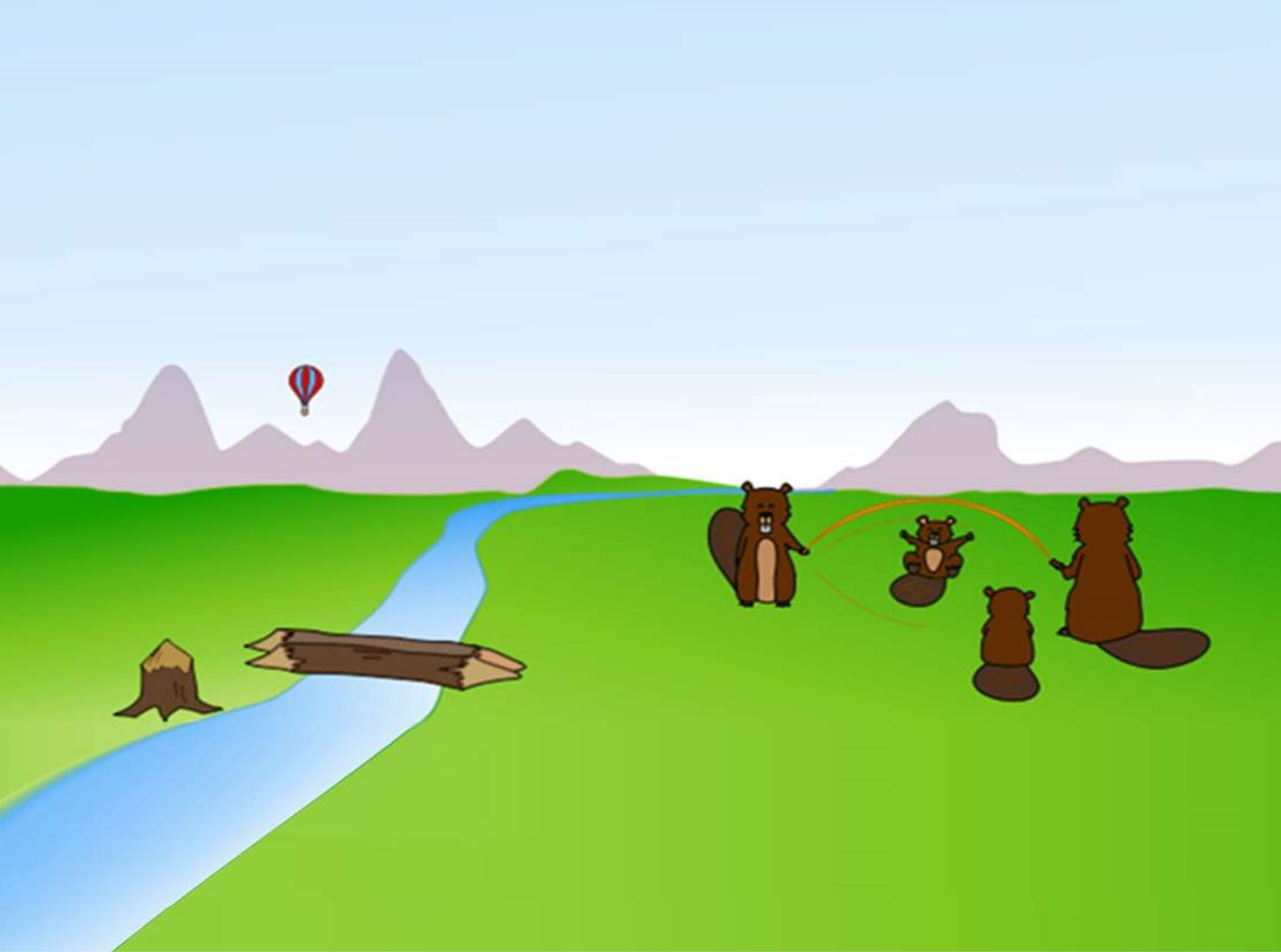




Bober 2018/19

Naloge in rešitve

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

Alenka Kavčič (FRI, Univerza v Ljubljani)

Janez Demšar (FRI, Univerza v Ljubljani)

Nežka Rugelj (PEF, Univerza v Ljubljani)

Špela Cerar (PEF, Univerza v Ljubljani)

Razvoj tekmovalnega sistema: Gašper Fele Žorž in Gregor Jerše (FRI, Univerza v Ljubljani)

Kazalo nalog

Ptičje barve	5	1 – 2 – 3 – hop!	48
Vrtenje kvadrata	6	Barvanje	49
Podobne jedi	7	Dvojiška torta	51
Baloni	8	Kje pušča?	52
Kup oblačil	9	Žarnice in stikala	53
Kaj bo danes oblekla Maja?	11	Bobrsko kodiranje	55
Lonec medu	12	Dvigalo	56
Metanje obročev	13	Jamb	58
Bobrova rojstnodnevna zabava	14	Most iz kamnov	60
Neskončni sladoled	17	Telegrafsko omrežje	62
Ovita drevesa	18	Urniki	64
Šopek	19	Vrsta v banki	66
Bobrovo jezero	20	Cimre	67
Ada in barvice	22	Koda ključavnice	69
Družinsko drevo	23	Pet prijateljev	70
Palindrom iz balonov	25	Prespana budilka	72
Iskanje poti	26	Vrstice in stolpci	74
Izleti	27	Koda za odklepanje	76
Urejanje knjig	28	Obračanje kart	77
Parkiranje na dovozu	29	Povezani krogi	80
Sprehod po parku	31	Skriti zemljevid	81
Dan kapitana Bobra	32	Analiza vzorcev	84
Besede iz Bobrove vasi	34	Delovna bobra	86
Gradovi	35	Izgubljen avto	88
Hotel Binarno	37	Ključne točke	90
Najdaljša veriga besed	38	Numerična tipkovnica	91
Utrjevanje bobrišč	39	Skakanje po kvadratih	92
Babuške	40	Velika skrivnost	93
Ena naloga na uro	41	Zveni podobno	95
Obiskovanje prijateljev	42	Kje je letalo?	96
Pustni sprehod	43	Krokodilja družina	98
Robot sadilec	44	Načrtovanje vaje	99
Skrivna koda	46	Prižgi vse luči!	101
Stapljanje kristalov	47	Urejanje knjig	103



MAVRIČNA PAPIGA IMA ŠTIRI MLADIČKE. VSAK IMA DRUGAČNO BARVO GLAVE, VRATU, TREBUHA IN HRBTA KOT VSI OSTALI.

SPODAJ SO SLIKE TREH MLADIČKOV. KAKŠNE BARVE PERJA IMA ČETRTE MLADIČEK?



Rešitev

KER SO IMELI DRUGI MLADIČI PERJE NA GLAVI RDEČE, RUMENE IN ZELENE BARVE, IMA LAHKO ČETRTE MLADIČEK NA GLAVI SAMO MODRO PERJE. ENAKO RAZMISLIMO TUDI ZA OSTALE DELE TELES.

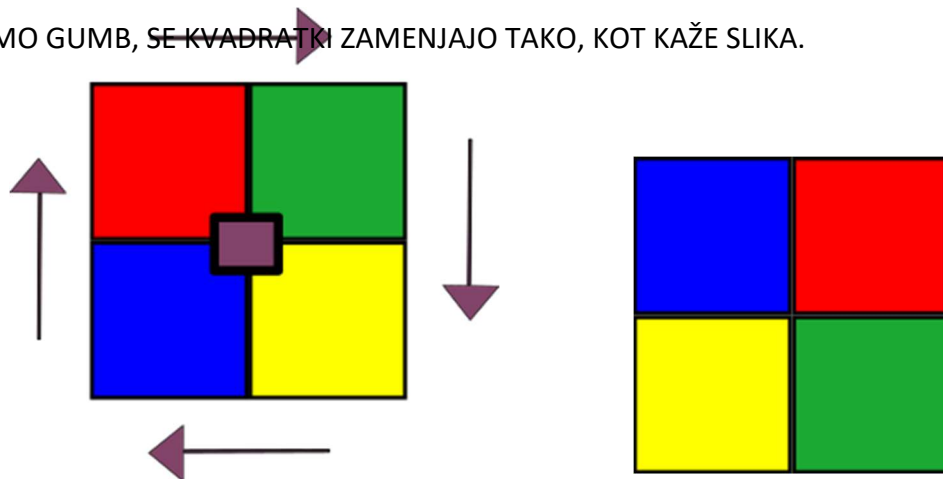
Računalniško ozadje

Tudi ko programiramo, pogosto razmišljamo o vseh možnih kombinacijah kakih reči ali o tem, katerih možnih kombinacij dogodkov naši programi še ne pokrivajo.

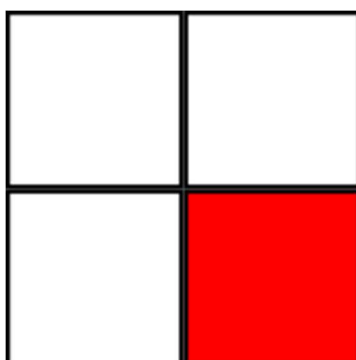




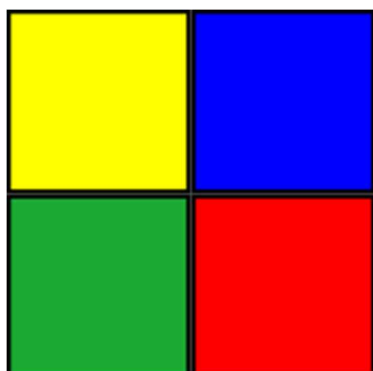
IMAMO KVADRAT, NA KATEREM SO ŠTIRJE MANJŠI BARVNI KVADRATI IN GUMB NA SREDINI. KO PRITISNEMO GUMB, SE KVADRATKI ZAMENJAJO TAKO, KOT KAŽE SLIKA.



KJE BODO STALI MODRI, ZELENI IN RUMENI KVADRAT, KO ŠE ENKRAT PRITISNEMO NA GUMB?



Rešitev



KVADRATI KROŽIJO V SMERI URINEGA KAZALCA.

Računalniško ozadje

Ko iščemo napake v programih, pogosto opazujemo – ali ugibamo – kaj se dogaja v programu in zakaj.



SLIKA KAŽE PET JEDI (TESTENINE, JAJČNA SOLATA ...) IN NJIHOVE SESTAVINE. DVE JEDI STA SI PODOBNI, ČE IMATA NAJMANJ DVE ENAKI SESTAVINI. KATERI DVE SPODNJI JEDI STA SI PODOBNI?

TESTENINE	JAJČNA SOLATA	SOLATA Z OREHI	KOKOŠJA JUHA	TORTA

Rešitev

PODOBNI STA JAJČNA SOLATA IN KOKOŠJA JUHA, SAJ SO V OBEH JAJCE, ČEBULA IN SOL. OSTALI RECEPTI IMAJO LE PO ENO SKUPNO SESTAVINO.

Računalniško ozadje

Ne boš verjel(a), a postopek, ki si ga uporabljal(a) v tej nalogi, se uporablja tudi v umetni inteligenci! Imenuje se "najbližji sosedi". Je zelo preprost: temelji na predpostavki, da so si stvari, ki so podobne po nekaj lastnostih, podobne tudi po drugih. Uporabljamo ga vsak dan: če imata dva bolnika enake težave, imata verjetno tudi isto bolezen. Veliko se uporablja tudi na spletu: če sta dva uporabnika spletne trgovine kupovala podobne stvari, lahko enemu od njiju priporočimo tisto, kar je kupil drugi (ta pa še ne) in obratno.



MARK JE BARVNO SLEP. ZELENA IN RDEČA BARVA SE MU ZDITA ENAKI. PRAV TAKO SE MU ZDITA ENAKI RUMENA IN MODRA. DA BI MU POMAGALI, SMO NA BALONE RAZLIČNIH BARV NAPISALI ČRKE.

KATERI DVE VERIGI BALONOV BI MARK VIDEL KOT ENAKI, ČE NA BALONIH NE BI BILO ČRK?

A



B



C



D



Rešitev

MARKU BI SE ZDELI ENAKI VERIGI BALONOV B IN D.

Računalniško ozadje

Ko pripravljamo računalniške programe in spletne strani, moramo misliti tudi na tiste, ki ne morejo razločevati barv. Prav tako moramo poskrbeti zanje, ko pripravljamo tekmovanje Bober.



MAMA VSAKO JUTRO ZLOŽI NA MIZO KUP OBLAČIL, KI JIH BO NJEN BOBRČEK
OBLEKEL TISTI DAN.



SRAJCA	SPODNJA MAJICA	HLAČE	SPODNJICE	NARAMNICE	NOGAVICE	ČEVLJI

BOBRČEK PO VRSTI OBLAČI OBLAČILA Z VRHA KUPA. DA SE MU SOŠOLCI NE BI SMEJALI, MORA:

- SPODNJO MAJICO OBLEČI PRED SRAJCO,
- SPODNJICE OBLEČI PRED HLAČAMI,
- SRAJCO OBLEČI PRED NARAMNICAMI,
- NOGAVICE OBUTI PRED ČEVLJI.

KATERI OD SPODNJIH KUPOV SO PRIPRAVLJENI PRAVILNO?

A	B	C	D

Rešitev

PRAVILNA ODGOVORA STA B IN D.

OBLAČILA OPAZUJEMO OD ZGORAJ NAVZDOL, TAKO KOT JIH BO JEMAL BOBRČEK. ODGOVOR A JE NAPAČEN, KER BI BOBRČEK NAJPREJ OBLEKEL SRAJCO IN NATO ČEZ NJO SPODNJO MAJICO. ODGOVOR C JE NAPAČEN, KER BI SI BOBRČEK NARAMNICE NADEL POD SRAJCO.

Računalniško ozadje

Nalogo, ki jo vsako jutro rešuje mama, opišemo z množico oblačil in množico odvisnosti (prednostne omejitve) oblike »oblačilo A si mora nadeti pred oblačilom B«. Podobne naloge pogosto rešujemo v računalništvu, ko je potrebno določiti smiselno zaporedje reševanja opravil.

Opravila – ali oblačila – je torej potrebno urediti. Za razliko od otrok, ki se urejajo po velikosti, ali plastenjk jogurta, ki jih urejamo po teži, pa pri opravilih in oblačilih ni možna le ena pravilna ureditev, temveč jih je lahko tudi več, tako kot se je zgodilo v tej nalogi. Takemu urejanju zato učeno rečemo *topološko urejanje*.

Kaj bo danes oblekla Maja?

3. razred



Maja se vsako jutro odloči, kaj bo oblekla.

- Če obleče hlače, obleče majico brez slike ali z zvezdami.
- Če obleče krilo, obleče majico z bobrom.
- Če obleče majico brez slike ali z zvezdami, obleče jakno s srcem.
- Če obleče jakno s srcem, obleče tudi kapo s sliko.

Katere od kombinacij na desni lahko obleče?

Rešitev

Pravilen odgovor je B.

Odgovor A ni pravilen, saj poleg hlač ne more obleči majice z bobrom.

Odgovor C ni pravilen, saj poleg plašča s srcem ne more obleči kape brez motiva. Odgovor D ni pravilen, saj poleg krila ne more obleči majice brez motiva.

Računalniško ozadje

V programe pogosto zapisujemo pogoje, ki določijo, kdaj naj se kaj zgodi. Majina pravila za oblačenje so tako nekakšen program.

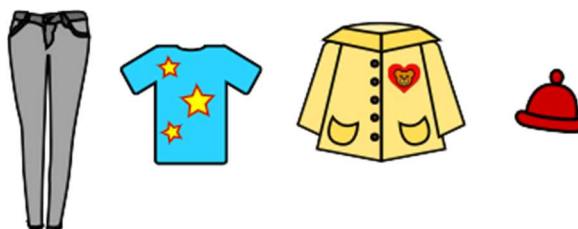
A



B



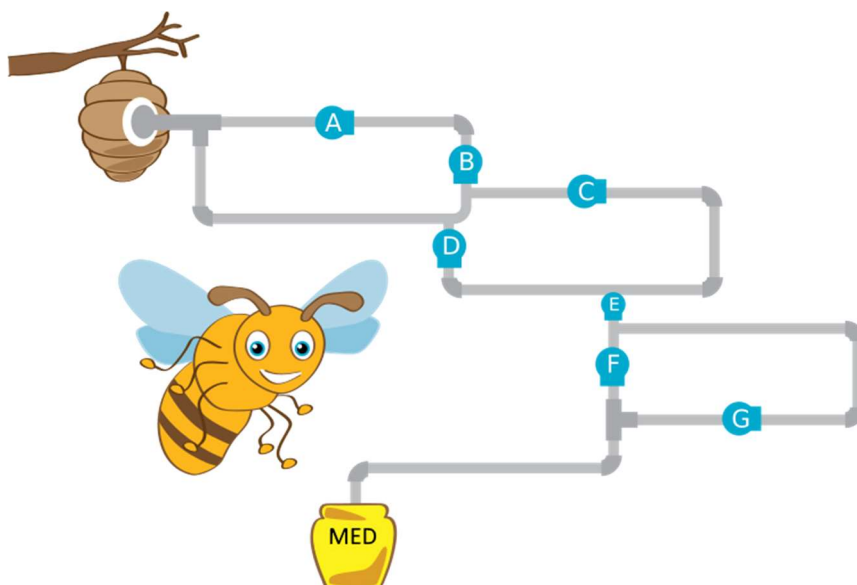
C



D



Obkroži ventile (●), ki morajo biti odprti, da čebela napolni LONEC z medom in pri tem umaže čim manj cevi.



Rešitev

Odprti morajo biti ventili D, E in F. Ostali morajo biti zaprti.

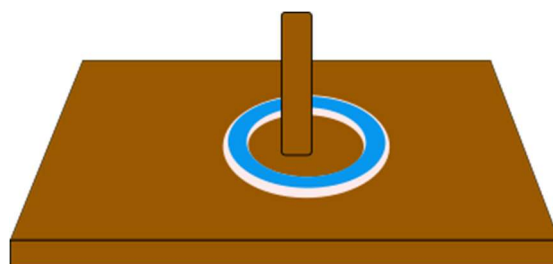
Računalniško ozadje

V nalogi iščemo najkrajšo pot. Računalniki, telefon in druge naprave to počnejo vsakič, ko jih vprašamo, kako čim hitreje priti v določen kraj ali na določen naslov.

Sara je s prijatelji igrala igro metanje obročev.

Vsak, ki je bil na vrsti, je vrgel pet barvnih obročev.

Ko vržejo prvi obroč, dobijo za zadeto palico 5 točk. Ko vržejo drugi obroč, dobijo za zadeto palico 4 točke. Za tretjega dobijo 3, za četrtega 2 in za petega 1 točko.



Sara je svoje obroče vrgla, kot kaže spodnja slika. Koliko točk je prejela?



Rešitev

Sara je prejela 6 točk. Palico je namreč zadela z drugim (4 točke) in četrnim (2 točki) obročem.

Računalniško ozadje

Si se ob reševanju naloge počutil skoraj kot detektiv, ki mora razvozlati, v kakšnem vrstnem redu so potekali neki dogodki? Tudi programerji pogosto počnejo isto, ko poskušajo odkriti, zakaj kak program ne deluje.

Bobrova rojstnodnevna zabava 3. – 5. razred



Bober je prijatelj z vsemi živalmi. Žal pa nekatere med njimi niso prijateljice med seboj.

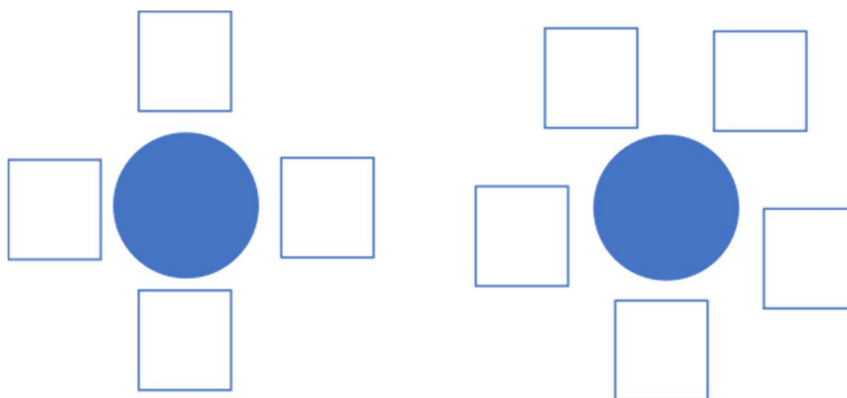
- **Zajček** je skregan z **lisico**, a je prijatelj z **medvedom**.
- **Veverica** je skregana z **medvedom**, a je prijateljica z **lisico**.
- **Sova** je prijateljica z **jelenčkom**, a se prepira z **rakunom**.
- **Jelenček** ne mara **srne**.
- **Jež** in **rakun** prijateljujeta z **zajčkom**, a se prepirata z **lisico**.

Bober bi rad praznoval svoj rojstni dan. Za dve mizi bi prijatelje posedel tako, da

- za nobeno mizo ne bo sedel kak par živali, ki je sprt,
- in da bodo vsi pari, ki so prijatelji, sedeli za isto mizo.

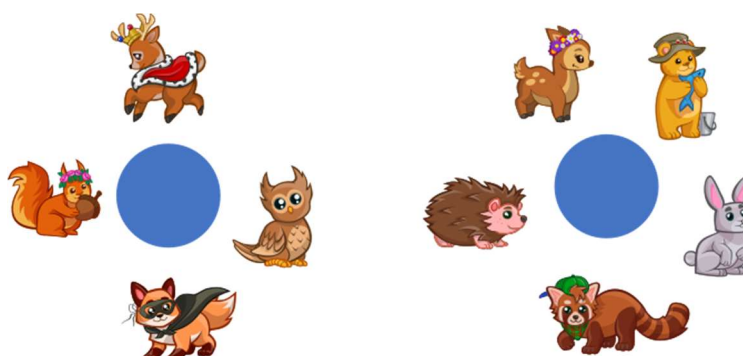
								
MEDVED	SRNA	ZAJEC	VEVERICA	SOVA	RAKUN	JELENČEK	LISICA	JEŽ

Pomagaj Bobru. Kako naj razporedi svoje prijatelje?



Rešitev

Dober sedežni red je na sliki.



Zajčka in lisice ne moremo posesti skupaj, torej ju damo za različni mizi.

MIZA 1	MIZA 2
LISICA	ZAJČEK

Za isto mizo kot lisico posedemo njeno prijateljico veverico. Medveda posadimo k zajčku, saj ne sme biti pri lisici.

MIZA 1	MIZA 2
LISICA	ZAJČEK
VEVERICA	MEDVED

K zajčku posedemo njegova prijatelja ježa in rakuna.

MIZA 1	MIZA 2
LISICA	ZAJČEK
VEVERICA	MEDVED
	JEŽ
	RAKUN

Ker sova ne mara rakuna, jo posedemo za mizo skupaj z lisico in veverico, poleg pa posedemo še njenega prijatelja jelenčka.

