedo »Kenguruji«, je verjetnost, da model tvori stavek »Kenguruji radi skačejo.«, enaka 56 % (0,56). To sledi iz obeh tabel, ker (1) je verjetnost, da je druga beseda »radi«, če je prva beseda »Kenguruji«, enaka 0,7; (2) če sta prvi dve besedi »Kenguruji radi«, je verjetnost naslednje besede »skačejo« enaka 0,8; (3) model napoveduje besede eno za drugo, zato je verjetnost celega stavka enaka 0,7 * 0,8 = 0,56.

Če se stavek začne z besedo »Bobri«, katerega od navedenih stavkov bo najbolj verjetno tvoril jezikovni model?

A) »Bobri neradi plavajo.«
B) »Bobri neradi skačejo.«
C) »Bobri radi plavajo.«
D) »Bobri radi skačejo.«

Rešitev

Pravilen odgovor je C: »Bobri radi plavajo.«

Do rešitve lahko pridemo tako, da glede na podani tabeli izračunamo verjetnosti za vsakega od navedenih stavkov:

»Bobri neradi plavajo.« ima verjetnost 0,4 * 0,1 = 0,04.

»Bobri neradi skačejo.« ima verjetnost 0,4 * 0,9 = 0,36.

»Bobri radi plavajo.« ima verjetnost 0,6 * 0,7 = 0,42. Ta verjetnost je največja.

»Bobri radi skačejo.« ima verjetnost 0,6 * 0,3 = 0,18.

Do rešitve pa lahko pridemo tudi tako, da pri podani začetni besedi v tabeli verjetnosti za vsako naslednjo besedo poiščemo tisto, ki ima največjo verjetnost. Če stavek začnemo z besedo »Bobri«, tej besedi z največjo verjetnostjo (0,6) sledi beseda »radi« (to razberemo iz prve tabele). In ko imamo dve zaporedni besedi »Bobri radi«, tema z največjo verjetnostjo (0,7) sledi beseda »plavajo« (kar razberemo iz druge tabele). Torej je »Bobri radi plavajo.« najbolj verjeten stavek, ki ga tvori naš jezikovni model.

Računalniško ozadje

V nalogi je poenostavljeno prikazano delovanje umetne inteligence z uporabo jezikovnega modela za generiranje besedila. To je verjetnostni model, kjer vsakokratni rezultat modela temelji na verjetnosti (te verjetnosti so vnaprej izračunane na podlagi velike količine podatkov, na katerih je model naučen). V nalogi opisan model ustvarja po eno besedo naenkrat, ta pa je odvisna od celega zaporedja predhodno ustvarjenih besed. Podobno deluje tudi ChatGPT, a seveda na podlagi veliko večjega modela.

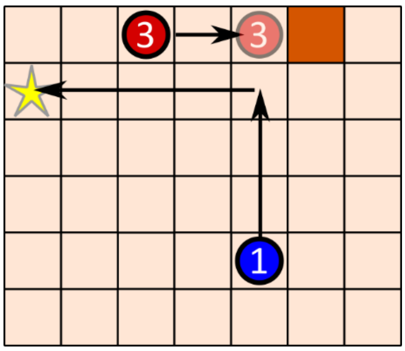
Zaletavanje robotov

srednja šola

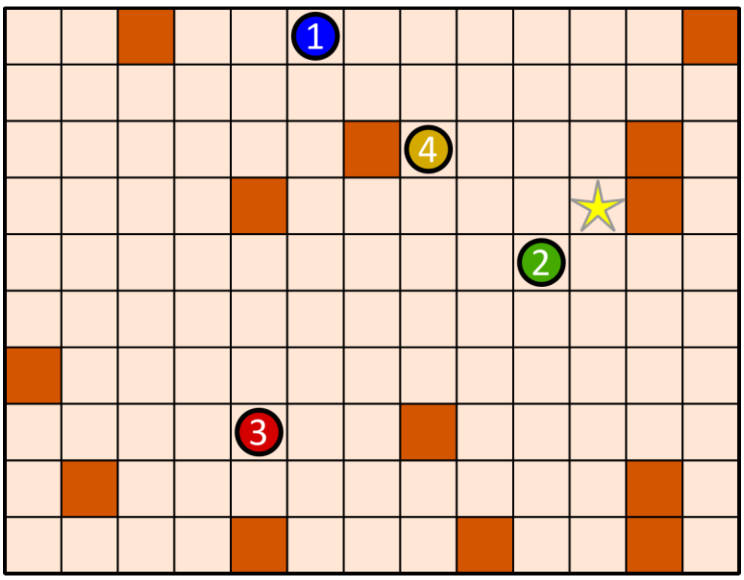
Robotu želimo podati taka navodila, da se premakne s trenutnega kvadratka na kvadratek z zvezdo ☆. Pri tem moramo upoštevati naslednja pravila igre:

1. Rjavi kvadratki ■ so ovire. Ovira je nekaj, kar zaustavi robota, da ne more nadaljevati poti v isti smeri (robot ne more iti skozi oviro).
2. Tudi ostali roboti so ovire.
3. Robotu lahko podamo ukaze za premike v štirih smereh: gor (↑), dol (↓), levo (←) in desno (→).
4. Ko se robot začne premikati v neko smer, se ne ustavi, dokler ne zadane ob oviro ali pride do roba igralne površine.
5. Ko se robot začne premikati, ostali roboti počakajo, da se ustavi (tj. naenkrat se lahko premika le en robot).

V primeru na desni lahko premaknemo robota 1 na zvezdo s kombinacijo treh ukazov. Najprej premaknemo robota 3 na desno. Robot se premakne v desno do ovire in se tam ustavi, kot je prikazano na sliki. Nato premaknemo robota 1 gor in pri tem kot oviro izkoristimo ustavljenega robota 3 (in s tem ustavimo premikanje robota 1). Na koncu robotu 1 pošljemo še ukaz za premik v levo, da se premakne do zvezde, kjer se ustavi.



Če imamo igralno površino na spodnji sliki, kakšno je najmanjše število ukazov, ki jih podamo robotom, da se robot 1 premakne in ustavi pri zvezdi ☆?

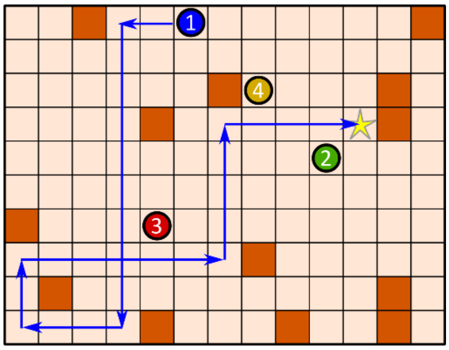


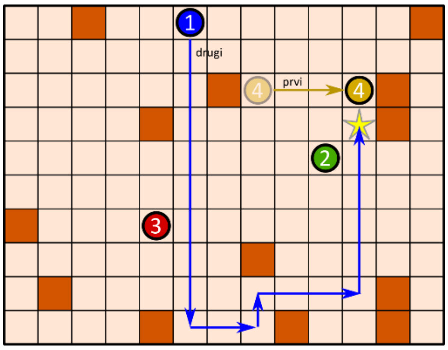
Rešitev

Robotu moramo podati najmanj 4 ukaze.

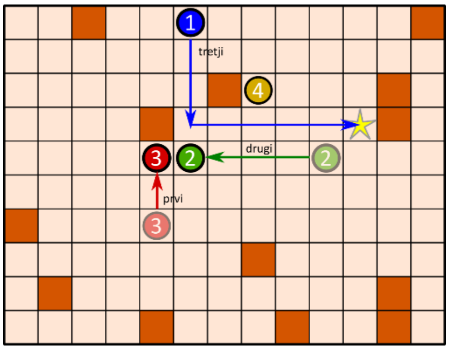
Za najkrajšo rešitev morajo roboti sodelovati. Poglejmo najprej, kakšna je rešitev, če premikamo le robota 1 (torej brez medsebojnega sodelovanja robotov).

Če uporabimo le robota 1, ga lahko najprej usmerimo v levo, nato dol, levo, gor, desno, gor in levo. Taka pot zahteva 7 ukazov. Na sliki je prikazana z modrimi puščicami (vsaka puščica ustreza enemu ukazu in premiku robota).



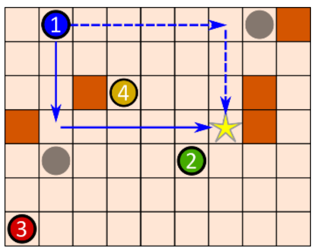


Če roboti sodelujejo, lahko najdemo krajšo rešitev. Tako lahko robota 4 postavimo kot oviro robotu 1 in s tem skrajšamo nalogo na 6 ukazov. Najprej robota 4 pošljemo desno do prve ovire, nato pa robotu 1 pošljemo ukaze: dol, desno, gor, desno, gor. Skupaj imamo torej 6 ukazov.