MIZA 1	MIZA 2
LISICA	ZAJČEK
VEVERICA	MEDVED
SOVA	JEŽ
JELENČEK	RAKUN

Ker jelenček ne mara srne, jo posedemo za drugo mizo.

MIZA 1	MIZA 2
LISICA	ZAJČEK
VEVERICA	MEDVED
SOVA	JEŽ
JELENČEK	RAKUN
	SRNA

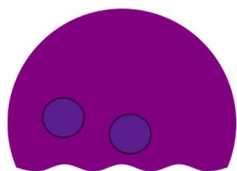
Računalniško ozadje

V tej nalogi iščemo rešitev problema, ki ustreza določenim omejitvam. Na takšne probleme naletimo zelo pogosto, zato so postopki za njihovo reševanje cela znanost.

Problem v tej nalogi je bilo lahko rešiti, saj smo začeli z eno od omejitev in druge omejitve so nas lepo vodile do rešitve. Pogosto pa se zgodi, da lahko v določenem koraku izberemo eno izmed več možnosti, pri čemer se lahko tudi zaplezamo, tako da v nadaljevanju ne najdemo rešitve. Tedaj se je potrebno vrniti nazaj in poskusiti drugače. Te to morda spominja na sudoku?



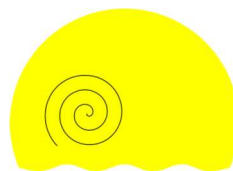
V eni od slaščičarn v Bobrovi deželi imajo dva sladoledna avtomata z enakimi okusi.



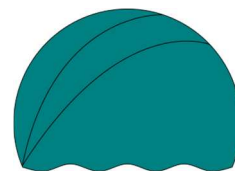
borovnica



jagoda



limona



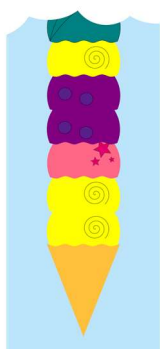
modro nebo

Prvi avtomat nalaga kepice v kornet po naslednjem postopku.

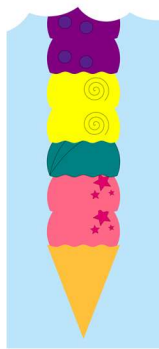
1. Začni s praznim kornetom.
2. Naključno izberi okus in naloži dve kepici tega okusa.
3. Dodaj eno kepico drugega okusa.
4. Če je število naročenih kepice doseženo, se ustavi, sicer se vrni na korak 2.

Drug avtomat kepice nalaga naključno.

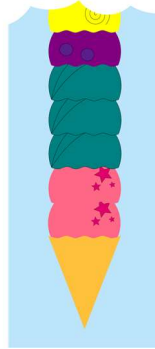
Na spodnjih slikah vidimo sladoled, ki so jih pripravili v slaščičarni. Vidimo le nekaj prvih kepice. Katerega od sladoledov so gotovo pripravili z drugim avtomatom?



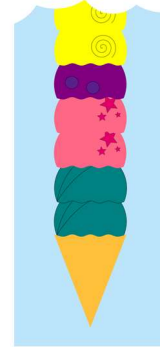
A



B



C



D

Rešitev

Z drugim avtomatom je bil zagotovo narejen sladoled D. To je edini sladoled, ki ne sledi postopku prvega avtomata. Sicer se pravilno začne z dvema kepicama istega okusa in nadaljuje z eno kepicco drugega okusa, vendar potem sledita dve kepici različnih okusov (jagoda in borovnica), prvi avtomat pa bi dodal dve kepici enakega okusa.

Računalniško ozadje

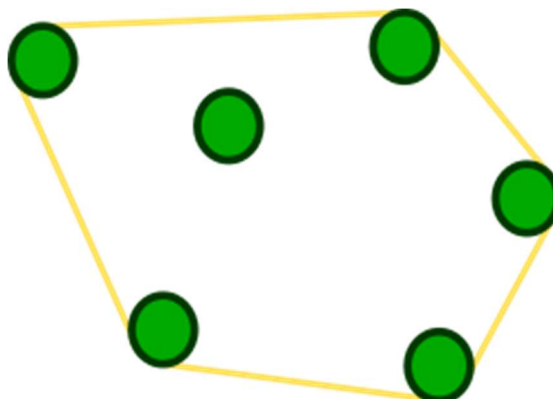
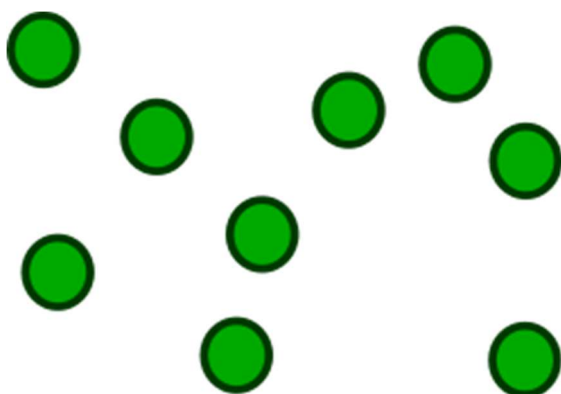
Prvi avtomat izvaja preprost program. V nalogi smo razmišljali o tem, kakšne rezultate bi lahko dal ta program ali, točneje, kakšnega rezultata zagotovo ni mogel dati.



Bobri označujejo drevesa, ki jih želijo uporabiti za izgradnjo jezusa, tako, da okrog njih ovijejo vrvi. Primer kaže slika na desni.

Znotraj vrvi je šest dreves, a le pet se jih dotika vrvi.

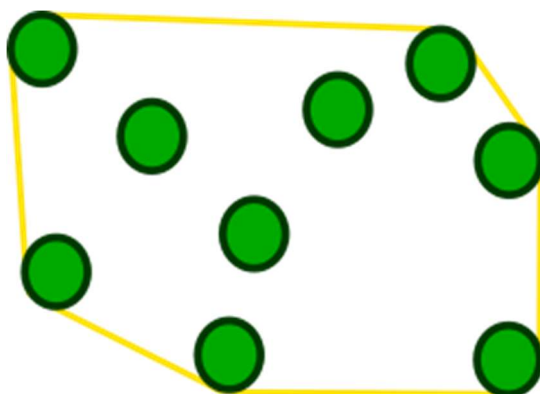
Drevesa, ki jih želi oviti Biba, so postavljena tako:



Koliko dreves se bo dotikalo vrvi, s katero jih bo ovila?

Rešitev

Šest.



Računalniško ozadje

Temu, kar smo naredili tu, se učeno reče *konveksna ovojnica*. Postopki, s katerimi jo lahko izračunamo, sodijo na zanimivo področje računalništva, ki se imenuje *računska geometrija*.



V cvetličarni prodajajo različne vrste rož.



GLADIOLA



LILIJA



TULIPAN



VRTNICA

Vsaka roža je na voljo v beli, modri ali rumeni barvi.

Klara se je odločila za šopek iz šestih rož.

1. Vsaka barva v šopku se pojavi natanko dvakrat.
2. Rože iste vrste niso enake barve.
3. Ista vrsta rože se v šopku pojavi največ dvakrat.

Katerega od spodnjih šopkov je kupila Klara?

A



B



C



D



Rešitev

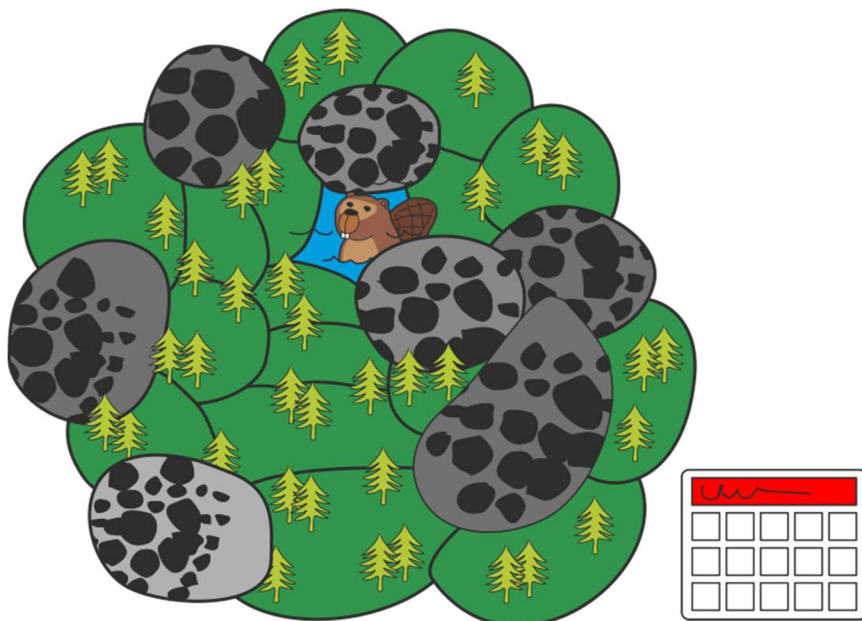
Pravilni odgovor je D. V šopku A so trije cvetovi bele barve, v B so tri vrtnice, v C pa sta obe gladioli rumene barve.

Računalniško ozadje

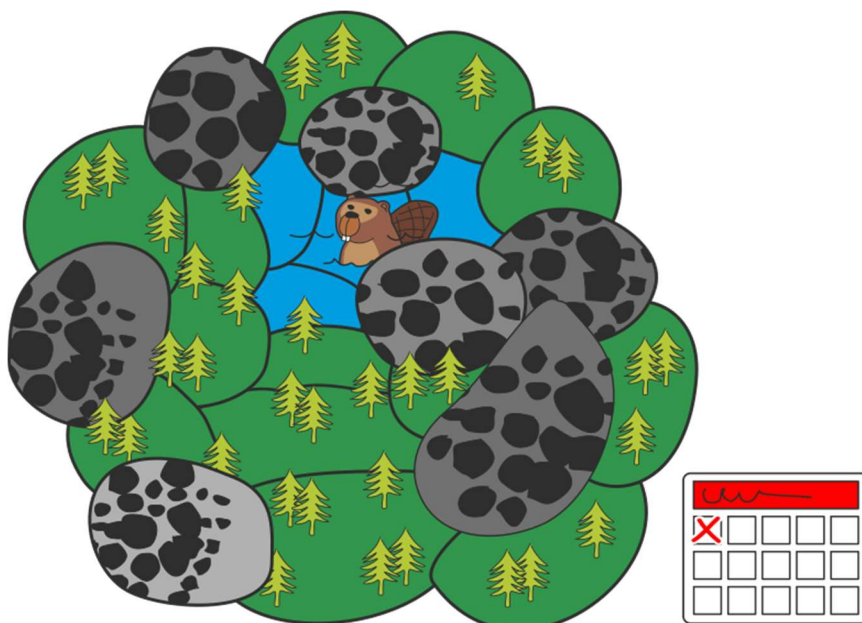
Tudi ko programiramo, pogosto razmišljamo o vseh možnih kombinacijah kakih reči ali o tem, katerih možnih kombinacij dogodkov naši programi še ne pokrivajo.



Bober živi v dolini, obdani z gorami. V dolini je jezero, okoli katerega so polja z drevesi ali kamni.



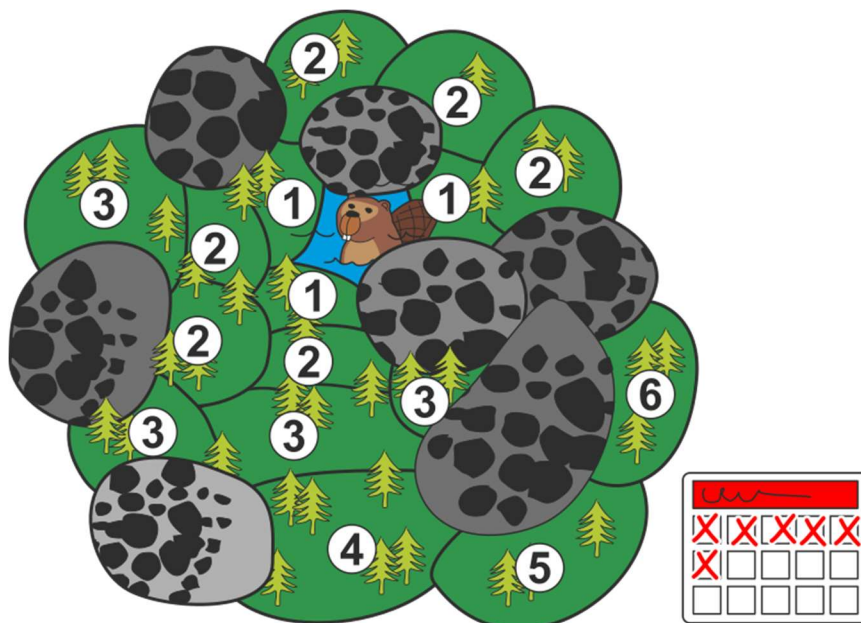
Vsak dan bober poplavi vsa polja, na katerih rastejo drevesa in ki mejijo na že poplavljen polja. Po prvem dnevu, na primer, poplavi tri sosednja polja. Polj s kamni ne poplavi.



Naslednji dan bo poplavlil vsa polja, ki mejijo na ta polja. Po koliko dneh bodo poplavljen vsa polja z drevesi?

Rešitev

Vsa polja bodo poplavljena na šesti dan.



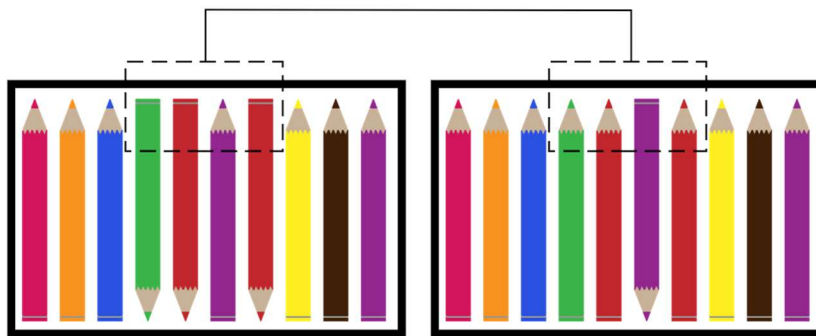
Računalniško ozadje

Računalnikarji pogosto uporabljajo postopek, podoben temu v tej nalogi, in rečejo mu – ne boš verjel(a) – poplavljanje! Potrebujejo ga za reševanje problemov, pri katerih jih zanima, na primer, najmanjše število korakov do nekega cilja. Zelo podoben je tudi postopek, ki ga naprave za navigacijo uporabljajo za iskanje najkrajših poti.

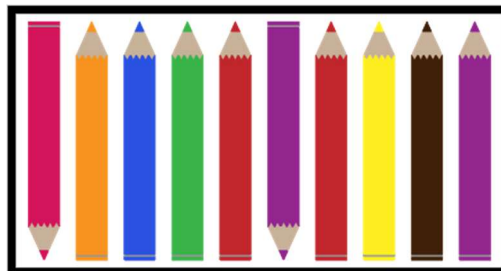


Ada ima škatlo z 10 barvicami. Nekatere so obrnjene navzgor in druge navzdol. Ada pravi, da so lepo urejene, če so obrnjene v isto smer.

Z igro poskuša vse barvice obrniti v isto smer. Odloči se, da nikoli ne bo obračala le ene barvice, temveč vsaj dve (lahko pa tudi več) sosednji barvici hkrati. Po obratu gledajo vse barvice, ki so bile prej obrnjene navzgor, dol in vse barvice, ki so bile prej obrnjene navzdol, gor. Kot na sliki.



Najmanj koliko zaporednih obratov je potrebnih, da uredi barvice na spodnji sliki?



Rešitev

Dva obrata.

Če obrnemo zaporedje barvic od 2 do 5 in zaporedje barvic od 7 do 10, dobimo lepo urejene barvice, saj vse gledajo dol, ali pa obrnemo najprej zaporedje barvic od 1 do 6, nato pa ponovno obrnemo zaporedje barvic od 2 do 5, da vse barvice gledajo navzgor.

Računalniško ozadje

Rešitev problema v čim manj korakih je ena od najpomembnejših veščin v računalništvu – in tudi vsakdanjem življenju.

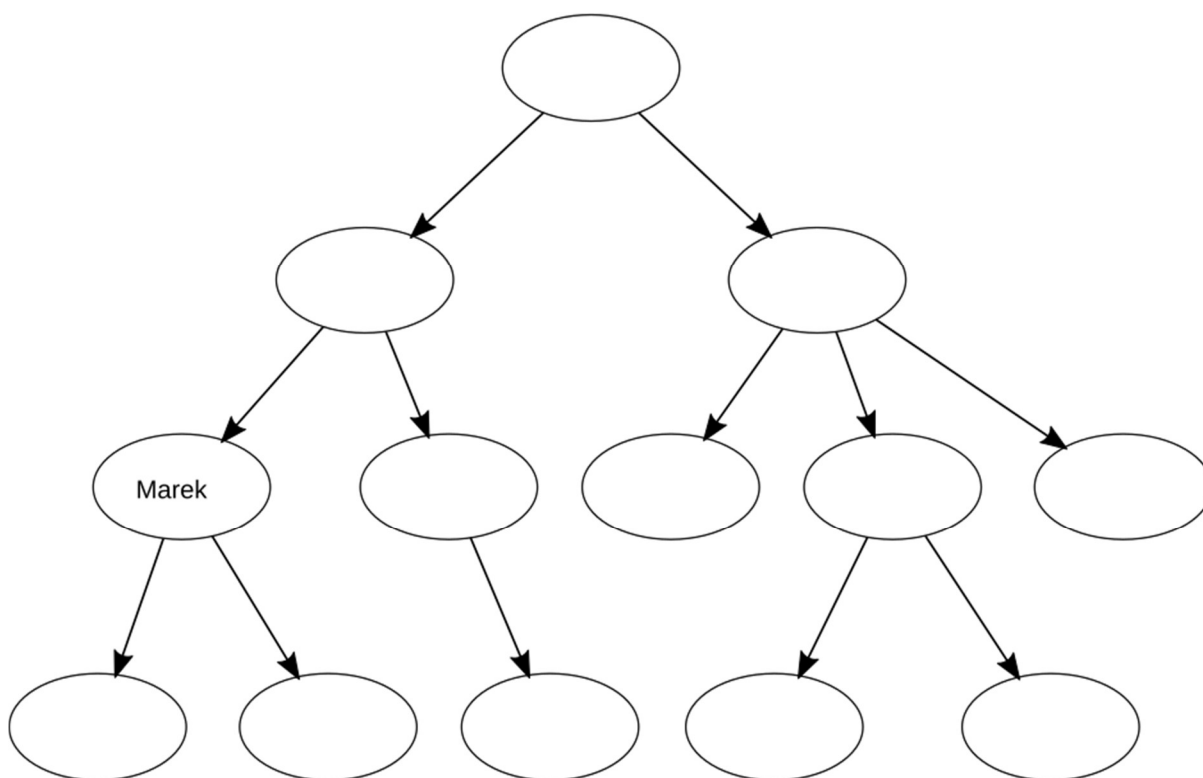


Marek si želi izdelati družinsko drevo vseh moških v svoji družini. O družini vemo tole.

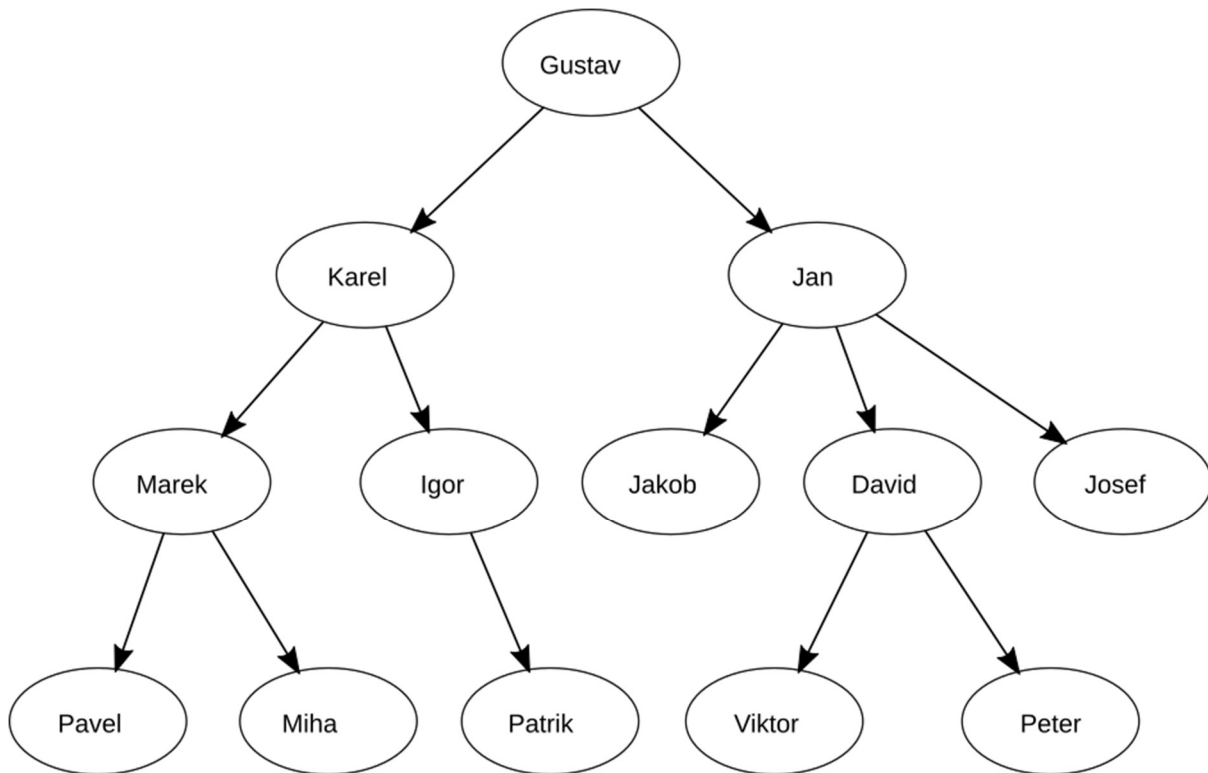
- Viktor je Davidov sin.
- Igor ima samo enega sina. Ime mu je Patrik.
- Miha je Pavlov brat.
- Gustav je Janov oče.
- Karel je Marekov oče.
- Pavel je Marekov sin.
- Jan je Marekov stric.
- Jakob, David in Josef so Marekovi bratranci.
- Tudi Peter je Marekov sorodnik.

Puščica od A do B v družinskem drevesu pomeni, da je A oče od B.

Dopolni drevo. Pomagaj si z zgoraj napisanimi razmerji.



Rešitev



Računalniško ozadje

Družinsko drevo tu predstavlja povezave med osebami. V računalništvu pa drevesa pogosto uporabljamo tudi kot predstavitev za druge povezave, zato so ena od osnovnih načinov shranjevanja podatkov. V tej nalogi si moral pokazati, da razumeš, kako drevo odraža povezave. Če ti je uspelo, si na dobri poti, da boš nekoč pravi računalnikar ali računalnikarka.



Poznaš stavek PERICA REŽE RACI REP, ki se z desne bere enako kot z leve? Ela počne nekaj podobnega z baloni: postavlja jih v vrsto tako, da je vrstni red barv balonov z ene in z druge strani vrste enak.

Ela pa je poleg tega barvno slepa in ne loči med zeleno (Z) in rdečo (Rd), prav tako ne med rumeno (Ru) in modro (M). Ti dve verigi vidi, kot da bi bili z leve in z desne enaki, čeprav nista.



Za katero od spodnjih dveh verig se ji prav tako zdi, da si sledijo barve z leve enako kot z desne?



Rešitev

Pravilni odgovor je B. Ela vidi balone približno tako (rumeno smo nadomestili z modro in rdečo z zeleno).



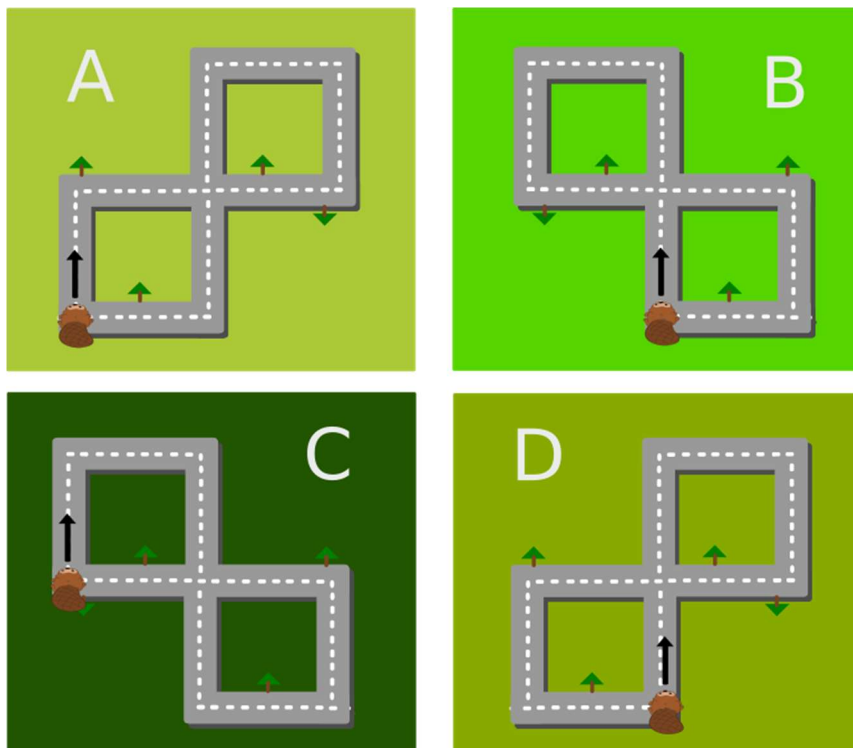
Računalniško ozadje

Ko pripravljamo računalniške programe in spletne strani, moramo misliti tudi na tiste, ki ne morejo razločevati barv. Prav tako moramo poskrbeti zanje, ko pripravljamo tekmovanje Bober. Zakaj imajo baloni v tej nalogi tudi črke? Prav zato, ker bi imeli barvno slepi lahko težave s prepoznavanjem njihovih barv. Morda pa je kateri od njih lažje reševal to nalogo?



Bober Žan je prehodil 8 kilometrov. Cesta, po kateri je šel, je sestavljena iz odsekov, dolgih 1 kilometer. Po vsakem kilometru se je obrnil levo ali desno in v zvezek zapisal znak 1 ali 0. Ne vemo pa, kateri znak predstavlja obrat v levo in kateri obrat v desno smer. Celotno pot si je zapisal kot 1000100.

Lahko uganemo? Kateri zemljevid predstavlja možno pot? Žanov začetni položaj in smer sta prikazana na vsaki sliki.



Rešitev

Pravilen odgovor je B in D.

Če desno zapišemo kot D in levo kot L, lahko Žanovo pot beremo kot LDDDLDD ali kot DLLLLDL.

- Pri A to ne gre, saj bi moral iti Žan najprej desno, nato levo, nato desno, DLD pa ni začetek nobene od možnosti.
- Pri B gre Žan najprej levo, nato 3-krat desno, levo in nato spet 2-krat desno, kar ustreza rešitvi.
- Pri C to ne gre, saj mora Žan najprej 2-krat desno, DD pa ni enako, kot je v Žanovih zapiskih.
- Pri D gre Žan najprej desno, nato 3-krat levo, nato desno in 2-krat levo, kar je ena od možnih rešitev.

Računalniško ozadje

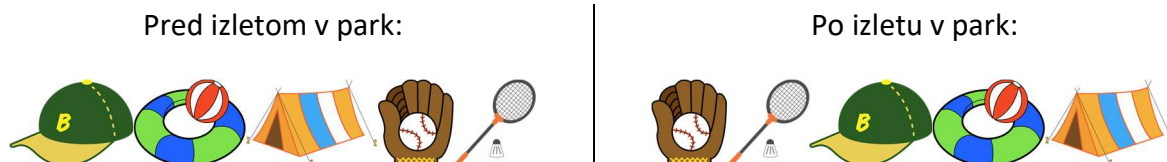
Z dvema vrednostma, ki ju pogosto predstavimo kot 0 in 1, lahko zapišemo različne stvari. V našem primeru je bober Žan z 0 in 1 zapisoval smeri, v kateri se lahko obrne.



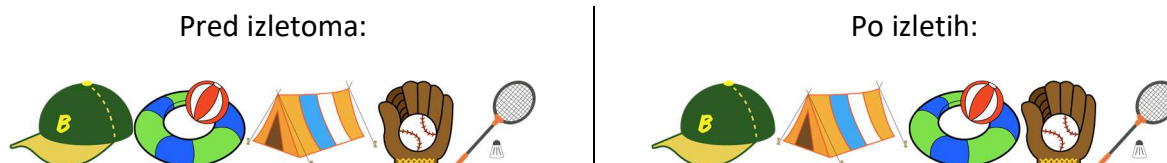
Jana gre rada na izlet. Katere stvari bo vzela s seboj, se odloči glede na to, kam bodo šli.



Jana shranjuje svoje stvari v kleti. Po vsakem izletu postavi zadnje uporabljene stvari levo od tistih, ki jih tokrat ni uporabila. Tako lahko izgleda klet pred in po izletu v park.



Jana je bila pred kratkim na dveh izletih. Kje?



- A V parku in na morju
- B V parku in ob reki
- C Na morju in v hribih
- D V parku in v hribih

Rešitev

Na morju in v hribih.

Na zadnja dva izleta Jana zagotovo ni vzela s seboj loparja, zato ni mogla biti na izletu v parku in ob reki, tako ostaneta možna le izleta v hribe in na morje. Ker sta zadnji uporabljeni stvari kapa in šotor, je bila Jana nazadnje v hribih, pred tem pa je bila na morju in uporabila plavalni obroč in šotor.

Računalniško ozadje

Programer je pogosto kot detektiv, ki mora na podlagi stanja, preden se izvede del programa, in stanja po tem sklepati, kaj se je dogajalo vmes.



Trije bobri sedijo vsak za svojo mizo. Na vsaki mizi sta dve knjigi. Vrstni red knjig je pomešan, zato jih želijo bobri urediti po vrsti od 1 do 6.

Da bo urejanje bolj zabavno, bodo v vsakem krogu knjige zamenjevali na dva različna načina (kot kaže slika).

- A. Vsak bober lahko zamenja vrstni red knjig na svoji mizi;
- B. Bobri lahko zamenjajo najbližji knjigi na vseh sosednjih mizah.



V prvem krogu vsak bober po potrebi zamenja vrstni red knjig na svoji mizi (način A).

Najmanj koliko krogov je potrebnih, da bodo bobri uredili vse knjige na sliki po vrsti od 1 do 6?

Rešitev

Potrebni so štirje krogi. Stanje po izvedbi posameznega kroga:

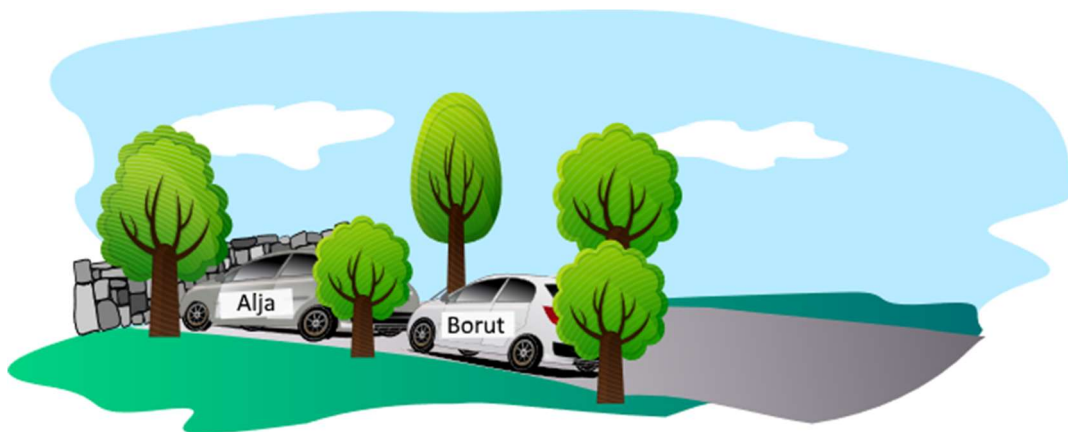
začetek	5	1	6	3	4	2
1. krog	1	5	3	6	2	4
2. krog	1	3	5	2	6	4
3. krog	1	3	2	5	4	6
4. krog	1	2	3	4	5	6

Računalniško ozadje

Urejanje je tipičen računalniški postopek in pri njegovem programiranju v konkretnih situacijah pogosto naletimo na omejitve, kot je ta, ki jo opisuje naloga.



Pri Alji doma imajo zelo dolg in ozek dovoz. Ker v okolici primanjkuje parkirišč, so se sosedi z njo dogovorili, da lahko parkirajo na njenem dovozu.



Da ne bi bil kdo zaparkiran, je pripravila tabelo, v kateri piše, kdaj lahko kdo parkira in kdaj mora njegov avto zapustiti dovoz.

Vsako jutro avtomobili, ki želijo zapustiti dovoz, odpeljejo, preden pridejo novi avtomobili.

Iz tabele lahko preberemo, da v ponedeljek dovoza ne zapusti nihče.

Alja prva parkira svoj avto, Borut parkira za njo.

dan	število avtov, ki odpeljejo	število avtov, ki na novo parkirajo	lastniki avtov in vrstni red prihodov
ponedeljek	0	2	Alja, Borut
torek	1	3	Katja, Branko, Rok
sreda	2	1	Darja
četrtek	0	2	Franci, Romana
petek	3	1	Vid

Čigavi avtomobili bodo na dovozu parkirani v petek **zvečer**?

- A. Borut, Vid, Darja
- B. Vid, Alja, Romana
- C. Alja, Katja, Vid
- D. Alja, Vid, Borut

Rešitev

Pravilen odgovor je C. Ob koncu vsakega dneva bo vrstni red parkiranih avtov na dovozu tak:

- Ponedeljek zvečer: Alja, Borut
- Torek zvečer: Alja, Katja, Branko, Rok
- Sreda zvečer: Alja, Katja, Darja
- Četrtek zvečer: Alja, Katja, Darja, Franci, Romana
- Petek zvečer: Alja, Katja, Vid

Računalniško ozadje

Dovoz v tej nalogi predstavlja *sklad*. Ta deluje po principu "zadnji pride, prvi gre". Podatek, ki se na sklad shrani kot zadnji, moramo pobrati z vrha sklada, preden lahko dostopamo do naslednjega, nove pa vedno dodajamo na vrh sklada.

Na sliki je načrt parka.

Zeleni krogi prikazujejo drevesa in rjave črte poti. Nekateri črke se pojavijo na več različnih drevesih. Sprehod od drevesa F do drevesa B opišemo kot **F D E C A B**.

V soboto sta se po parku sprehajali dve družini.

Družina Širok je šla mimo dreves

B A A A C E D E E D A,

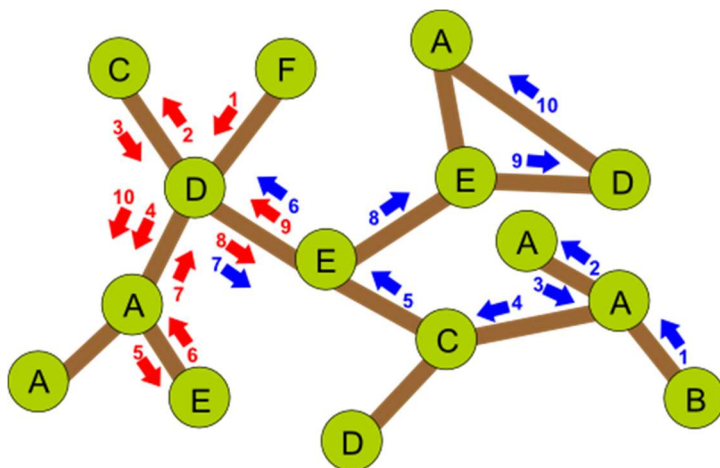
družina Skok pa mimo

F D C D A E A D E D A.

Kolikokrat se bosta družini srečali pri kakem drevesu, če sta obe odšli na sprehod istočasno in potrebujeta enako časa za pot od enega do drugega drevesa?

Rešitev

Družini se ne srečata nikoli. Sprehoda obeh družin sta označena na spodnji sliki.



Računalniško ozadje

Kdor zna rešiti takšno nalogo, bo znal slediti tudi izvajanju programa.



Kapitan Bober rad pomaga bobrom, ki ga prosijo za pomoč. Žal lahko kapitan Bober opravlja le eno nalogo naenkrat. Če bober prosi kapitana za pomoč, medtem ko ta že pomaga drugemu bobru, se kapitan glede na pomembnost naloge odloči, kaj bo naredil:

Pomembnost Naloga

Visoka: Ukvarjanje z mladimi bobrčki.

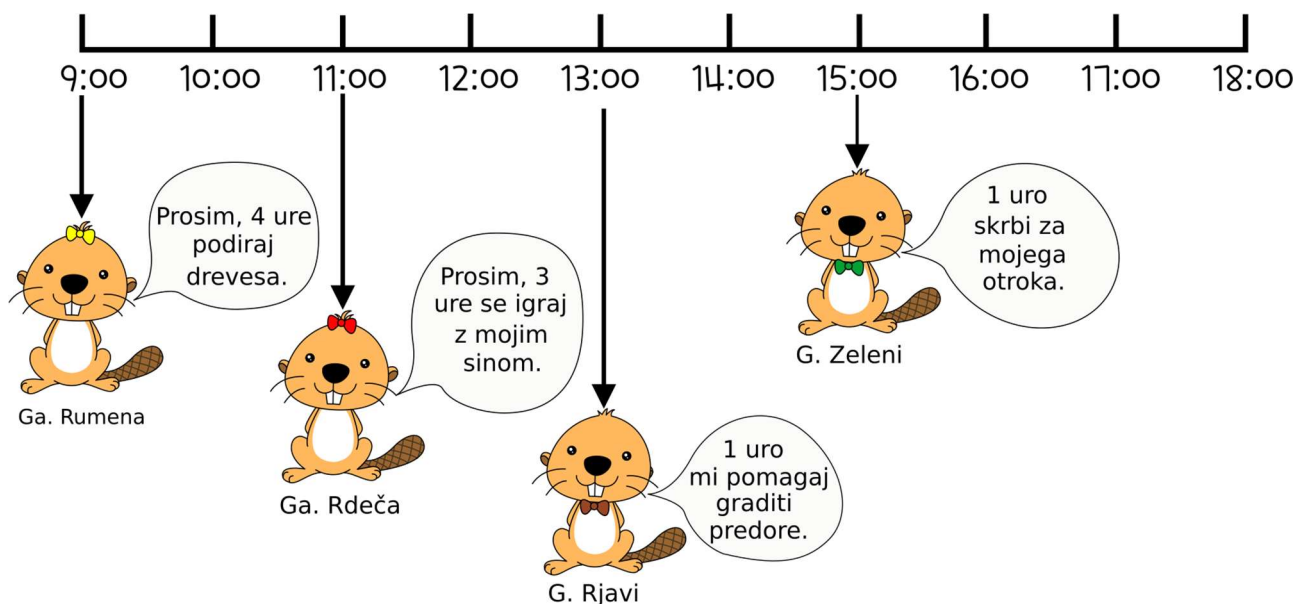
Srednje visoka: Delo z drevesi.

Nizka: Druge naloge.

Če kapitana prosijo za pomoč pri nalogi, ki je

- manj pomembna od trenutne, bo nadaljeval s trenutno nalogo, novo nalogo bo opravil kasneje.
- enako pomembna od trenutne, bo opravljal obe nalogi izmenično, vsako po eno uro.
- bolj pomembna od trenutne, bo prekinil opravljanje trenutne naloge in začel opravljanje pomembnejšo. K trenutni se bo vrnil, ko opravi pomembnejšo nalogo.

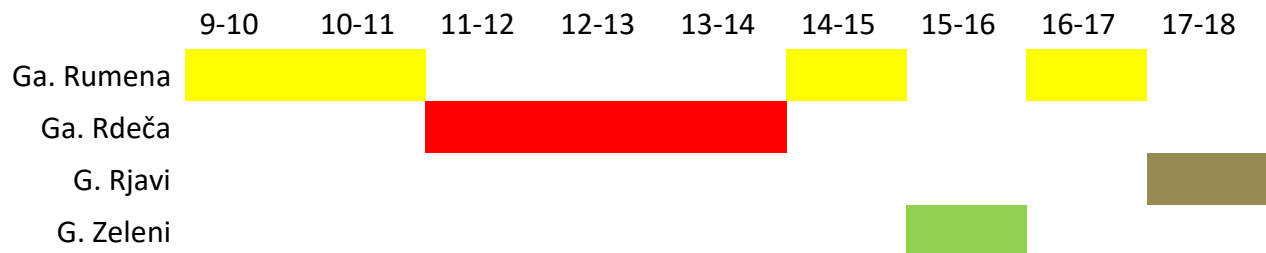
Nekega dne so kapitana Bobra za pomoč prosili štirje bobri:



Kdaj bo kapitan Bober dokončal nalogo gospe Rumene?

Rešitev

Ob petih popoldan.



Računalniško ozadje

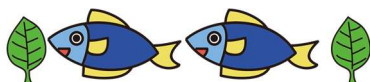
Računalnik je kot kapitan Bober. Operacijski sistem (MS Windows, macOS, Linux) odloča, kaj bo počel v posameznem trenutku. Če je naloga, ki jo trenutno izvaja, manj pomembna, to nalogo začasno shrani in začne z izvajanjem tiste, ki je bolj pomembna oziroma ima višjo prioriteto.



Za besede, ki jih uporabljajo v Bobrovi vasi, veljajo naslednja pravila:

- besede so sestavljene iz naslednjih znakov: ,  in  ;
- veljavna beseda se enako prebere tako z leve kot tudi z desne.

Primeri veljavnih besed:

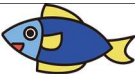
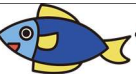
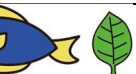


Dolžina besede je enaka številu uporabljenih znakov. Beseda      je dolga 5 znakov.

Koliko veljavnih besed, sestavljenih iz treh znakov, lahko sestavimo?

Rešitev

Devet.

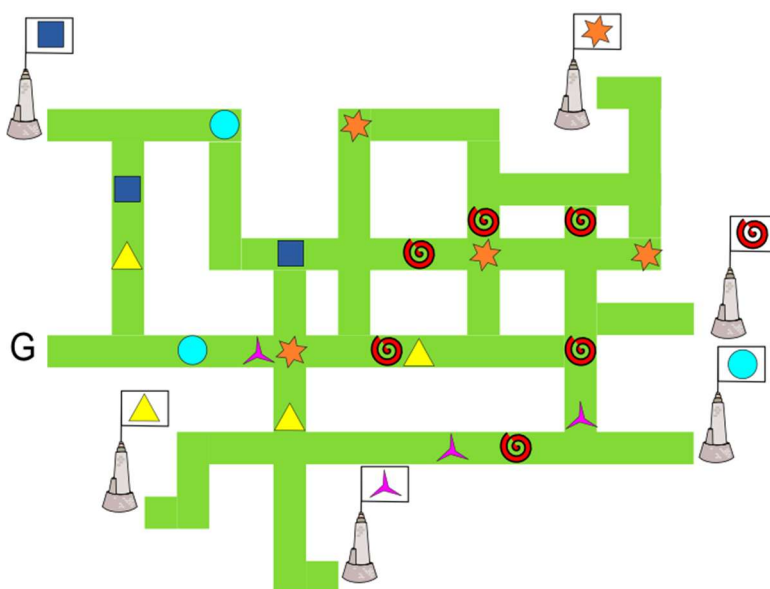
	  	  	  
	  	  	  
	  	  	  

Računalniško ozadje

Besedi, ki se z desne bere enako kot z leve, rečemo palindrom. Kakšno zvezo ima to z računalništvom? Ne prav velike, razen te, da palindrome pogosto uporabljamo za vajo iz programiranja, posebej zanimivi pa so za vaje iz nekih zapletenih reči, ki jim pravijo skladovni avtomati. Več o njih lahko izveš na študiju računalništva. :)

V neznani deželi stoji šest gradov, ki so označeni vsak z zastavo s svojim grbom. General (označen z G) jih želi zavzeti, zato ima v vsakem gradu svoje vohune.

Lastniki gradov so na poti postavili svoje stražarje (grbi na poteh). Tako stražarja, predstavljena z modrimi kvadrati, preprežeta vsakogar, ki bi odnašal podatke iz gradu, katerega grb je kvadrat. Stražarji z zvezdami preprežajo vohune iz gradu, katerega grb je zvezda ...



Iz katerih gradov lahko general neovirano dobi tajne podatke?

Rešitev

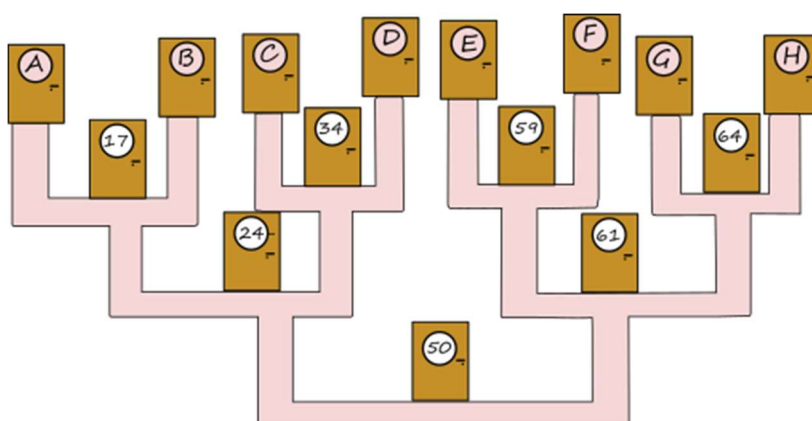
S 4 gradov, ki imajo v grbu: oranžno šesterokrako zvezdo, rdečo spiralo, rumen trikotnik in vijolično trikrako zvezdo.