Če v rešitev vključimo še več robotov, lahko dobimo rešitev, ki zahteva le 4 ukaze. Najprej premaknemo gor robota 3 in tako pripravimo oviro za robota 2. Nato premaknemo robota 2 levo. Ko se robot 2 ustavi, ga uporabimo kot oviro za robota 1. Ta potrebuje le dva premika do zvezde; pošljemo mu ukaza dol in desno. Skupaj smo uporabili tri robote in le 4 ukaze.

Ali lahko dobimo rešitev z manj kot 4 ukazi? Robota 1 nikakor ni možno premakniti do zvezde le z enim ukazom. Če uporabimo dva ukaza, imamo dva načina, kako pripeljemo robota do zvezde (glej desno sliko). Vendar bi se v tem primeri moral robot 1 ustaviti pred kvadratkom, ki je označen s sivo piko, da ga preusmerimo proti zvezdi. Ker na nobenem od obeh kvadratkov s sivo piko na začetni igralni površini ni ovire, jo moramo tja postaviti, preden podamo ukaze robotu 1. Torej moramo najprej postaviti drugega robota na kvadratek s sivo piko, tako da se robot 1 ustavi, ko se zaleti vanj. To pomeni, da potrebujemo najmanj tri ukaze (dva za robota 1 in najmanj enega za drugega robota). Pa lahko katerega od drugih robotov postavimo na eno od obeh sivih pik z le enim ukazom? Ne, to ne bo šlo. Robota 3 in 4 nista v istem stolpcu/vrstici kot sivi piki, robota 2 pa sicer lahko pošljemo levo, a se ne more ustaviti na sivi piki. To pomeni, da z manj kot 4 ukazi ne moremo priti do rešitve.



Verjetno si opazil, da zgoraj opisana rešitev s 4 ukazi ni edina možna. In tudi ni potrebno, da v najkrajši rešitvi uporabimo več kot enega robota pomočnika. A za odgovor na vprašanje v nalogi to niti ni pomembno.

Računalniško ozadje

Sodelovanje je zelo pomembno za učinkovito delovanje tako robotov kot tudi informacijskih sistemov. Pri programiranju robotov moramo poskrbeti, da zagotovimo njihovo usklajeno delovanje ter se s tem izognemo nevarnostim poškodovanja robotov ali celo ljudi. Vendar pa je sodelovanje več robotov tudi zahtevna optimizacijska naloga.

Pregled nalog

Šolsko tekmovanje

Naloga	Država	2.	3.	4.	5.	6.	7.	8.	9.	SŠ	Str.
Sosedje	Filipini	•	•								
Rože	Romunija	•	•								
Žoge	Saudova Arabija	•	•								
Dežnik	Švica	•	•								
Jabolka	Češka	•	•	•							
Slike s poti 1	Japonska		•	•							
Ogrlica	Slovaška		•	•	•						
Hamburger	Japonska		•	•	•						
Železnica	Češka			•							
Rikolini	Kanada			•							
Fotografija	Litva			•	•						
Zmanjšanje fotografij	Vietnam			•	•						
Bobri in vidre	Južna Koreja			•	•						
Kroglice	Uzbekistan			•	•						
Magična jablana	Kanada			•	•	•	•				
Vodenke 1	Vietnam				•						
Žaba in komar	Hrvaška				•	•	•				
Božičkovi pomočniki	Slovenija				•	•	•				
Parkirne dovolilnice 1	Estonija				•	•	•				
Polena	Švica				•	•	•	•	•		
Sejanje korenja	Slovaška				•	•	•	•	•		
Slike s poti 2	Japonska				•	•	•	•	•	•	
Squash turnir	Slovaška					•	•				

		2.	3.	4.	5.	6.	7.	8.	9.	SŠ	Str.
Stikala	Peru					●	●				
Vodenke 2	Vietnam					●	●	●	●		
Sporočila po reki	Švica					●	●	●	●		
Tekmovalci	Finska					●	●	●	●		
Banane	Švica					●	●	●	●	●	
Pametno ravnilo	Južna Koreja							●	●	●	
Robota	Finska					●	●	●	●	●	
Vzorci	Švica					●	●	●	●	●	
Stilist za ljubljenčke	Tajvan							●	●		
Toplo - hladno	Slovaška							●	●		
Parkirne dovolilnice 2	Estonija							●	●	●	
Odpri	Madžarska							●	●	●	
Snemanje cest	Irska							●	●	●	
Mostovi	Nova Zelandija							●	●	●	
Premiki po otokih	Slovenija									●	
Urejevalnik besedil	Avstrija									●	
Novi učitelj	Iran									●	
Aljini opravki	Belgija									●	
Matematični izrazi	Romunija									●	
Pisane hiške	Ciper									●	
BoberGPT	Portugalska									●	
Zaletavanje robotov	Češka									●	