Ali je mogoče priti neovirano od generala do gradu, je razvidno z naslednjih slik:

Miloš dela v hotelu. Njegova današnja naloga je, da na vsa vrata v hotelu obesi nove številke sob. Začel bo pri številki 50 in sledil naslednjim navodilom:

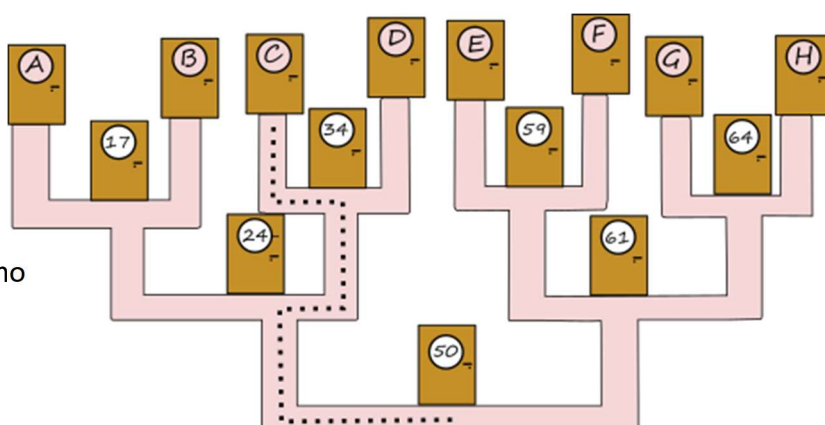
- Če je nova številka sobe manjša kot številka sobe, pri kateri si, pojdi levo.
- Če je nova številka sobe večja kot številka sobe, pri kateri si, pojdi desno.
- Če soba, pri kateri si, še nima številke, tja obesi novo številko sobe.

Nekatere sobe so že oštevilčene, nekatere pa imajo na vratih črke in jih je potrebno še oštevilčiti. Nova številka sobe, ki jo mora obesiti Miloš, je 29. Kam naj jo obesi?



Rešitev

Začnemo pri vratih 50. Ker je 29 manj kot 50, gremo levo in pridemo do vrat 24. Ker je 29 več kot 24, gremo desno in pridemo do vrat 34. Ker je 29 manj kot 34, gremo levo in pridemo do vrat z oznako C. Na vrata z oznako C pritrdimo številko 29.

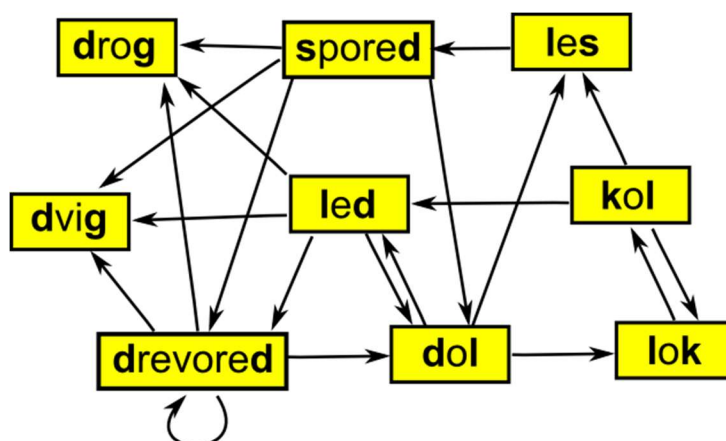


Računalniško ozadje

Sobe v hotelu predstavljajo vozlišča v *dvojiškem iskalnem drevesu*. Dvojiško iskalno drevo je v računalništvu pogosto uporabljena podatkovna struktura, saj je podatke, ki so shranjeni v njem, preprosto iskati. Vse vrednosti, ki so manjše od vrednosti v nekem vozlišču, so namreč shranjene v levem delu drevesa, večje pa v desnem delu. Posledično določeno vrednost vedno iščemo samo v eni od obeh strani drevesa (tako kot pri naši razlagi).



Bobra se igrata besedno igro. Prvi bober si izbere eno besedo. Drugi bober si mora izmisliti besedo, ki se začne z zadnjo črko te besede. Prvi bober mora nato najti drugo besedo, ki se prav tako začne na zadnjo črko zadnje besede in še ni bila uporabljena in tako naprej. Žal ta dva bobra ne poznata prav veliko besed. Njuno besedišče lahko predstavimo kar s spodnjo sliko.



Največ koliko besed je lahko dolga veriga besed v eni igri?

Rešitev

Obstaja 16 različnih verig iz 8 besed. Primer je led – dol – lok – kol – les – spored – drevored – drog. Verige iz 9 besed ni mogoče sestaviti. To je očitno: čim pridemo v drog ali dvig, verige ne moremo več nadaljevati, torej bo vsaka veriga vsebovala (kvečjemu) eno od teh dveh besed.

Računalniško ozadje

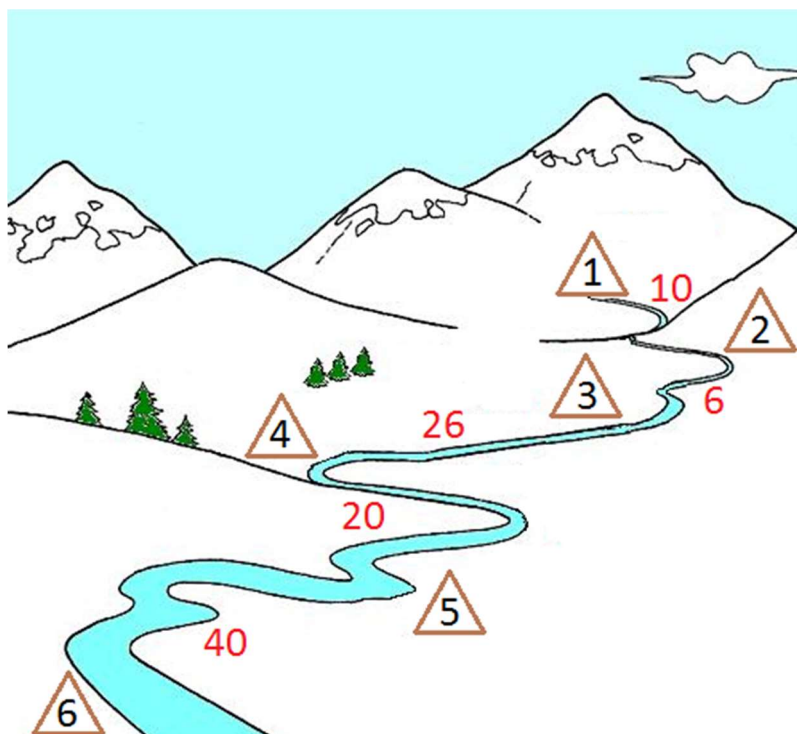
Kako bi reševal nalogo, če bi ti dali le seznam besed, ne da bi jih povezali v sliko? Bilo bi precej težje, ne? Pri reševanju takšnih problemov je pomembno, da znamo podatke pretvoriti v primerno obliko (gornji rečemo *graf*) in da se domislamo, s katerim od običajnih algoritmov za delo s takšnimi podatki se je bomo lotili. Na ta način nalogo, ki jo moramo rešiti *prevedemo* na katero od običajnih, abstraktnejših nalog, za katere že poznamo dobre postopke reševanja. V gornjem primeru smo iskali najdaljšo pot v grafu. Dolžini najdaljše poti v grafu rečemo tudi premer grafa.

Tudi iskanje najdaljše poti v grafu pa nikakor ni preprost problem. Nasprotno, sodi v kategorijo, ki jih učeno pravimo "NP-polni problemi", kar pomeni, da za njihovo reševanje v splošnem ne poznamo nobenega bistveno učinkovitejšega postopka kot preverjanje vseh možnih poti. Natančno tako smo naračunali gornji podatek, da obstaja 16 različnih verig iz 8 besed.

Bobri živijo v šestih bobriščih vzdolž reke. Na sliki je označeno, koliko minut bobri potrebujejo za pot od enega do drugega bobrišča. Da bodo lažje preživeli zimo, so se odločili, da dve bobrišči razširijo in utrdijo, da ju sneg ne bo mogel poškodovati.

Kadar bodo napovedane obilne snežne padavine, bodo bobri kar se da hitro prišli v ti dve bobrišči. Čas, ki ga bober potrebuje za to pot, imenujemo evakuacijski čas. Bobri bi radi zgradili večji bobrišči tako, da bo najdaljši čas, ki ga potrebujejo bobri do najbližjega utrjenega bobrišča, čim krajši.

Na katerih dveh mestih morajo razširiti in utrditi svoji bobrišči?



Rešitev

Možnosti je več: 1 in 5, 2 in 5, 3 in 5, 2 in 6 ali 3 in 6; pri vseh najdaljši čas ne preseže 40 minut. Vendar je za vse bobre najboljša druga možnost.

Najdaljša razdalja je med 5 in 6: 40 minut, zato morajo utrditi bobrišče bodisi v 5 bodisi v 6. Ker je 4 bližje 5 (20 minut) kot 3 (26 minut), bo utrjeno bobrišče v 5.

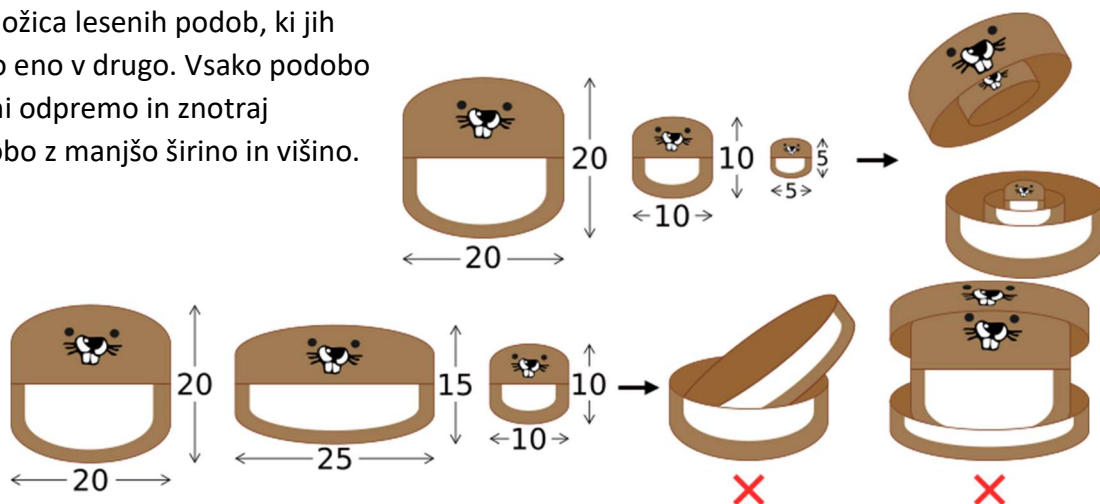
Drugo bobrišče bo v 2, saj so tako bobri iz 1 v utrjenem bobrišču v 10 minutah, bobri iz 3 pa v 6 minutah.

Računalniško ozadje

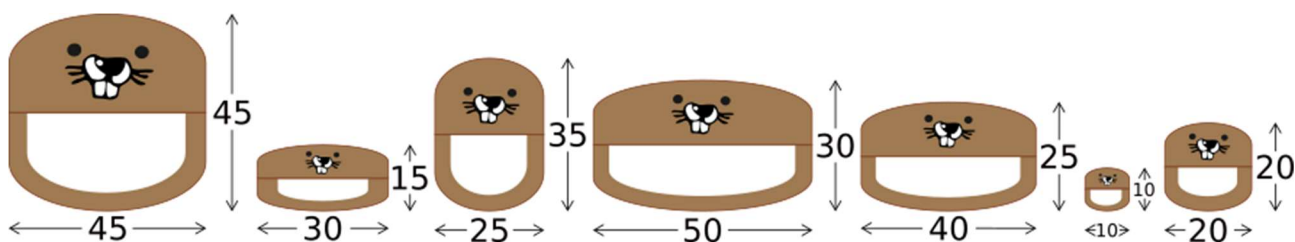
Naloga je tipičen primer optimizacijskega problema. V optimizacijskih problemih moramo iz množice možnih rešitev (v našem primeru možnih parov bobrišč, ki jih lahko utrdimo) poiskati tisto rešitev, ki je najugodnejša glede na določen kriterij. Kriterij je lahko preprostost, cena, estetskost ali, kot v našem primeru, čas.

	1	2	3	4	5	6
1	0	10	16	42	62	102
2	10	0	6	32	52	92
3	16	6	0	26	46	86
4	42	32	26	0	20	60
5	62	52	46	20	0	40
6	102	92	86	60	40	0

Babuške so množica lesenih podob, ki jih lahko shranimo eno v drugo. Vsako podobo lahko na sredini odpremo in znotraj shranimo podobo z manjšo širino in višino.



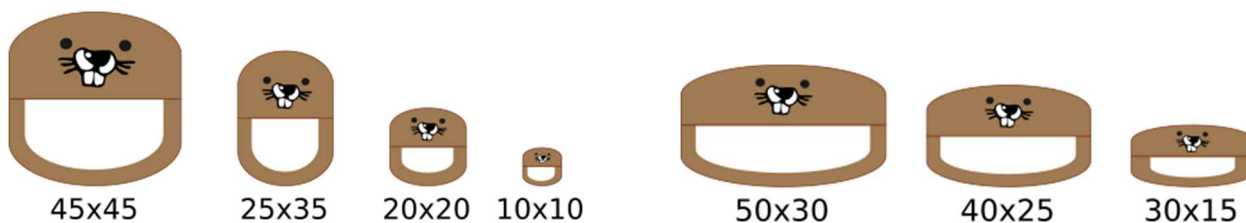
Ema ima veliko podobno oblikovanih babušek:



Pospraviti jih želi eno v drugo, tako da jih bo na mizi ostalo čim manj. Koliko jih bo?

Rešitev

Štiri babuške je mogoče zložiti v babuško 45x45, tri pa v 50x30.



Računalniško ozadje

Nalogo lahko rešimo s požrešno metodo, pri kateri v vsakem koraku iščemo najboljšo izbiro, za katero upamo, da bo v končni fazi pripeljala do najboljše rešitve. Uporaba požrešne metode sicer ne privede vedno do optimalne rešitve.

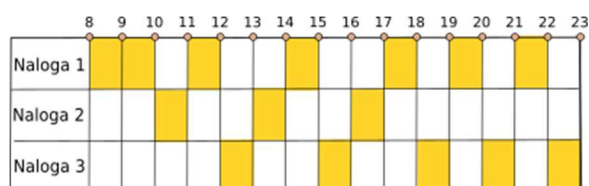


Robot lahko izvede različne naloge, za katere potrebuje 1, 2, 3 ali več ur. V vsaki uri lahko robot opravlja le eno nalogo. Ob koncu vsake ure robot preveri, če je dobil novo nalogo.

- Če jo je dobil, mora takoj začeti opravljati novo nalogo.
- Če ne, nadaljuje z nalogo, ki je najdlje časa ni opravljal.

Naslednja časovnica prikazuje primer robotovega dela v enem dnevu:

- 8.00 prejem naloge 1, ki traja 7 ur
- 10.00 prejem naloge 2, ki traja 3 ure
- 12.00 prejem naloge 3, ki traja 5 ur



Nalogo 1 je končal ob 22.00, nalogo 2 ob 17.00 in nalogo 3 ob 23.00.

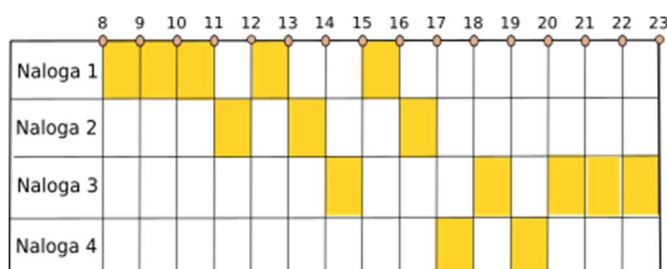
Nekega drugega dne je dobil naslednje naloge:

- 8.00 prejem naloge 1, ki traja 5 ur
- 11.00 prejem naloge 2, ki traja 3 ure
- 14.00 prejem naloge 3, ki traja 5 ur
- 17.00 prejem naloge 4, ki traja 2 uri

Kdaj konča nalogo 4?

Rešitev

Nalogo 4 konča ob 20.00.

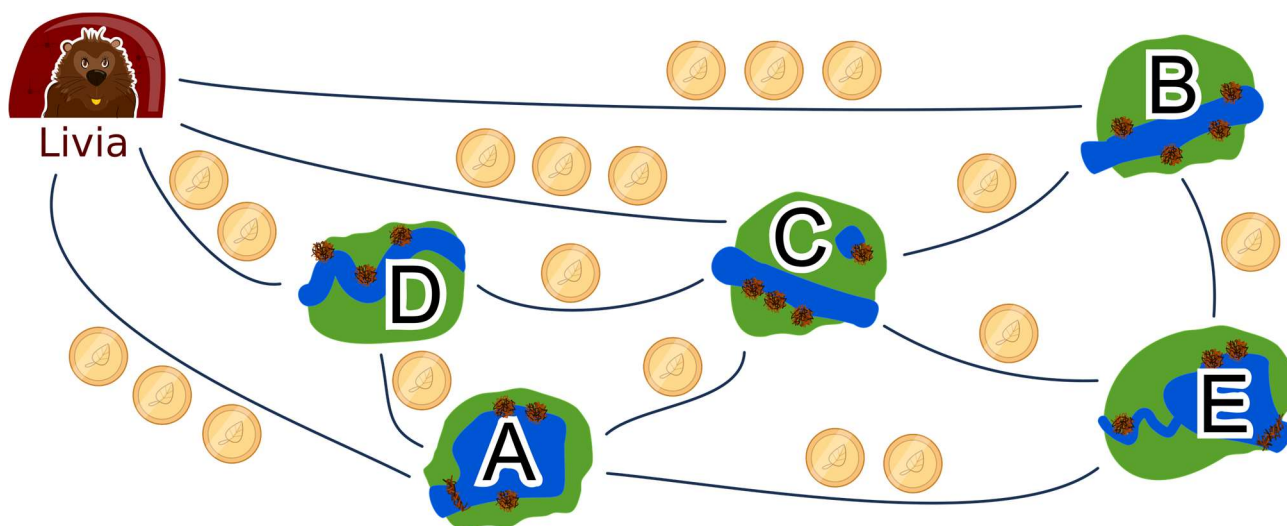


Računalniško ozadje

Ena pomembnejših nalog operacijskega sistema je razvrščanje nalog, ki jih mora opraviti procesor. Za ta namen obstajajo različni algoritmi, ki zagotavljajo, da procesorju ne zmanjka dela.



Livia želi z javnim prevozom obiskati vse svoje prijatelje v vaseh A, B, C, D in E. Vse svoje prijatelje želi obiskati v enem dnevu, ne da bi se vmes vrnila domov ali kakšno od vasi obiskala dvakrat. Cena vozovnice za posamezno linijo je zapisana na spodnji sliki. Potovanje na posamezni liniji lahko stane največ tri kovance.



Livia lahko svoje prijatelje obišče na primer takole: dom $\rightarrow$ B $\rightarrow$ E $\rightarrow$ A $\rightarrow$ D $\rightarrow$ C $\rightarrow$ dom. Za to bi morala plačati 11 kovancev.

Se da pot izvesti ceneje? Najmanj koliko bo Livia plačala za svoje potovanje k prijateljem?

Rešitev

Za najcenejše potovanje od prijatelja do prijatelja bo Livia plačala devet kovancev. Taki poti sta dom $\rightarrow$ B $\rightarrow$ E $\rightarrow$ C $\rightarrow$ A $\rightarrow$ D $\rightarrow$ dom in dom $\rightarrow$ D $\rightarrow$ A $\rightarrow$ C $\rightarrow$ E $\rightarrow$ B $\rightarrow$ dom. Vse ostale poti so dražje, stanejo bodisi deset bodisi enajst kovancev.

Računalniško ozadje

V ozadju naloge je znani problem trgovskega potnika, ki mora kar najceneje obiskati vsa zahtevana mesta, ne da bi pri tem obiskal katerokoli mesto več kot enkrat. Take naloge so časovno zelo zahtevne za računalnik.

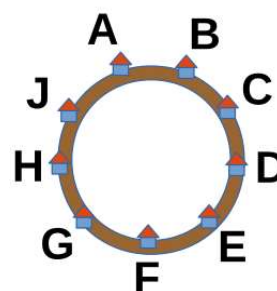
Hugo živi v majhni vasici z 9 hišami, ki so razporejene okoli jezera in povezane s krožno potjo. Vhoda v sosednji hiši sta vedno 10 metrov narazen.

V času pusta se je Hugo našemil in šel od hiše do hiše obiskat svoje sovaščane.

Pustni sprehod je začel pred hišo H in si izbral smer, v katero je hodil, dokler se ni odločil obiskati hiše. Ko je odšel iz hiše, si je ponovno izbral smer in v to smer hodil, dokler se ni odločil obiskati naslednje hiše, in tako naprej. Vsakič ko je obiskal hišo, si je zapisal razdaljo, ki jo je prehodil od prejšnje hiše.

Hugo je končno prišel do hiše F. Na njegovem listu je pisalo **20 10 20 40**.

Katere tri hiše je Hugo obiskal na svoji poti do hiše F?

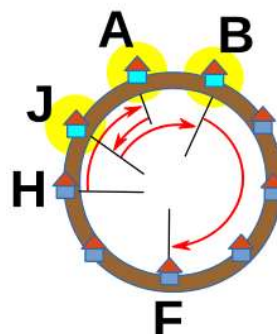


Rešitev

Hugo je na svojem pustnem sprehodu najprej obiskal hišo A, nato hišo J in nato hišo B.

Računalniško ozadje

Hugov zapis predstavlja podatke, iz katerih zlahka izvemo, kolikšno razdaljo je prehodil. Ta zapis pa ni najboljši za predstavitev, katere hiše je obiskal. Boljši zapis bi lahko bil že, če bi Hugo številkam dodal predznak kot kodo za smer, v katero je šel.





Angelo si je kupil robota za sajenje sadik, ki jih želi posaditi po robu svojega vrta.

Robotov programski jezik vsebuje naslednje ukaze:

Začni: prižge robota.

Naprej(X): robot se premakne X metrov naprej.

Nazaj(X): robot se premakne X metrov nazaj.

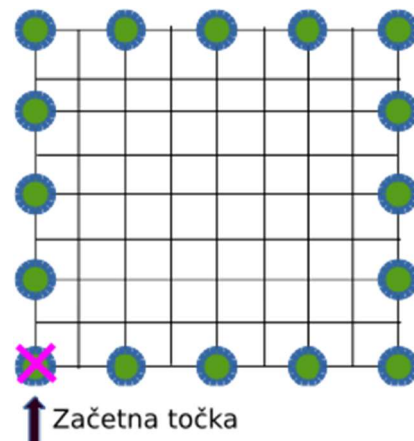
Levo: robot se obrne levo (na mestu, brez premika!).

Desno: robot se obrne desno.

Posadi: robot posadi sadiko.

Ponovi X {ukazi}: robot ponovi ukaze v oklepaju X-krat.

Stop: robot se ugasne.



Angelo želi posaditi 16 dreves na mesta, ki so označena na sliki. Vrt je dolg 8 metrov in širok 8 metrov. Sadike želi posaditi po robu svojega vrta tako, da bodo 2 metra narazen.

Na sliki je označeno mesto, na katerem se na začetku nahaja robot, in smer, v katero je obrnjen. Robot je ugasnjen in se mora, ko konča svojo nalogo, ponovno ugasniti. Ko robot posadi sadiko, se lahko neovirano premakne naprej.

Katerega od naslednjih programov mora izvesti robot, da bo posadil vse sadike, kot kaže slika?

A.

Začni

Ponovi 4{

 Ponovi 4{Posadi, Naprej(2)},

 Desno}

Stop

B.

Začni

Ponovi 4{

 Ponovi 4{Posadi, Naprej(2)},

 Levo}

Stop

C.

Začni

Ponovi 4{

 Ponovi 4{Naprej(1), Posadi},

 Desno}

Stop

D.

Začni

Ponovi 4{

 Ponovi 4{Naprej(2), Posadi},

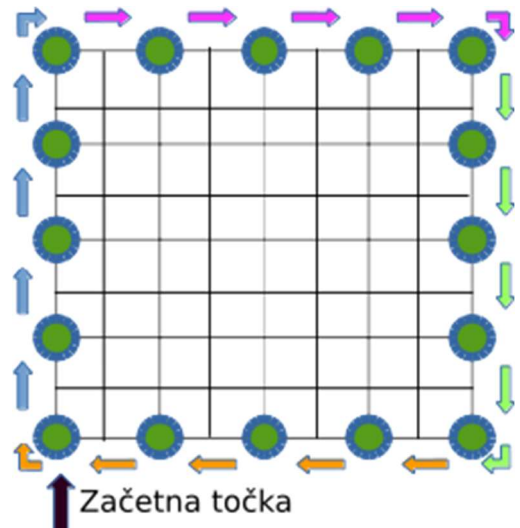
 Levo}

Stop

Rešitev

```
Začni
Ponovi 4 {Posadi, Naprej(2)},
Desno}
Stop
```

Robot bo najprej z ukazi `Ponovi 4{Posadi, Naprej(2)}` posadil 4 drevesa nad njim, nato se bo obrnil `Desno`. Zaradi zanke `Ponovi 4` bo te ukaze nato ponovil še 3-krat.



Odgovor C ni pravilen, ker robot drevesa sadi 1 meter narazen (`Naprej(1)`).

Odgovora B in D nista pravilna, ker se robot obrača Levo, kar pomeni, da bi zavil in se premaknil izven vrta.

Računalniško ozadje

V nalogi sledimo ukazom in preverjamo, če robot, ki jih izvaja, opravi zastavljeno nalogo.

Preverjanju pravilnosti programa rečemo tudi testiranje programa, iskanje in odpravljanje napak v programu pa imenujemo razhroščevanje (angl. debugging).



Ker bobri nočejo, da bi jih razumeli plenilci, so iznašli poseben način komunikacije.

V eno vrstico bobri zapišejo abecedo s črkami v običajnem vrstnem redu. V naslednji vrstici pa najprej zapišejo skrivno kodo, nato pa preostale črke abecede v običajnem vrstnem redu. Vsaka črka se lahko v eni vrstici pojavi natančno enkrat.

Recimo, da bo njihova skrivna koda beseda »BOBER«. Ta vsebuje dve črki B, zato obdržimo samo prvo in zapišemo »BOER«, za tem pa po vrsti še preostale črke abecede.

Abeceda:

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

Abeceda s skrivno kodo:

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

S pomočjo te kode lahko črke iz abecede zamenjamo s črkami iz abecede s skrivno kodo. Tako besedo NALOGA zapišemo kot: LBJMDB.

Če prejmemo skrivno sporočilo, ga lahko razumemo, ko ga z obratnim postopkom prevedemo v običajno abecedo.

Če uporabimo skrivno besedo »ŽIRAFa«, kako bi zakodirali besedo LISICA?

Rešitev

Kot je razvidno iz vrstice abecede s skrivno kodo je pravilen odgovor JEPERŽ.

Abeceda:

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

Abeceda s skrivno kodo

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

Računalniško ozadje

V nalogi je predstavljen primer šifriranja s substitucijsko oz. zamenjalno šifro. Poslano sporočilo lahko dešifrirajo le tisti, ki poznajo skrivno besedo, ki je bila uporabljena pri šifriranju sporočila, zato to ostane skrito pred plenilci.



Magična škatla staplja kristale v nove kristale. Pri tem sledi naslednjim pravilom:

- Hkrati lahko škatla stopi le dva kristala, ki sta različne oblike in različne barve.
- Pri stapljanju nastane nov kristal, ki je drugačne oblike in barve.

$$\triangle + \square = \bigcirc$$

$$\square + \pentagon = \hexagon$$

$$\triangle + \pentagon = \heptagon$$

$$\square + \hexagon = \trapezoid$$

$$\triangle + \hexagon = \diamond$$

	+		=	
	+		=	
	+		=	
	+		=	
	+		=	

- Stapljanje se lahko zgodi večkrat.

Na začetku imamo naslednje kristale:

rumen trikotnik



moder kvadrat



rdeč trikotnik



rumen petkotnik



Katerega od spodnjih kristalov **ne moremo** dobiti iz njih po stapljanju?

Zelen krog



Črn romb



Črn trapez



Vijoličen krog



Rešitev

Pravilen odgovor je

A) + =

B) + = ; + =

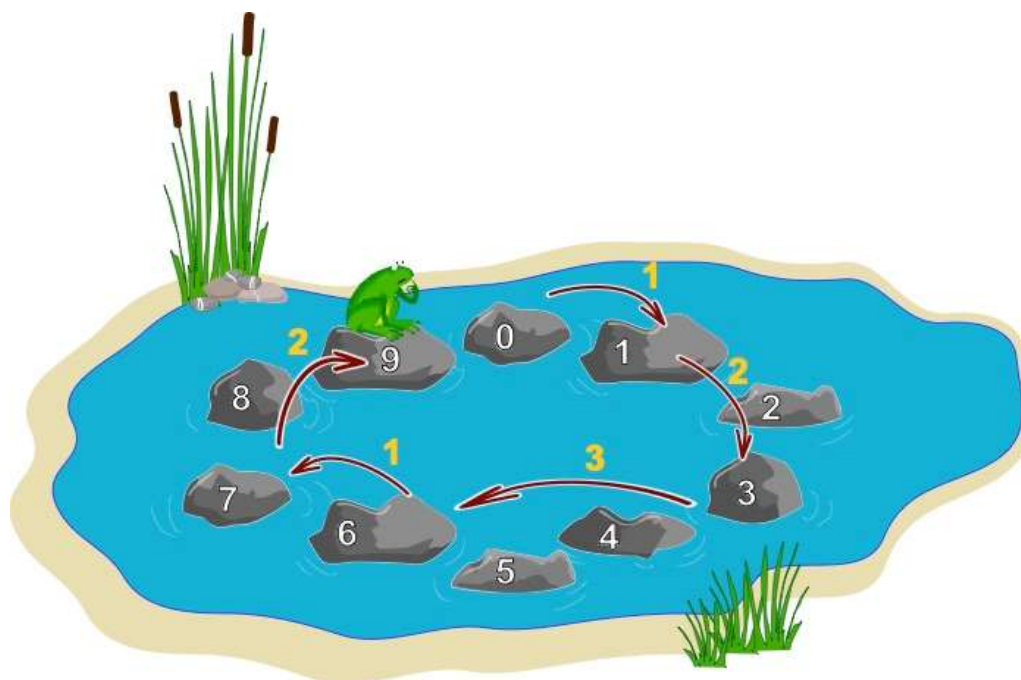
D) + =

Računalniško ozadje

Od vhodnih podatkov je odvisno, kaj bomo dobili kot izdelek po stapljanju. Podobno v računalništvu na podlagi določenih vhodnih podatkov dobimo določen rezultat.

Nastop Roka Skoka bi lahko opisali z eno samo besedo, vendar jo bomo z dvema: *popolna katastrofa*.

V prestižni disciplini "1 – 2 – 3 – hop" žabe skačejo po desetih kamnih, ki so postavljeni v krogu. Začnejo na kamnu 0. Najprej skočijo en kamen daleč, nato dva,



nato tri, pa spet enega, dva, tri, enega, dva, tri ... Tako krožijo po kamnih, dokler ne omagajo.

Rok je omagal po borih petih skokih (1, 2, 3, 1, 2) in žalostno občepel na kamnu številka 9. Če bi zmoget le še en samo skok, za 3 kamne, bi končal na kamnu 2.

Kako zanikrna je bila Rokova predstava (kdo ga je spustil na tekmo?!), pove podatek, da je naslednji tekmovalec, Peter Skok (Rokov bratranec po očetovi strani) naredil kar 37 skokov.

Saj res, na katerem kamnu je zaključil skakanje Peter?

Rešitev

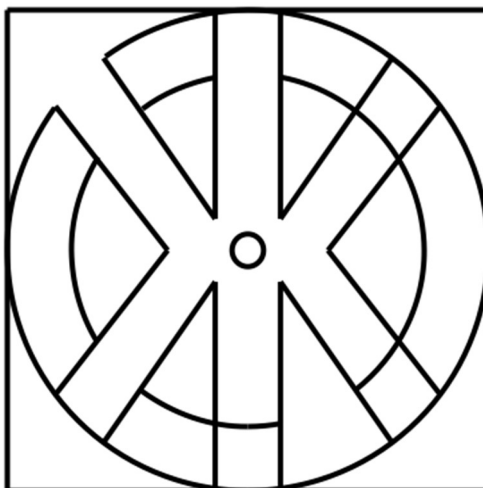
Peter je skočil 37-krat. To pomeni, da je naredil 12-krat po 3 skoke in še 1 skok. V treh skokih preskoči $1 + 2 + 3 = 6$ kamnov. Torej je preskakal $12 \times 6 = 72$ kamnov in še enega zraven. Skupaj torej 73 kamnov. V krogu je 10 kamnov, torej se po vsakih desetih kamnih vrne na kamen 0. 73 kamnov je 7×10 in še 3 zraven. Končal bo torej na kamnu 3.

Mimogrede lahko povemo še, da je sedemkrat ($73 / 10$) skočil na ali preko kamna 0. Če bi ravno koga zanimalo.

Računalniško ozadje

V nalogi smo računali z ostanki, najprej po deljenju s 3, potem po deljenju z 10. Zraven smo malo seštevali. Takšna preračunavanja so v računalništvu zelo pogosta.

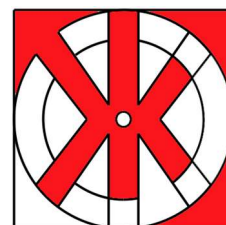
Spodnji vzorec moramo pobarvati tako, da dve sosednji območji (tj. območji, ki si delita isti rob) ne bosta enake barve.



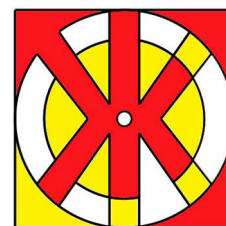
Najmanj koliko barv potrebujemo?

Rešitev

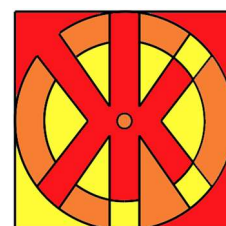
Obstaja več možnih načinov, kako pobarvati vzorec, odvisno od tega, katero barvo izberemo prvo in kateri del vzorca najprej pobarvamo. Prikazana je ena od možnih rešitev. Vzeli smo prvo barvo (rdečo) in v vzorcu pobarvali vse dele, ki smo jih lahko (začeli smo v levem zgornjem kotu).



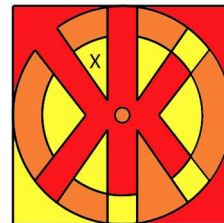
Nato vzamemo drugo barvo (rumeno) in pobarvamo toliko delov, kot jih lahko; pri tem začnemo barvati v spodnjem levem kotu.



Ko izberemo še tretjo barvo (oranžno), lahko z njo pobarvamo vse preostale dele slike, ne da bi pri tem kršili pravilo glede dveh sosednjih območij.



S tem smo pokazali, da tri barve zadoščajo. Sedaj moramo še ugotoviti, da problema ne moremo rešiti z manj kot tremi barvami. Poglejmo območje, ki je označeno z X in pobarvano rumeno. To območje meji na dve različni območji, torej imamo tri območja, od katerih mora biti vsako v svoji barvi. Vzorca torej ne moremo pobarvati z manj kot tremi barvami.



Računalniško ozadje

Barvanje takšnih reči – in njegova splošna različica, ki ji pravimo barvanje grafov – uporabljamo za reševanje številnih problemov, od sestavljanja urnikov in dodeljevanja frekvenc za mobilna omrežja do dodeljevanja koridorjev letenja posameznim letalom tako, da se izognemo trčenjem med pristajanjem ali vzletanjem.

Pri tej nalogi je zanimivo še to, da lahko podano dvodimenzionalno obliko (na primer zemljevid), ki je razdeljena na poljubno število območij poljubnih oblik, vedno pobarvamo z uporabo največ štirih barv, ne da bi prekršili pravilo, da morata biti dve sosednji območji vedno različnih barv.



Bobri krasijo torte tako, da je vsaka drugačna. Na vsako torto zato postavijo drugačno kombinacijo svečk. Na voljo imajo rumene in rdeče svečke. Na vsaki torti mora biti vsaj ena svečka. Pomembno je tudi, kakšno je zaporedje svečk na tortah. Na primer, če damo na torto najprej rdečo svečko in nato rumeno, je to drugače, kot če damo najprej rumeno in nato rdečo, ne glede na to, da je na vsaki torti ena rumena in ena rdeča svečka.



Bobri želijo s kar najmanj svečkami okrasiti 14 različnih tort, zato začnejo krasiti torte najprej z eno, nato z dvema, nato tremi svečkami in tako dalje.

Koliko svečk bodo morali uporabiti, da okrasijo 14 tort?

Rešitev

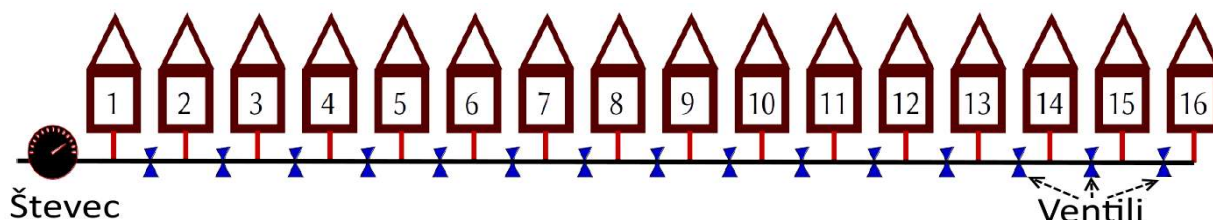
Potrebovali bodo 34 svečk: prvi 2 torti okrasijo vsako s po eno svečko (2 svečki), naslednje 4 torte lahko okrasijo s po dvema svečkama na vsaki torti (8 svečk), naslednjih 8 tort lahko okrasijo s po 3 svečkami (24 svečk). To je ravno 14 tort, torej potrebujejo $2 + 8 + 24 = 34$ svečk.

Računalniško ozadje

V tej nalogi računamo, koliko števil lahko v dvojiškem številskem sestavu zapišemo z eno števk (0 in 1), z dvema števka (00, 01, 10 in 11) in koliko s tremi števki (000, 001, 010, 011, 100, 101, 110 in 111).



Vodovodno podjetje je ugotovilo, da pušča vodovodna napeljava do ene od 16 hiš v isti ulici. Sedaj njihov serviser poskuša ugotoviti, pri kateri hiši pušča. Pri vseh hišah mu pomagajo pri iskanju napake tako, da so zaprli vodo v svojih hišah.



Serviser bo zaprl ventil med dvema hišama in na števcu preveril, ali še kaže porabo vode. Na primer, če zapre ventil med hišama na številki 8 in številki 9 in če števec še vedno kaže porabo vode, bo serviser vedel, da mora puščati nekje med hišama na številkah 1 in 8 (in da ne pušča med hišama na številkah 9 in 16).

Če predpostavimo, da vodovod pušča le na enem mestu pri priključku do hiše (rdeči navpični deli na sliki), kakšno je najmanjše število ventilov, ki jih mora serviser zapreti, da z gotovostjo najde priključek, ki pušča?

Rešitev

Če je napaka lahko na kateremkoli priključku, je najbolje, da se pri iskanju napake ne zanašamo na srečo, ampak jo poiščemo sistematično: področje iskanja razdelimo na dva enaka dela ter preverimo, v katerem delu je napaka. To ponavljamo, dokler ne izoliramo napake.

Najprej zapremo ventil med hišama 8 in 9. Če števec kaže porabo vode, je napaka v prvem delu ulice (hiše 1 do 8), sicer pa v drugem delu ulice (hiše od 9 do 16). V obeh primerih smo izločili 8 hiš, kjer sigurno ne pušča. Nato zapremo ventil na polovici med hišami tistega dela ulice, kjer pušča. Tako lahko izločimo še 4 hiše. Tretjič zapremo ventil in izločimo še dve hiši. Pri četrtem zapiranju ventila pa med preostalima dvema hišama določimo tisto, pri kateri pušča.

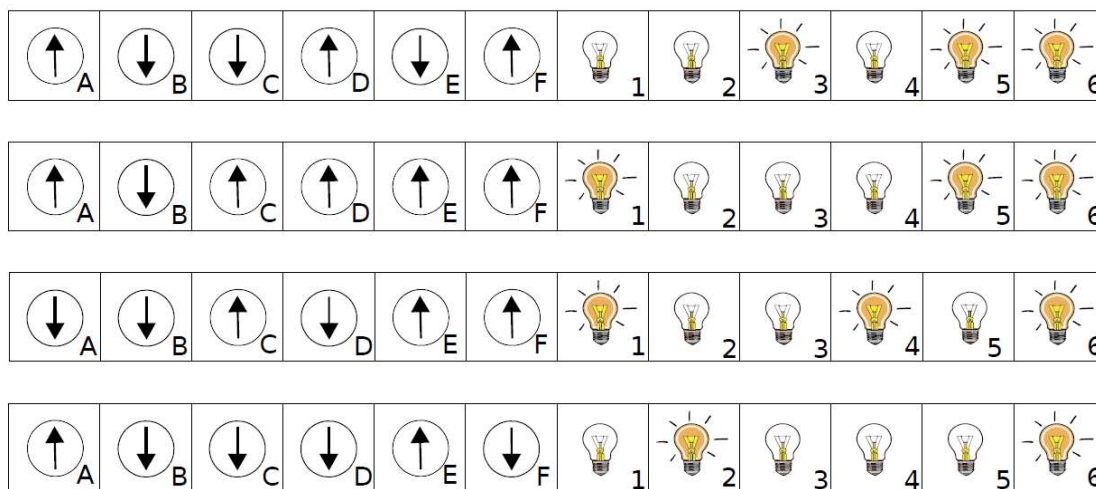
Računalniško ozadje

Naloga prikazuje praktičen primer iskanja z bisekcijo – razpolavljanjem, ki ga lahko s pridom uporabimo tudi izven računalništva, na primer za iskanje določene knjige v knjižnici.

Bisekcija je klasičen algoritem iskanja. Iskano vrednost primerjamo s srednjim elementom v urejenem seznamu: če nista enaka, obdržimo le polovico seznama, v katerem leži iskani element. Isto operacijo nato ponovimo na preostalem seznamu, dokler ne najdemo iskanega elementa ali pa nam ostane le še prazen seznam (v tem primeru iskanega elementa ni v seznamu).

Bojan je povezal 6 žarnic s 6 stikali. Vsako stikalo vklaplja le eno žarnico, a ne vemo, katero. Ne poznamo niti položaja stikala, ko je vklopljeno ali izklopljeno.

Da bi ugotovili, katero stikalo je povezano s katero žarnico, smo se poigrali s stikali. Na sliki lahko vidite rezultate naših štirih poizkusov (tj. položajev stikal in prižganih oz. ugasnjenih žarnic):



Katero stikalo je povezano s katero žarnico?

Rešitev

Pravilen odgovor je 1 – C, 2 – F, 3 – E, 4 – A, 5 – D, 6 – B.

- Razlika v prvi in drugi sliki je pri stikalih C in E ter pri žarnicah 1 in 3. Torej vemo, da so ti povezani: 1 – C in 3 – E ali pa 1 – E in 3 – C.
- Med prvo in četrto sliko je razlika pri žarnicah 2, 3 in 5 ter pri stikalih D, E in F. Vemo tudi, da je žarnica 3 povezana ali s stikalom C ali s stikalom E. Iz tega sledi, da je edina možna kombinacija povezava žarnice 3 s stikalom E. Torej imamo en par 3 – E (in seveda tudi par 1 – C).
- Med drugo in tretjo sliko so razlike pri žarnicah 4 in 5 ter pri stikalih A in D. Vemo tudi, da je žarnica 5 lahko povezana le s stikalom D ali F. Torej mora biti žarnica 5 povezana s stikalom D (imamo še par 5 – D). Pri primerjavi obeh slik nam preostane tako le še povezava 4 – A.
- Ker smo že odkrili povezavo 5 – D, ostane pri primerjavi prve in četrte slike še par 2 – F.
- Na koncu nam ostale le še en možen par: 6 – B.

Računalniško ozadje

V računalniških sistemih je zelo pomembno sledenje delovanju sistema ali programa. Kadar delamo s kodo ali s sistemom, moramo razumeti, kako deluje, da ga lahko spreminjamo ali razvijamo novo kodo ali sistem.