Avtorji nalog

Akram Ahmed, Saudova Arabija
Mohammed Al-Badawi, Oman
Esraa Almajhad, Saudova Arabija
Daumilas Ardickas, Litva
Aldrich Ellis Asuncion, Filipini
James Atlas, Nova Zelandija
Masiar Babazadeh, Švica
Leonardo Barichello, Brazilija
Wilfried Baumann, Avstrija
Tim Bell, Nova Zelandija
Tobias C. Berner, Švica
Javier Bilbao, Španija
Eugenio Bravo, Španija
Špela Cerar, Slovenija
Sarah Chan, Kanada
Wan-Hsin Cheng, Tajvan
Omar Colon Reyes, Portoriko
Sébastien Combéfis, Belgija
Raluca Constantinescu, Romunija
Kris Coolsaet, Belgija
Valentina Dagienė, Litva
Darija Dasović, Hrvaška
Christian Datzko, Madžarska
Susanne Datzko, Švica
Mahmoud Elias, Sirija
Ali A.Elrowayati, Libija
Lidia Feklistova, Estonija
Kaja Ferenc, Slovenija
Fabian Frei, Švica
Gerald Futschek, Avstrija
Bence Gaál, Madžarska
Sonali Gogate, Indija
Adam Grodeck, Avstralija
Štefan Gura, Slovaška
Juan Gutiérrez Alva, Peru
Husnul Hakim, Indonezija
Tracy Henderson, Nova Zelandija
Angélica Herrera Loyo, Švica
Juraj Hromkovic, Švica
Andrea Hrušecká, Slovaška
Heikki Hyyrö, Finska
Yukio Idosaka, Japonska
Alisher Ikramov, Uzbekistan
Thomas Ioannou, Ciper
Takeharu Ishizuka, Japonska
Hyun-seok Jeon, Južna Koreja
Sangsu Jeong, Južna Koreja
Ungyeol Jung, Južna Koreja
Filiz Kalelioğlu, Turčija
Atheer Khabti, Saudova Arabija
Gohar Khachatryan, Armenija
Dong Kim, Južna Koreja
Dong Yoon Kim, Južna Koreja
Hakin Kim, Južna Koreja
Jihye Kim, Južna Koreja
Vaidotas Kinčius, Litva
Sebastian Knüsli, Švica
Jia-Ling Koh, Tajvan

Sophie Koh, Singapur
Victor Koleszar, Urugvaj
Maja Kosmač, Slovenija
Maria Krisel Cristobal, Filipini
Regula Lacher, Švica
Taina Lehtimäki, Irska
Carlos Luna, Peru
Yong Mao, Kitajska
Pedro Marcelino, Portugalska
Yoshiaki Matsuzawa, Japonska
Miha Medved, Slovenija
Anna Morpurgo, Italija
Khawla Mourad, Sirija
Madhavan Mukund, Indija
Tom Naughton, Irska
Jalil Nedaeepour, Iran
Henry Ong, SIngapur
Özgür Özdemir, Turčija
Marika Parviainen, Finska
Jean-Philippe Pellet, Švica
Zsuzsa Pluhár, Madžarska
Wolfgang Pohl, Nemčija
Pedro Ribeiro, Portugalska
João Rico, Portugalska
Karima Sayeh, Alžirija
Maiko Shimabuku, Japonska
Bronius Skūpas, Litva
Kristina Slišurić, Hrvaška
Bernadette Spieler, Švica
Alieke Stijf, Nizozemska
Gabriele Stupuriene, Litva
Chin-Hsin Sun, Tajvan
Goran Šuković, Črna gora
Ezra Templonuevo, Filipini
Monika Tomcsányiová, Slovaška
Ahto Truu, Estonija
Laura Ungureanu, Romunija
Vu Van Luan, Vietnam
Jiří Vaníček, Češka
Christine Vender, Kanada
Yuliana Viterbori, Paragvaj
Michael Weigend, Nemčija
Kyra Willekes, Nizozemska
Yi Shan Yeh, Tajvan
Khairul Anwar Mohamad Zaki, Malezija