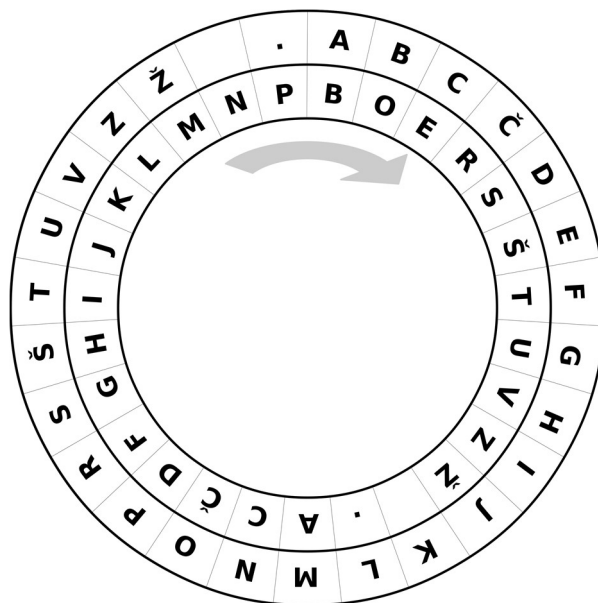
Razhroščevanje (tj. iskanje vzrokov za hrošče ali napake, ki povzročijo neželeno vedenje) je tudi pomemben proces v razvoju kode ali sistema. Napake poiščemo s postavljanjem hipotez in manjšimi kontroliranimi eksperimenti, ki pokažejo na napako.



Za varen prenos podatkov so si Bobri izmislili naslednje kodiranje. Uporabijo dva kroga, na vsakem so zapisane vse črke abecede ter presledek in pika.

Za kodiranje sporočila vsak znak posebej z zunanjega kroga zamenjamo z znakom z notranjega kroga. Po kodiranju vsakega znaka notranji krog zavrtimo za en znak v smeri urinega kazalca:

Prvi znak kodiramo glede na začetni položaj krogov (na sliki). Na primer, prvo črko S zamenjamo s črko G. Če je druga črka sporočila, ki ga želimo kodirati, L, jo zamenjamo s presledkom in tako naprej. Če bi na primer želeli kodirati besedo POR, bi ta postala DCC.



V katerega od spodnjih nizov znakov je zakodirano sporočilo »PRIDI OB TREH.«?

- A. DFZSZ ČO IFŠV.
- B. DFZSZNČOPIFŠVP
- C. DDUOŠIZJF UJLA
- D. .DVŠŽPGGŠOMMD_

Rešitev

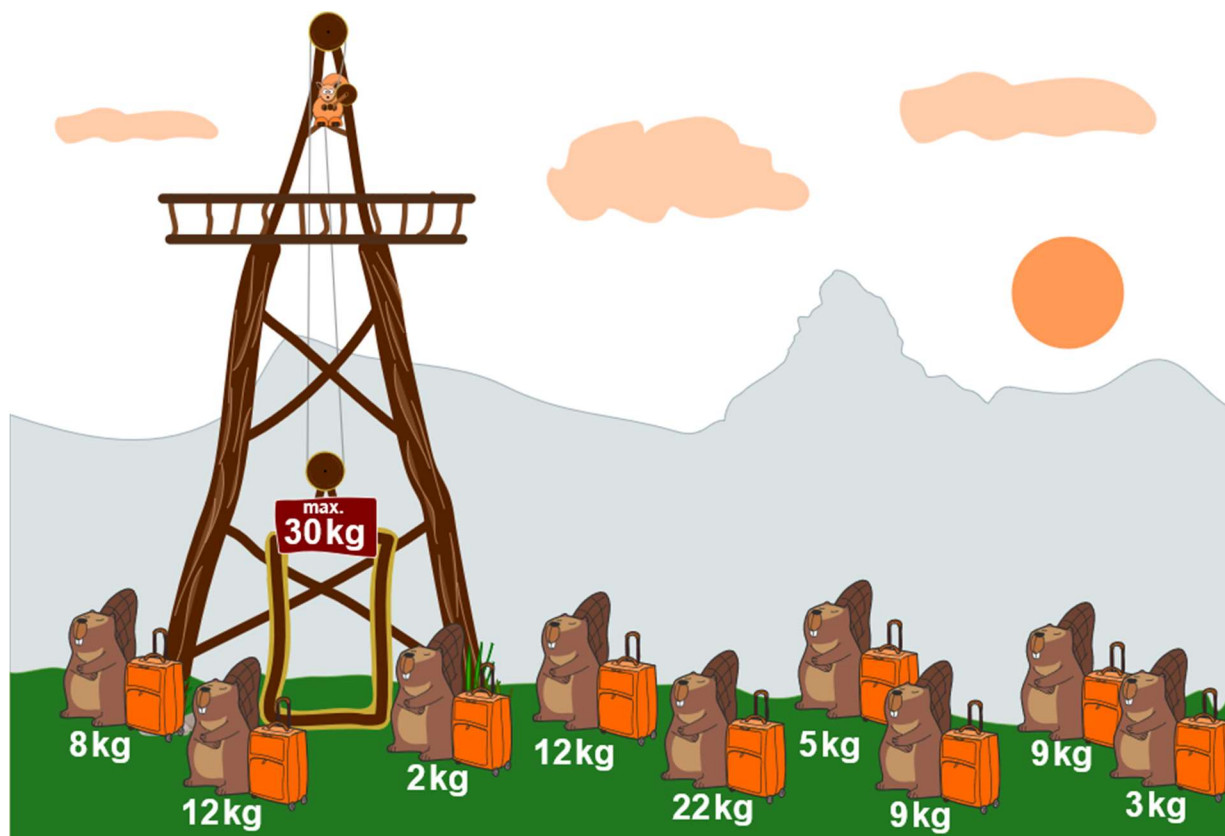
Pravilno zaporedje znakov je zaporedje C. Prva črka P se zakodira v črko D. Nato notranji krog zavrtimo za en znak v smeri urinega kazalca. Črko R tako zakodiramo v črko D. Zopet notranji krog zavrtimo v smeri urinega kazalca. Črko I nato zakodiramo v črko U. Postopek nadaljujemo do konca.

Računalniško ozadje

Kriptiranje, ki ga uporabljamo v tej nalogi, se imenuje Trithemiusovo kodiranje. Očitno je malenkost bolj zapleteno od Cezarjevega, v katerem vsako črko zamenjamo s črko, ki je oddaljena za določeno, vnaprej dogovorjeno število mest v abecedi, a manj od Vigenerejevega, v katerem se razdalje spreminjajo po dogovorjenem ključu. Vsa takšna kriptiranja smo v bobrovih nalogah že večkrat videli, zato gotovo veš tudi, da so danes primerna le še za otroško igro, saj je sporočila, skrita na ta način, s sodobnimi računalniki (pa tudi brez njih) precej preprosto razkriti.



Skupina devetih bobrov si ogleduje podeželje in se želi z dvigalom povzpeti na razgledni stolp. Ker je že pozno, ima dvigalo samo še dve vožnji. Dvigalo lahko obremenimo z največ 30 kg.



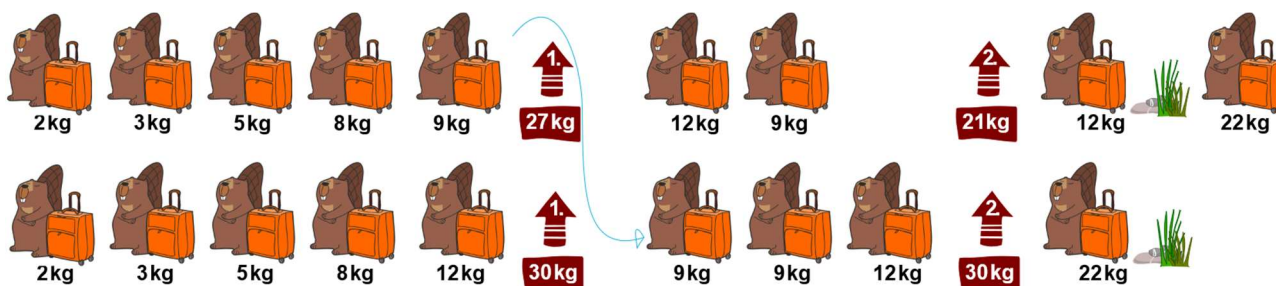
Kako naj razporedimo bobre na prvo in drugo vožnjo, da bo čim več bobrov lahko prišlo do razgledne ploščadi? Teža posameznega bobra skupaj z njegovo prtljago je zapisana ob njem (če še ne veste, bobri se nikoli ne ločijo od svoje prtljage).

Rešitev

Ker želimo v dvigalo spraviti čim več bobrov, je prva rešitev, ki nam pride na pamet, da v dvigalo damo bobre z najmanjšimi težami. Tako lahko bobre uredimo po teži in jih po vrsti razporejamo v dvigalo. Na prvo vožnjo spravimo: $2 + 3 + 5 + 8 + 9 = 27$ kg. Na drugo vožnjo pa nato prideta še dva bobra: $9 + 12 = 21$ kg. Skupaj bi tako na razgledno ploščad prišlo sedem bobrov ($5 + 2 = 7$), preostala dva (najtežja) pa ne.

Vendar lahko hitro ugotovimo, da imamo v prvi vožnji še 3 kg prostora do največje teže, ki jo zmore dvigalo, na drugi vožnji pa kar 9 kg. Bi lahko obe skupini bobrov preuredili tako, da bi skupaj v dvigalo spravili 8 bobrov? Seveda. Če bobra z 9 kg iz prve vožnje prestavimo v drugo, bomo na drugi vožnji dosegli 30 kg. Na prvi vožnji pa imamo tako prostor še za 12 kg ($3 + 9 = 12$), kar ravno ustreza naslednjemu bobru v vrsti. Tako lahko na razgledno platformo spravimo skupaj osem

bobrov (5 v prvi vožnji in 3 v drugi vožnji) in s tem tudi zapolnimo vse kapacitete dvigala. Deveti (najtežji) bober pa bo žal moral ostati pred stolpom.



Računalniško ozadje

V podanem problemu smo imeli zelo veliko možnih rešitev in je zato praktično nemogoče, da bi vse preverili v sprejemljivem času. Zato so taki problemi praktično nerešljivi. Vendar pa lahko z nekaj znanja o problemu poskusimo najti neko dovolj dobro rešitev problema, ki pa ni nujno optimalna rešitev.

V naši nalogi smo pri razvrščanju bobrov sledili preudarnemu načrtu: na prvo vožnjo dvigala smo poskusili spraviti čim večje število bobrov, zato smo izbrali bobre z najmanjšo težo. Tak načrt, ki vodi do dobre rešitve, čeprav morda ne najboljše, v računalništvu imenujemo heuristika.

Jamb je igra s petimi kockami, ki jih mečemo trikrat v vsaki rundi. Po metu kock se igralec za vsako kocko lahko odloči, da jo obdrži ali pa jo ponovno vrže. Če po treh metih vse kocke kažejo enako številko, igralec dobi »jamb«.

Slika prikazuje možen izid ene igre:

Met 1					Met 2			Met 3		Meti naslednjega igralca				
														
	Obdrži		Obdrži			Obdrži		Obdrži	Obdrži					

V tem primeru je igralec v prvem metu vrgel kocke     . Nato je obdržal obe kocki, ki sta imeli enako številko, to je 4. V naslednjem metu je vrgel preostale tri kocke in obdržal kocko s številko 4. V tretjem metu (zadnjem metu v rundi) je igralec vrgel preostali dve kocki in obe sta pokazali 4. Tako je igralec vrgel pet enakih števil in s tem dobil jamb.

Berta si je zapisovala mete kock v igri, v kateri je uspela dobiti jamb v natanko 3 metih. Vendar pa je na koncu pomešala zapiske svojih metov in metov njenih soigralcev.

Če predpostavimo, da je Berta v igri vedno obdržala kocke s številkami, ki so se pojavile največkrat, kateri od zapisov prikazuje igro, v kateri je dobila jamb (zapisi se začnejo z Bertinim prvim metom)?

- A.               
- B.               
- C.               
- D.               

Rešitev

Pravilen odgovor je B:

Met 1					Met 2			Met 3	Meti naslednjega igralca					
														
		Obdrži	Obdrži		Obdrži		Obdrži	Obdrži						

V zaporedju metov kock pod B je igralec v prvem metu vrgel dve šestici in ju obdržal. Nato je v drugem metu vrgel tri kocke in ponovno dobil dve šestici. Te je tudi obdržal. V tretjem metu je vrgel še preostalo kocko in dobil šestico ter s tem tudi jamb. Meti so prikazani na zgornji sliki.

Odgovor A:

Met 1					Met 2			Met 3		Meti naslednjega igralca				
		Obdrži	Obdrži				Obdrži	Obdrži						

V zaporedju metov pod A je igralec v prvem metu dobil dve šestici, ki ju je tudi obdržal. V drugem metu je vrgel tri kocke in dobil še eno šestico. V tretjem metu je vrgel še preostali dve kocki in dobil šestico in enko. Tako je imel po treh metih le štiri šestice in je ostal brez jamba.

Odgovor C:

Met 1					Met 2			Met 3		Meti naslednjega igralca				
		Obdrži	Obdrži		Obdrži	Obdrži								

V zaporedju metov pod C je igralec v prvem metu dobil dve šestici in ju obdržal. V drugem metu je vrgel tri kocke in dobil še dve šestici. V tretjem metu pa je vrgel še eno kocko in dobil trojko. Tako je skupaj v treh metih dobil le štiri šestice in je ostal brez jamba.

Odgovor D:

Met 1					Met 2			Met 3		Meti naslednjega igralca				
		Obdrži	Obdrži				Obdrži	Obdrži						

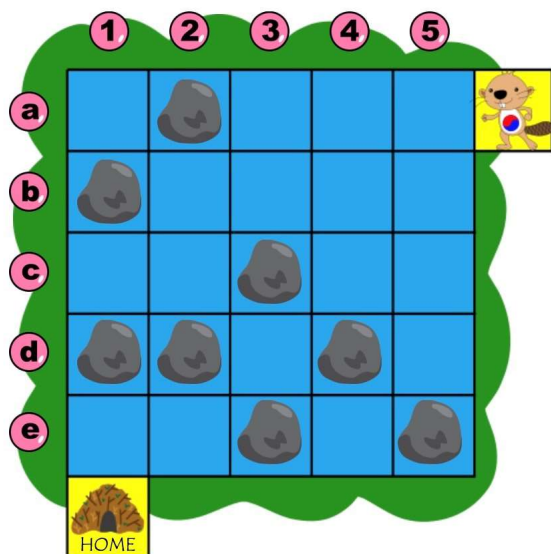
V zaporedju metov pod D je igralec v prvem metu dobil dve šestici in ju obdržal. V drugem metu je vrgel tri kocke in dobil še eno šestico. V tretjem metu je vrgel preostali dve kocki in dobil še eno šestico. S skupaj štirimi šesticami je ostal brez jamba.

Računalniško ozadje

Veliko problemov, kot je tudi ta v naši nalogi, lahko opišemo z modelom vhod-obdelava-izhod (angleško *input-processing-output* ali IPO). Vhod je zaporedje izidov meta kocke, ki ga moramo obdelati, da dobimo izhod, ki pove, ali je bil dosežen jamb ali ne.

Most iz kamnov

državno, 6. – 9. razred in srednja šola



V priljubljeni korejski igri je glavni junak bober Jin, ki želi priti do svojega doma, zato moramo zgraditi most iz kamnov od njega do doma (HOME). Jin se lahko premika levo, desno, gor ali dol, ne more pa se premakniti po diagonali.

V igri pritisnemo gumb s številko in v ustreznem stolpcu se kamni potopijo ali dvignejo iz vode (kamen se nahaja na vsakem polju; če je kamen v polju nad vodo, se potopi, če pa je pod vodo, se dvigne nad gladino). Če pritisnemo gumb s črko, se enako zgodi s kamni v ustrezni vrstici.

Najmanj koliko gumbov moramo pritisniti, da

zgradimo most od Jina do doma?

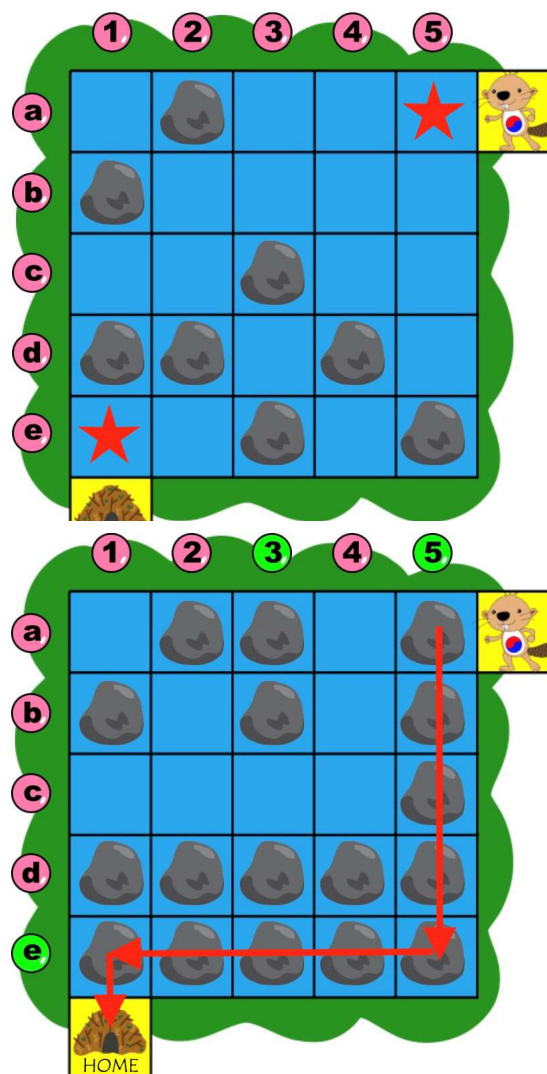
Rešitev

Pritisniti moramo gumbe 5, e in 3.

Takoj lahko vidimo, da je vseh možnih zaporedij pritisnjenih gumbov veliko preveč, da bi lahko posebej preverili vsakega od njih pri iskanju prave rešitve. Zato moramo problem podrobneje analizirati in rešitev poiskati s pomočjo sklepanja.

Najprej lahko ugotovimo, da moramo nad gladino postaviti oba kamna na mestih, ki sta označeni z rdečo zvezdo. Torej mora biti v rešitvi kombinacija gumba a ali 5 in gumba e ali 1. Možne kombinacije so torej štiri: [a, 1], [a, e], [5, 1] in [5, e].

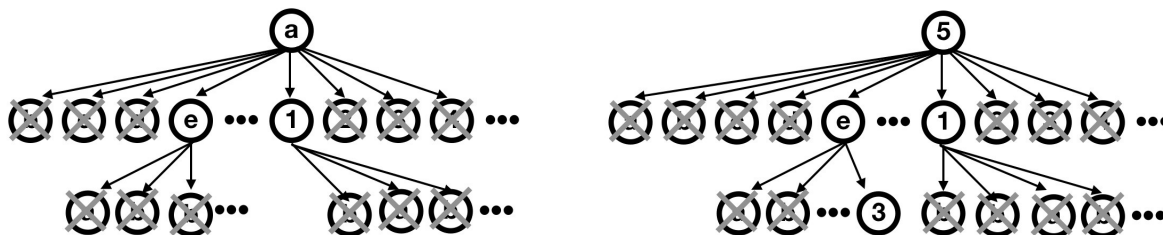
Vendar nobena od teh štirih možnosti ne omogoča izgradnje celega mosta. Torej le z dvema gumboma ne moremo dobiti rešitve. Še najbližje rešitvi je kombinacija [5, e], pri kateri nam manjka le en kamen v spodnji vrstici na sredini (na poziciji e3), da bi imeli zgrajen cel most. Ta kamen lahko postavimo s pritiskom na gumb 3. Torej lahko most zgradimo s tremi gumbi: [5, e, 3].



Računalniško ozadje

Iskanje v širino je algoritem za prehod preko drevesa ali grafa, pa tudi za iskanje v drevesu ali grafu. Začnemo v korenu drevesa (ali katerem koli vozlišču grafa) ter najprej raziščemo njegova sosednja vozlišča, nato pa se premaknemo na naslednji nivo (sosednja vozlišča sosedov).

V naši nalogi bi lahko kombinacije pritisnjenih gumbov predstavili z drevesom, kjer so vozlišča vsi gumbi, povezave pa predstavljajo kombinacijo teh gumbov. Iskanje lahko poenostavimo, če preiskujemo le tiste poti, ki zadoščajo pogojem. Na spodnji sliki vidimo, da lahko preiskujemo pot od a do e ali do 1, drugo pot od 5 do e ali do 1 in tako naprej.



Od vseh štirih poti je le pot 5 -> e -> 3 pravilen odgovor.



Mednarodno podatkovno omrežje sestavlja več nacionalnih omrežij, kot je prikazano na sliki spodaj. Vsako nacionalno omrežje ima vozlišča iste barve. Vozlišča z dvema različnima barvama označujejo stik med dvema različnima nacionalnima omrežjema. Pri gostovanju v tujem omrežju nastanejo dodatni stroški. Če torej sporočilo pošljemo iz enega v drugo omrežje, se strošek pošiljanja izračuna kot:

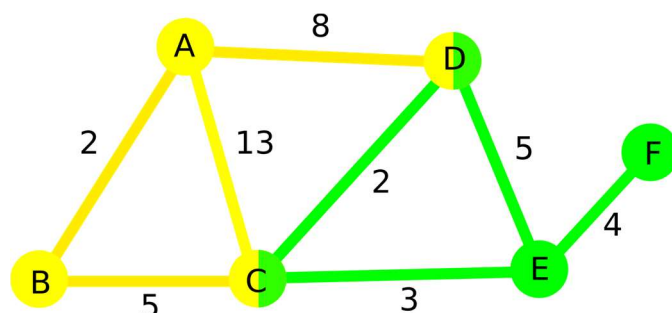
(vsota cen poti po izvornem omrežju) + $2 \times$ (vsota cen poti po drugih omrežjih)

Vsaka veja omrežja ima svojo ceno, prikazano na sliki.

Recimo, da je omrežje takšno, kot ga kaže slika na desni.

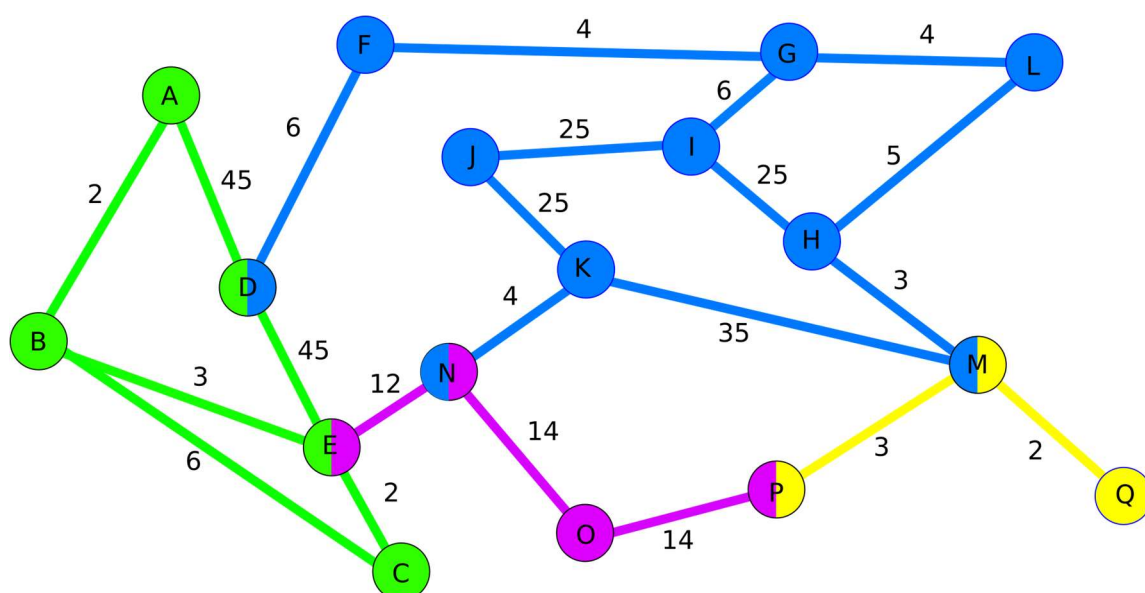
Kakšna je najcenejša pot od A do F?

A→B	B→C	C→E	E→F
2	5	3	4



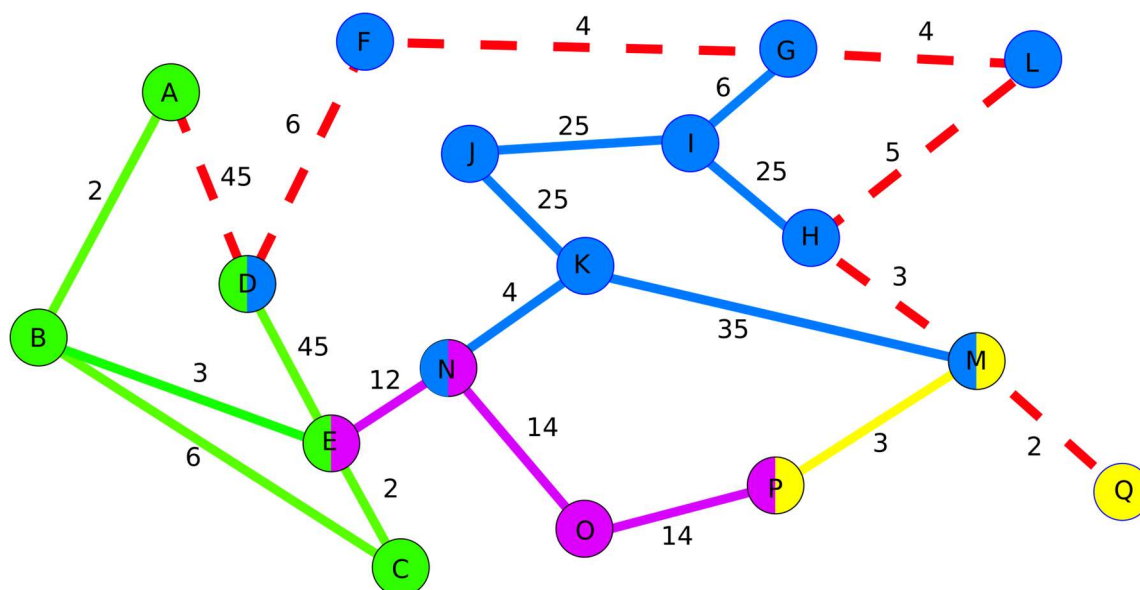
$$(2 + 5) + 2 \times (3 + 4) = 21.$$

Kakšen pa je najmanjši možen strošek pošiljanja sporočila od vozlišča A do vozlišča Q v spodnjem primeru?



Rešitev

Pravilen odgovor je 93.



Pri reševanju moramo biti pozorni na več pasti. Paziti moramo, da iz vozlišča ne uberemo vsakič najcenejše poti in da najkrajša pot ni vedno tudi najcenejša. Ob površnem reševanju bi tako lahko vzeli najkrajšo pot: $A \rightarrow B \rightarrow E \rightarrow N \rightarrow K \rightarrow M \rightarrow Q$, katere cena je $(2 + 3) + 2 \times (12 + 4 + 35 + 2) = 111$. Cenejša pot bi bila npr. $A \rightarrow B \rightarrow E \rightarrow N \rightarrow O \rightarrow P \rightarrow M \rightarrow Q$, katere cena je $(2 + 3) + 2 \times (12 + 14 + 14 + 3 + 2) = 95$. Na ti dve poti smo postali pozorni zaradi cenovno ugodne poti po zelenem omrežju. Vendar pa na ceno bolj vpliva vrednost poti po drugih omrežjih, ki je po poti $D \rightarrow F \rightarrow G \rightarrow L \rightarrow H \rightarrow M \rightarrow Q$ veliko cenejša kot prvi omenjeni. Vrednost te celotne poti znaša $(45) + 2 \times (6 + 4 + 4 + 5 + 3 + 2) = 93$ in je najcenejša.

Računalniško ozadje

Veliko problemov lahko prevedemo na problem iskanja najcenejših poti. Tudi tale je očitno eden od njih. Da, tudi za pošiljanje podatkov po internetu uporabljamo podobne algoritme kot v cestni navigaciji.

Štirje prijatelji obiskujejo večerno tehnično šolo. Izbirajo lahko med petimi predmeti:

- znanost (Z),
- tehnologija (T),
- inženirstvo (I),
- umetnost (U) in
- matematika (M).

Vsi so zelo zagreti študenti, zato vsak izbere več kot le en predmet:

- Anja želi študirati I, U in T,
- Borut želi študirati Z, T in U,
- Cirila želi študirati M in U,
- Davor želi študirati I, M in U.

Šola želi razporediti predmete na čim manj večerov v tednu. Pri tem pa morajo paziti tudi na to, da ima vsak študent največ en predmet na dan.

Katero je najmanjše število dni, v katerih šola lahko izvede vse predmete tako, da bo vsak od prijateljev lahko obiskoval svoje izbrane predmete?

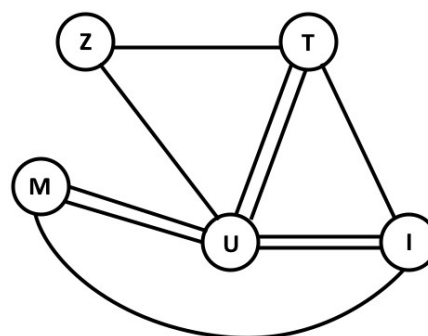
Rešitev

Pravilen odgovor je 3 dni.

Vse predmete moramo razdeliti po dnevih tako, da na isti dan ne bo predmetov, ki jih obiskuje posamezen študent. Opazimo lahko, da so vsi študenti izbrali umetnost (U). Zato mora imeti umetnost svoj dan na urniku. Po drugi strani pa je večina izbrala po tri predmete, zato bomo potrebujemo najmanj 3 dni za ustrezno razvrstitev predmetov. Z nekaj poskusov in sklepanja lahko pridemo do rešitve.

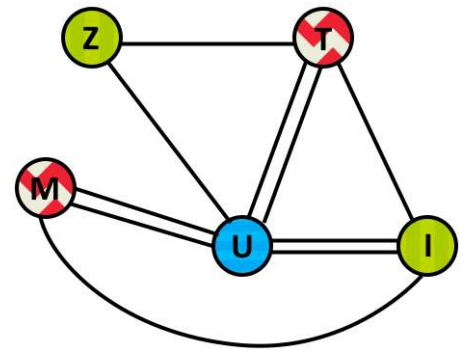
Problem pa lažje rešimo tako, da narišemo graf prekrivanj: vozlišča v grafu predstavljajo 5 predmetov, dve vozlišči pa sta povezani, če je nek študent izbral oba predmeta. Povezave tako določajo, da povezanih predmetov ne moremo izvajati v istem dnevu (predmeta se prekrivata).

V naslednjem koraku v grafu pobarvamo vsa vozlišča tako, da uporabimo čim manj barv. Pri tem morata biti dve povezani vozlišči vedno različnih barv.



Število barv, ki smo jih porabili za barvanje grafa, pove število dni, ki jih potrebujemo za izvedbo predmetov. Predmete iste barve lahko izvedemo isti dan. S slike lahko vidimo, da lahko na šoli sestavimo urnik, kjer izvedejo vse predmete v treh dneh, kot na primer:

- ponedeljek: umetnost (U),
- torek: matematika (M) in tehnologija (T),
- sreda: znanost (Z) in inženirstvo (I).



Računalniško ozadje

Problem v tej nalogi bi lahko rešili s podrobnim pregledom podatkov in logičnim sklepanjem, saj je dovolj majhen. Če pa problem povečamo na stotine študentov in predmetov ter več terminov v urniku, postane naenkrat preveč kompleksen, da bi ga lahko rešili brez ustreznega algoritma. Eden takih algoritmov je tudi barvanje grafa, ki smo ga uporabili pri razlagi rešitve te naloge. Barvanje grafa se uporablja za reševanje vrste težkih problemov v teoriji grafov. Uporablja se tudi v različnih praktičnih rešitvah, kot je na primer izdelava in tudi reševanje ugank Sudoku. Žal pa ne poznamo nobenega učinkovitega algoritma za barvanje grafov. Če imamo stotine vozlišč, bi tudi najhitrejši računalniki potrebovali tisoče let za rešitev problema.



V banki so zaposleni trije uslužbenci: Alica, Miha in Klara. Miha dela dvakrat počasneje kot Alica, Klara pa trikrat počasneje kot Alica.

	kratko opravilo	običajno opravilo	dolgo opravilo
Alica	1 minuta	2 minuti	3 minute
Miha	2 minuti	4 minute	6 minut
Klara	3 minute	6 minut	9 minut

V vrsti stojijo:

- 5 strank s kratkimi opravili,
- 2 stranki z običajnimi opravili in
- 6 strank z dolgimi opravili.

Stranke bomo razporedili k uslužbencem tako, da bo delo čimprej končano. Koliko časa bo trajalo?

Rešitev

Potrebujejo petnajst minut.

Alica prevzame 5 strank z daljšimi opravili (petnajst minut).

Miha prevzame 2 stranki s hitrimi opravili, 1 stranko z običajnim opravilom in 1 stransko z daljšim opravilom (štirinajst minut).

Klara prevzame 3 stranke s hitrimi opravili in 1 stranko z običajnim opravilom (petnajst minut).

Kako dokažemo, da je to najhitrejši način?

Vsako stranko z običajnim opravilom lahko predstavimo kot 2 stranki s hitrim opravilom. Vsako stranko z daljšim opravilom predstavimo kot 3 stranke s hitrim opravilom. Zdaj imamo $5+4+18=27$ enako dolgih opravil. Naloga ni enaka zgornji, ampak zagotovo ne more imeti slabše rešitve kot zgornja.

Alica obdela 1 opravilo na minuto, Miha obdela $\frac{1}{2}$ opravila na minuto, Klara obdela $\frac{1}{3}$ opravila na minuto. Skupaj v eni minuti opravijo $1 + \frac{1}{2} + \frac{1}{3} = \frac{11}{6}$ opravil. Za 27 transakcij potrebujejo vsaj $27 : \frac{11}{6} = \frac{162}{11} > 14$ minut, torej potrebujemo vsaj 15 minut, kar pa je enako našemu rezultatu.

Računalniško ozadje

V nalogi je predstavljen problem razvrščanja nalog. Tudi računalniki morajo razvrščati naloge tako, da bi kar se da hitro opravili vse.



Članice dekliškega računalniškega krožka se za vikend odpravljajo na izlet. Prespale bodo v hostlu z velikimi, šestposteljnimi sobami. Dogovoriti pa se morajo, katere punce si bodo delile sobe.

Zato vsaka punca zapiše svoje želje glede deljenja sobe:

- s katerimi puncami želi biti skupaj v sobi (+) ter
- katerih punc nikakor ne želi imeti v sobi (-).

<u>Ana</u> +: <i>Lina</i> -:	<u>Ema</u> +: -: <i>Ana</i>	<u>Lara</u> +: -: <i>Ema</i>	<u>Lina</u> +: -: <i>Lara</i>	<u>Mija</u> +: <i>Ema, Zala</i> -:	<u>Zala</u> +: <i>Mija</i> -: <i>Ana</i>
------------------------------------	-----------------------------------	------------------------------------	-------------------------------------	--	--

Njihova mentorica želi ustreči željam vseh punc. Zato razdeli punce po sobah glede na njihove želje, kar ji tudi uspe. Koliko punc je v sobi, v kateri spi Ema?

Rešitev

Tri.

Če želimo izpolniti vse želje punc, se lahko osredotočimo na pozitivne želje (s kom bi želela biti skupaj v sobi) in dodeljujemo sobe po naslednjem postopku:

1. Če imamo prazno sobo,
 - a. vanjo nastanimo punco;
 - b. sicer jo nastanimo v sobo, kjer ta punca ni na negativnem seznamu (-) nobene punce, ki je že v tej sobi (če take sobe ni, postopek neuspešno zaključimo).
2. Vse punce, ki so na pozitivnem seznamu (+) prve punce v sobi, nastani v isti sobi. Če smo sobe dodelili vsem puncam, končamo.
3. Ponovi korak 1.

Če uporabimo zgornji postopek, dobimo naslednji rezultat:

<u>Ana</u> +: <i>Lina</i> -:	<u>Lina</u> +: -: <i>Lara</i>	
<u>Mija</u> +: <i>Ema, Zala</i> -:	<u>Ema</u> +: -: <i>Ana</i>	<u>Zala</u> +: <i>Mija</i> -: <i>Ana</i>
<u>Lara</u> +: -: <i>Ema</i>		

To je tudi edina razdelitev, ki zadosti vsem željam: Ana mora biti skupaj z Lino, vendar Ana (in torej tudi Lina) ne sme biti v isti sobi niti z Emo niti z Zalo. Posledično tudi ne more biti v isti sobi kot Mija, saj mora biti Mija v sobi z Emo in Zalo. Lara pa mora dobiti svojo sobo, saj si je ne more deliti niti z Lino (in torej tudi ne z Ano) niti z Emo (in torej tudi ne z Mijo in Zalo). Torej je Ema v sobi skupaj le z Mijo in Zalo, tako so v sobi tri punce.

Kot vidimo, smo dobili ustrezen rešitev tega problema. Lahko pa bi se zgodilo tudi, da rešitev ne bi obstajala. Če bi, na primer, Ana želela biti v isti sobi kot Lina, a Lina ne bi želela biti v isti sobi z Ano, ne bi mogli ustreči obema.

Računalniško ozadje

Za rešitev problema, da so vse punce srečne, je mentorica poiskala takšno dodelitev sob, ki ustreza vsem željam. A kako lahko takšno rešitev poiščemo? Najpreprostejši način je, da punce po sobah razporedimo na vse možne načine in preverjamo, ali je zadoščeno vsem njihovim preferencam. Ker pa je takih načinov razporeditve zelo veliko, lahko iskanje rešitve traja zelo dolgo. Bolj učinkovit način je, če najprej upoštevamo preference prve punce in na tem zgradimo razdelitev, ki upošteva podane omejitve.



Robi je vse svoje dragocenosti zaklenil v omaro s ključavnico na kodo. Ker se je bal, da bo pozabil trimestno kodo, si je zapisal naslednje namige:

1 7 2:	Le ena od številke je pravilna, a se nahaja na napačnem mestu.
8 5 4:	Dve številki sta pravilni, a se nahajata na napačnih mestih.
9 8 6:	Le ena od številke je pravilna, a se nahaja na napačnem mestu.
7 5 1:	Le ena številka je pravilna, nahaja pa se tudi na pravem mestu.

Kakšna je koda?

Rešitev

Pravilen odgovor je **7 4 8**.

Glede na prvi in drugi namig ima koda lahko naslednje številke: 1 7 2 8 5 4.

Če upoštevamo še tretji namig, vidimo, da je 8 edina številka, ki se nahaja med možnimi številkami (1 7 2 8 5 4). Številka 8 mora biti na tretjem mestu, saj ne more biti niti na prvem mestu (drugi namig pravi, da je na napačnem mestu) niti na drugem mestu (tretji namig tudi pravi, da je na napačnem mestu). Torej, na tretjem mestu je številka 8.

Upoštevajoč četrti namig ugotovimo, da v kodi ni številke 1 (če bi bila, bi bila na tretjem mestu, tam pa že imamo 8), pa tudi ne številke 5 (če bi bila, bi morala biti na drugem mestu, kar pa se ne ujema z drugim namigom). Torej imamo v kodi številko 7, nahaja pa se na prvem mestu.

Sedaj moramo poiskati le še številko na drugem mestu. Tu upoštevamo drugi namig in ugotovimo, da je manjkajoča številka 4.

Iskana koda je tako **7 4 8**.

Računalniško ozadje

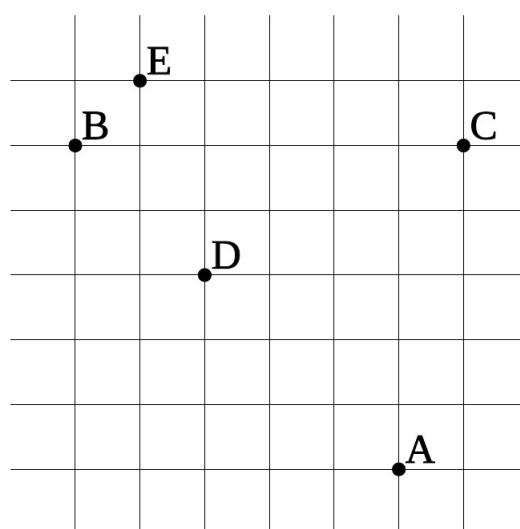
Naloga se navezuje na logiko: podani namigi so logične omejitve, ki jih mora upoštevati vsaka pravilna rešitev.

Vendar pa moramo pri tem poudariti, da je v našem primeru rešitev ena sama, zato to ni najboljši način za skrivanje kode: če poznamo namige, poznamo tudi samo kodo.



Prijatelji Ana, Borut, Ciril, Darko in Ema živijo na lokacijah, ki so na spodnjem zemljevidu označene s črkami A, B, C, D in E. Prijatelji se želijo srečati na enem od križišč, da bi se skupaj učili.

Do kraja srečanja gredo po ulicah in vsak od njih izmeri razdaljo, ki jo je moral prehoditi (to je skupno število vodoravnih in navpičnih blokov od njegovega doma). Tako je na primer razdalja med Anino in Emino hišo 10 blokov.



Križišče, na katerem se bodo srečali, bodo izbrali tako, da bo vsota razdalj, ki jih morajo prehoditi vsi skupaj, čim manjša. Kakšna bo torej vsota razdalj od vseh petih domov do kraja srečanja?

Rešitev

Pravilen odgovor je 18 blokov.

Najprej lahko opazimo, da se razdalja v smeri vzhod – zahod ne spremeni za nobenega od prijateljev, če kraj srečanja premikamo proti severu ali jugu. In podobno se razdalja v smeri sever – jug ne spremeni, če kraj srečanja premikamo proti vzhodu ali zahodu. To pomeni, da lahko rešitev poiščemo ločeno za vsako dimenzijo (vodoravno in navpično).

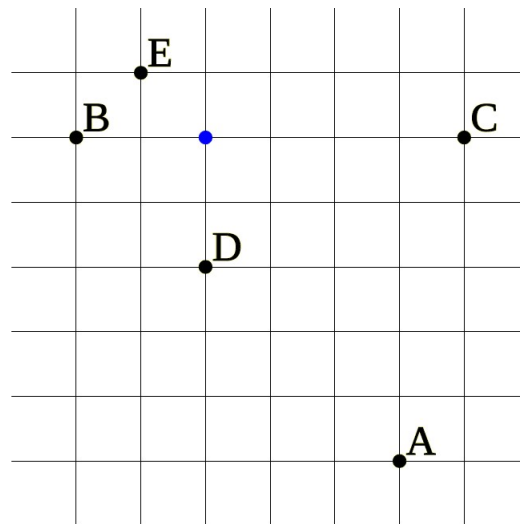
Najprej poiščimo optimalno pozicijo v smeri vzhod – zahod. Celoten zemljevid lahko preslikamo v eno samo linijo, ki jo prikazuje spodnja slika.



Razmislimo, kaj se zgodi, ko predlagan kraj srečanja, predstavljen z modro piko, premikamo proti vzhodu ali zahodu. Če ga premaknemo proti vzhodu, se razdalja od doma do kraja srečanja poveča za tri prijatelje (Boruta, Darka, Emo) in zmanjša za dva od njih (Ano in Cirila), torej se skupna

razdalja poveča. Če jo premaknemo proti zahodu, je situacija obrnjena in se skupna razdalja zmanjša. Zato je najbolje, da točko prestavimo proti zahodu, dokler ne dosežemo točke D. Po tem bi namreč premikanje dalje proti zahodu povečevalo razdaljo trem prijateljem, zmanjševalo pa le dvema. Torej je točka D optimalna rešitev v smeri vzhod – zahod. Taka rešitev je splošna in ni odvisna od dejanskih razdalj med posameznimi točkami: optimalna pozicija je vedno tista na sredini urejenega zaporedja (imenujemo jo mediana).

Podobno razmišljanje uporabimo tudi v smeri sever – jug in ugotovimo, da mora biti kraj srečanja na tisti ulici, kjer živi Borut in Ciril, kot kaže spodnja slika.



Sedaj lahko še preštejemo razdalje od hiše vsakega prijatelja do kraja srečanja: 8 blokov za Ano, 2 za Boruta, 4 za Cirila, 2 za Darka in 2 za Emo, skupaj torej 18 blokov.

Računalniško ozadje

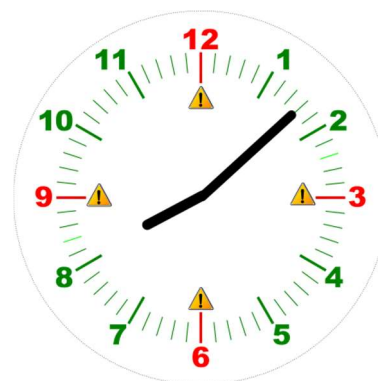
Pri tej nalogi smo uporabili dekompozicijo, da smo problem razdelili na dva podproblema, ki ju lahko rešujemo ločeno. Ta trik je zelo uporaben tako v računalništvu kot tudi drugje.



Branko dela na Centralni železniški postaji. Njegov delovnik se začne ob 8.00. Za vsakih 15 minut zamude mora plačati 10 evrov kazni. Na primer, če pride na delo pred 8.15 (npr. ob 8.11), jo odnese brez plačila, če pa pride ob 8.20, mora plačati za prvih 15 minut zamude.

Danes je Branko prespal jutranjo budilko in prišel na železniško postajo v svojem domačem kraju šele ob 8.08.

Spodnja tabela prikazuje vozne rede različnih vlakov:



Vlak	Vozni red	Čas do Centralne postaje	Cena karte
Redni	od 6.00 dalje vsakih 5 minut	40 minut	5 EUR
Delavski	od 6.00 dalje vsakih 10 minut	30 minut	10 EUR
Hitri	od 7.00 dalje vsakih 15 minut	20 minut	15 EUR
Ekspres	od 7.00 dalje vsakih 20 minut	12 minut	20 EUR

S katerim vlakom naj se pelje, da ga bo jutranja zamuda stala najmanj?

Rešitev

Pravilen odgovor je: delavski vlak.

Delavski vlak je za Branka najcenejša rešitev, kar zlahka preverimo z nekaj računanja:

- Če izbere Redni vlak (ob 8.10), bo prišel na delo šele ob 8.50, kar ga bo stalo 35 EUR (5 EUR za karto in 30 EUR (3*10 EUR) kazni za zamudo).
- Če izbere Delavski vlak (ob 8.10), bo prišel na delo ob 8.40, kar ga bo stalo 30 EUR (10 EUR za karto in 20 EUR (2*10 EUR) kazni za zamudo).
- Če izbere Hitri vlak (ob 8.15), bo prišel na delo ob 8.35, kar ga bo stalo 35 EUR (15 EUR za karto in 20 EUR (2*10 EUR) kazni za zamudo).
- Če izbere Ekspresni vlak (ob 8.20), bo prišel na delo ob 8.32, kar ga bo stalo 40 EUR (20 EUR za karto in 20 EUR (2*10 EUR) kazni za zamudo).

Računalniško ozadje

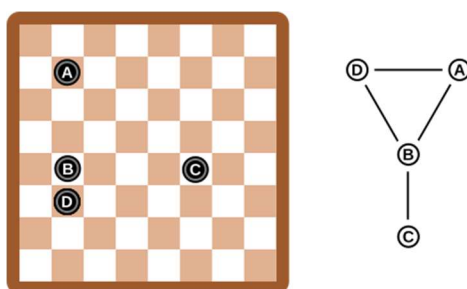
Naloga prikazuje koncept optimizacije. Optimizacija pomeni izbiro najboljšega elementa (v našem primeru najcenejšo možnost) iz množice alternativ, ki so na voljo (v našem primeru različni vlaki), pri tem pa upoštevamo podane kriterije (v našem primeru čas in stroški).

Optimizacijski problem navadno zahteva maksimiziranje ali minimiziranje vrednosti funkcije s sistematičnim izborom vhodnih vrednosti in izračunavanja vrednosti funkcije.



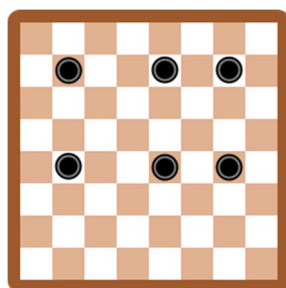
Slika na levi prikazuje šahovnico s štirimi figurami. Glede na pozicijo figur lahko narišemo diagram, ki je prikazan na desni sliki. Pri tem smo sledili naslednjim navodilom:

- Za vsako figuro na šahovnici narišemo krogec.
- Če sta dve figuri na šahovnici v isti vrstici ali v istem stolpcu, potem v diagramu njuna krogeca povežemo s črto.
- V diagramu ne rišemo nobenih drugih črt.



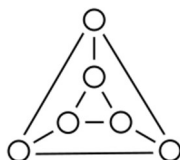
Na zgornji sliki smo figure in pripadajoče krogece označili s črkami, tako da lažje preverite pravilnost izrisane diagrama.

Na enak način smo narisali tudi diagram za šahovnico s šestimi figurami, ki jo prikazuje slika:

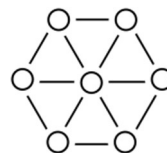


Kategorega od naslednjih diagramov smo dobili?

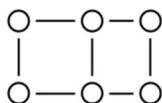
A.



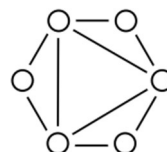
C.



B.



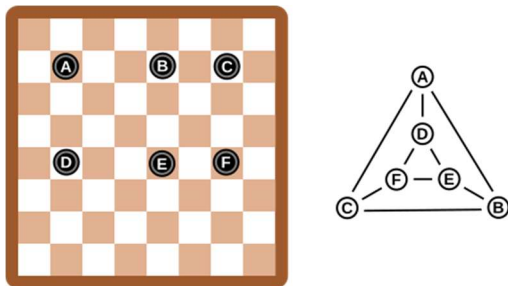
D.



Rešitev

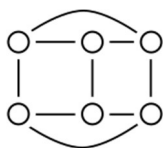
Pravilen odgovor je A.

Odgovor lahko enostavno preverimo na spodnji sliki, kjer so figure označene s številkami.



Da ostale tri rešitve niso pravilne, lahko najenostavneje preverimo tako, da opazimo, da ima vsaka figura na šahovnici še dve drugi figuri v isti vrstici in eno drugo figuro v istem stolpcu. Torej mora biti na diagramu vsak krogec povezan z $2 + 1 = 3$ ostalimi krogi. (In mimogrede, odgovor C ima 7 krogcev, torej enega preveč.)

Bi lahko bila rešitev B tudi pravilna? Nenazadnje izgleda zelo podobna postavitvi figur na šahovnici. Vendar pa imata skrajno leva in skrajno desna krogca le dve povezavi z drugimi krogi, torej diagram ne more biti pravilen. Če pa dodamo še dve povezavi, kot je prikazano na spodnji sliki, dobimo pravilen diagram.



Računalniško ozadje

V računalništvu se podobni diagrami uporabljajo za predstavitev pomembnih informacij podanega problema. Tak diagram imenujemo *graf*. Krogi v diagramu se imenujejo *vozlišča* grafa.

V grafu je pomembno le, ali sta dve vozlišči povezani ali ne. Kje je vozlišče narisano, pa je vseeno. Tako lahko isti graf narišemo na več načinov, kot smo to naredili tudi v rešitvi naše naloge: tako odgovor A kot zadnja slika v razlagi sta pravilna diagrama podane šahovnice ter predstavljata isti graf.



Danijel je od dedka dobil skrinjo, ki je zaklenjena z elektronsko ključavnico. Ključavnico lahko odpre z vpisom kode, ki jo sestavlja 9 števk. Dobil je tudi naslednje namige:

- Koda ima le števke 2, 6, 7 in 9.
- Števka z največjo vrednostjo je v kodi uporabljena najmanjkrat.
- Števka z najmanjšo vrednostjo je v kodi uporabljena največkrat.
- Koda je enaka, tudi če jo beremo od zadaj naprej.
- Vse sosednje števke so različne.
- Zadnja števka je liha.

Kakšna je koda ključavnice?

Rešitev

Pravilen odgovor je 726292627.

7	2	6	2	9	2	6	2	7
---	---	---	---	---	---	---	---	---

Prvi, drugi in tretji namig nam povedo, da se števka 9 pojavi enkrat, števki 6 in 7 dvakrat ter števka 2 na preostalih 4 mestih.

Četrty namig pove, da mora biti števka 9 v sredini.

Šesti namig (skupaj s četrty namigom) določa, da je števka 7 na začetku in na koncu.

Peti namig pove, da je števka 2 na mestih 2, 4, 6 in 8.

Torej ostane le še števka 6, ki jo postavimo na preostali dve mesti.

Računalniško ozadje

Do rešitve bi lahko prišli tudi tako, da bi preverili vse možne kombinacije devetih števk, ali sledijo vsem navedenim namigom. Vendar pa ob upoštevanju namigov dobimo dovolj informacij, da lahko sestavimo rešitev, čeprav na prvi pogled to ni povsem očitno. Pri tem smo uporabili *sklepanje na podlagi znanja*: o problemu imamo neko znanje in na podlagi tega lahko z logičnim sklepanjem izpeljemo nova dejstva in rešimo problem.



Zamislili smo si naslednjo igro. Na mizo položimo vrsto kart. Vsaka karta je lahko obrnjena tako, da vidimo njen hrbet  ali pa njeno lice .

V vsakem koraku igre naredimo naslednje:

- preverjamo karte *od desne proti levi*,
- če karta kaže *hrbet*, jo obrnemo (tako da kaže lice) in končamo,
- če karta kaže *lice*, jo obrnemo (tako da kaže hrbet) in nadaljujemo z naslednjo karto,
- ko pridemo do konca kart, končamo.

Na spodnji sliki lahko vidimo, kako se vrsta kart spremeni po enem koraku: najprej obrnemo skrajno desno karto (tretje pravilo), nato karto levo od nje (tretje pravilo) ter še eno karto na levi. Na tem mestu končamo, saj smo obrnili karto, ki je kazala hrbet (drugo pravilo).

Pred korakom:



Po izvedenem koraku igre:



Igro začnemo z naslednjo vrstico kart:



Koliko korakov igre moramo narediti, da pridemo do spodnjega vzorca (vse karte kažejo lice)?



- A. Potrebujemo 10 korakov ali manj.
- B. Potrebujemo več kot 10 korakov, a manj kot 100.
- C. Potrebujemo več kot 100 korakov, a manj kot 1000.
- D. Potrebujemo več kot 1000 korakov.
- E. Rešitve, kjer vseh sedem kart kaže lice, ne moremo nikoli dobiti.

Rešitev

Pravilen odgovor je B.

Da odkrijemo, kaj se dogaja, začnimo z vrstico kart, obrnjenih s hrbtom navzgor, in opazujemo prve korake.

Začetek	
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	

Vidimo lahko, da skrajno desno karto prikažemo z licem navzgor po prvem koraku, za drugo karto z desne potrebujemo 2 korake, za tretjo z desne 4 korake, za četrto z desne pa 8 korakov.

Če pogledamo to zaporedje, 1, 2, 4, 8 ..., ugotovimo, da je vsako naslednje število dvakrat večje od predhodnega. Torej lahko uganemo, da bomo za prikaz pete karte z desne potrebovali 16 korakov, za prikaz šeste 32 ter za prikaz sedme karte (tj. prve karte z leve) 64 korakov. Za prikaz osme, če bi obstajala, bi potrebovali 128 korakov.

Poleg tega vidimo, da se nova karta na levi obrne takrat, ko so obrnjene vse karte na desni. Prve tri karte (z desne) bodo obrnjene tik preden obrnemo četrto, torej po 15 korakih.

Vseh sedem kart bo obrnjenih, tik preden bi obrnili osmo, torej po 127 korakih. A to velja le, če bi začeli tako, da vse karte kažejo hrbte. Ker je druga karta (z leve) že obrnjena, pa smo jih naredili že vsaj 32. Torej bo potrebnih manj kot 100 korakov.

Računalniško ozadje

Bi znali izračunati točno število korakov?

Številke so v računalniku predstavljene v dvojiškem (binarnem) zapisu, pri katerem namesto števk 0 do 9 uporabimo le dve vrednosti: 0 in 1 (imenujemo jih tudi *biti*). Če za predstavitev števila uporabimo 7 bitov (sedem ničel ali enk), lahko prvih nekaj števil zapišemo takole: 1 je predstavljena kot 0000001, 2 kot 0000010, 3 kot 0000011, 4 kot 0000100, 5 kot 0000101, 6 kot 0000110 in tako naprej.

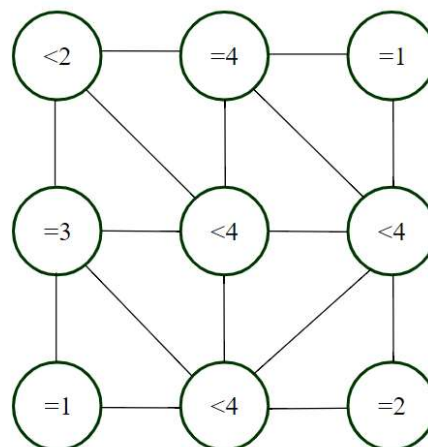
Karte v naši nalogi bi lahko zapisali kot 0 (vidimo hrbet karte) ali 1 (vidimo lice karte). Pravila, ki jih upoštevamo pri obračanju kart v enem koraku, pa številu prištejejo 1 (po koraku vrstica kart predstavlja število, ki je za 1 večje od števila, ki ga je predstavljala vrstica kart pred korakom). Tako lahko tudi hitreje rešimo nalogo: na začetku imamo število 42 ($2^5 + 2^3 + 2^1 = 32 + 8 + 2 = 42$) in po izvedenih korakih moramo priti do števila 127 ($2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 64 + 32 + 16 + 8 + 4 + 2 + 1 = 127$). Koliko je teh korakov? Seveda, $127 - 42 = 85$.

V našem primeru smo za predstavitev števil uporabili 7 bitov, v sodobnih računalnikih pa se navadno uporablja 32 ali 64 bitov.



Imamo 9 krogov, ki so povezani z nekaterimi sosednjimi krogi, skupaj imamo 16 povezav. Kroge na sliki želimo pobarvati tako, da bodo veljala pravila, ki so zapisana v krogih – številka v krogu pove, koliko krogov, ki so povezanih s tem krogom, je pobarvanih. Tako mora na primer krog, ki je označen z »= 3«, imeti pobarvane natanko tri od svojih štirih povezanih sosedov. Podobno mora imeti krog, označen z »< 4«, manj kot 4 pobarvane sosedov.

Koliko krogov moramo pobarvati?



Rešitev

Pravilen odgovor je 5 krogov.

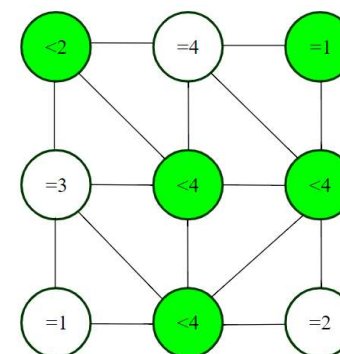
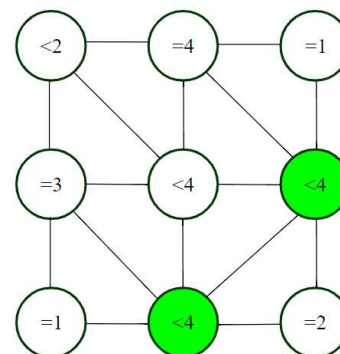
Če pogledamo spodnji desni krog, vidimo, da moramo pobarvati oba njegova soseda. Tako dobimo sliko na desni.

Prav tako morajo biti pobarvani vsi štirje sosedi kroga v sredini zgoraj (= 4). Tako dobimo spodnjo sliko.

S tem smo zadostili pogojem v vseh krogih. Vidimo lahko tudi, da ne smemo pobarvati nobenega od preostalih krogov.

- Če pobarvamo levi spodnji krog z »= 1«, ima krog »= 3« napačno oznako.
- Če pobarvamo krog z »= 2«, to ni skladno z oznako kroga nad njim.
- Če pobarvamo krog z »= 3«, to ni skladno z oznako kroga nad njim.
- Če pobarvamo krog z »= 4«, to ni skladno z oznako kroga na desni.

Enak rezultat bi dobili, če bi barvanje začeli s sosedi kakšnega drugega kroga.



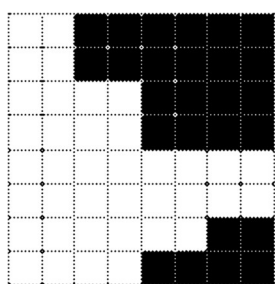
Računalniško ozadje

V tem problemu smo morali uporabiti logično sklepanje, saj rešitev s preizkušanjem vseh možnosti ni učinkovita. Ker imamo 9 krogov, vsak pa je lahko pobarvan ali ne (2 možnosti za vsak krog), je skupno število možnosti za barvanje teh krogov $2^9 = 512$. Če bi za vsako od 512 možnosti preverjali, ali ustrezajo zapisanim pogojem, bi bila pot do rešitve zelo dolgotrajna. Zato smo se je raje lotili tako, da smo s sklepanjem zmanjšali število možnih rešitev na bolj obvladljivo.

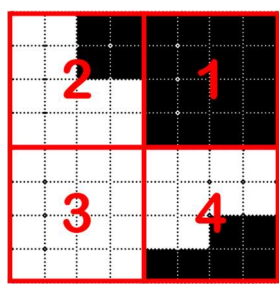
Pirat Enooki ima zelo velik zemljevid, ki prikazuje pot do skritega zaklada. Zemljevid je razrezan na več manjših kosov, vsak kos pa prikazuje območje velikosti 8 x 8 enot (en kos zemljevida je prikazan na sliki 1).

Enooki se je s čolnom odpravil na lov za zakladom, a ker je njegov čoln zelo majhen, ne more s seboj vzeti celega zemljevida. Ker pa je Enooki zelo iznajdljiv, se je domislil načina za zapis vsakega območja (vsakega kosa zemljevida) v obliki diagrama ter ga zapisal v svojo beležko. Pri tem je uporabil naslednji postopek:

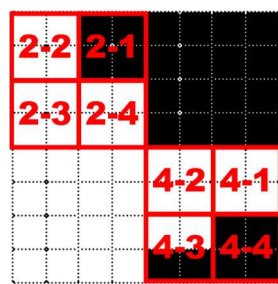
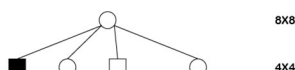
1. Če so vsi deli tega območja enake barve, v beležki to označi kot kvadratega te barve.
2. Sicer pa v beležki označi krogec (glej diagram 1) ter območje razdeli na štiri manjša podobmočja (glej sliko 2).
3. Koraka 1 in 2 ponavljaš, dokler v beležki nimaš zapisanih vseh kosov zemljevida (kar prikazuje diagram 4).



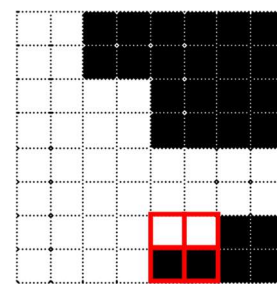
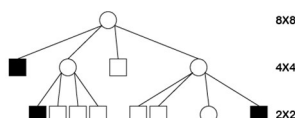
Slika 1



Slika 2



Slika 3



Slika 4

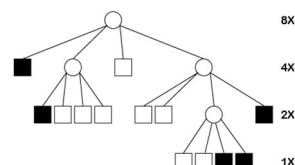


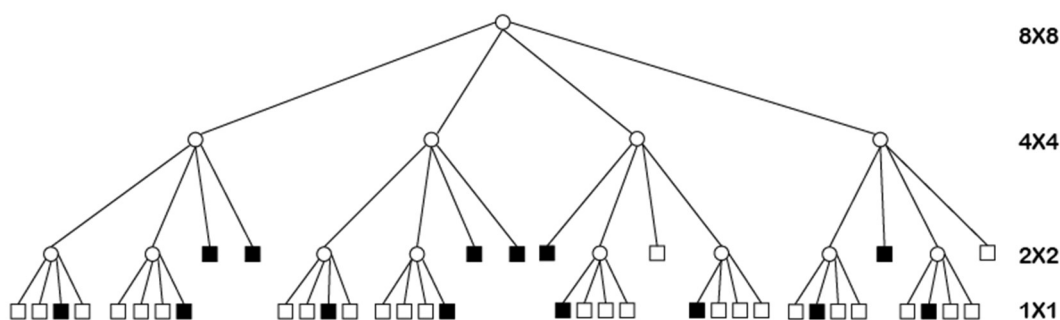
Diagram 1

Diagram 2

Diagram 3

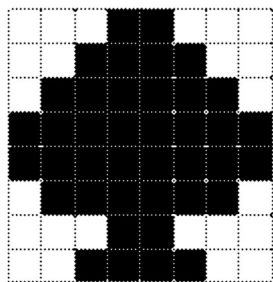
Diagram 4

Enooki je v beležko zapisal naslednji diagram:

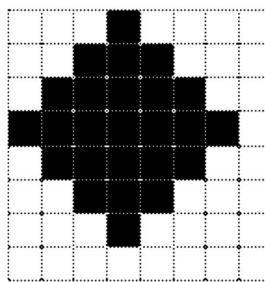


Kateri del zemljevida predstavlja ta diagram?

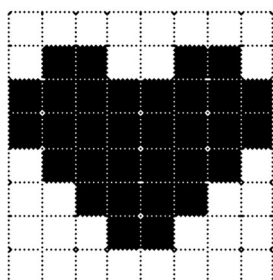
A.



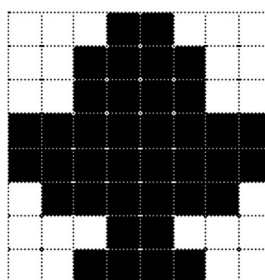
C.



B.



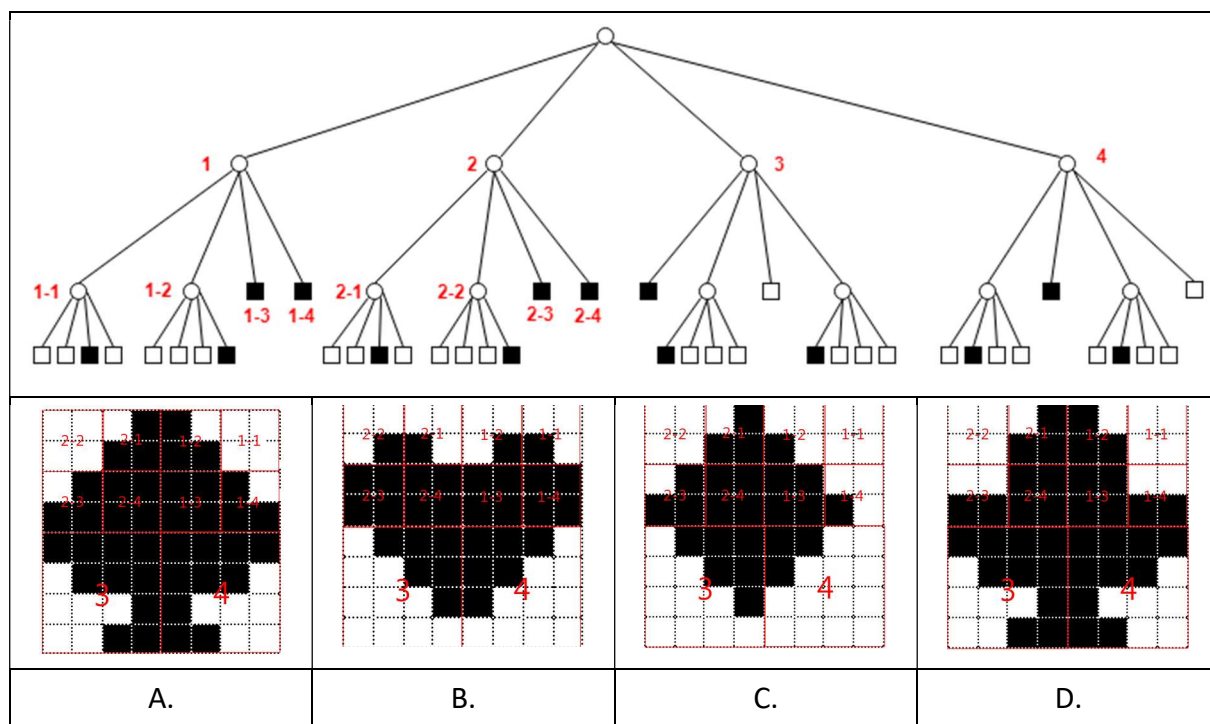
D.



Rešitev

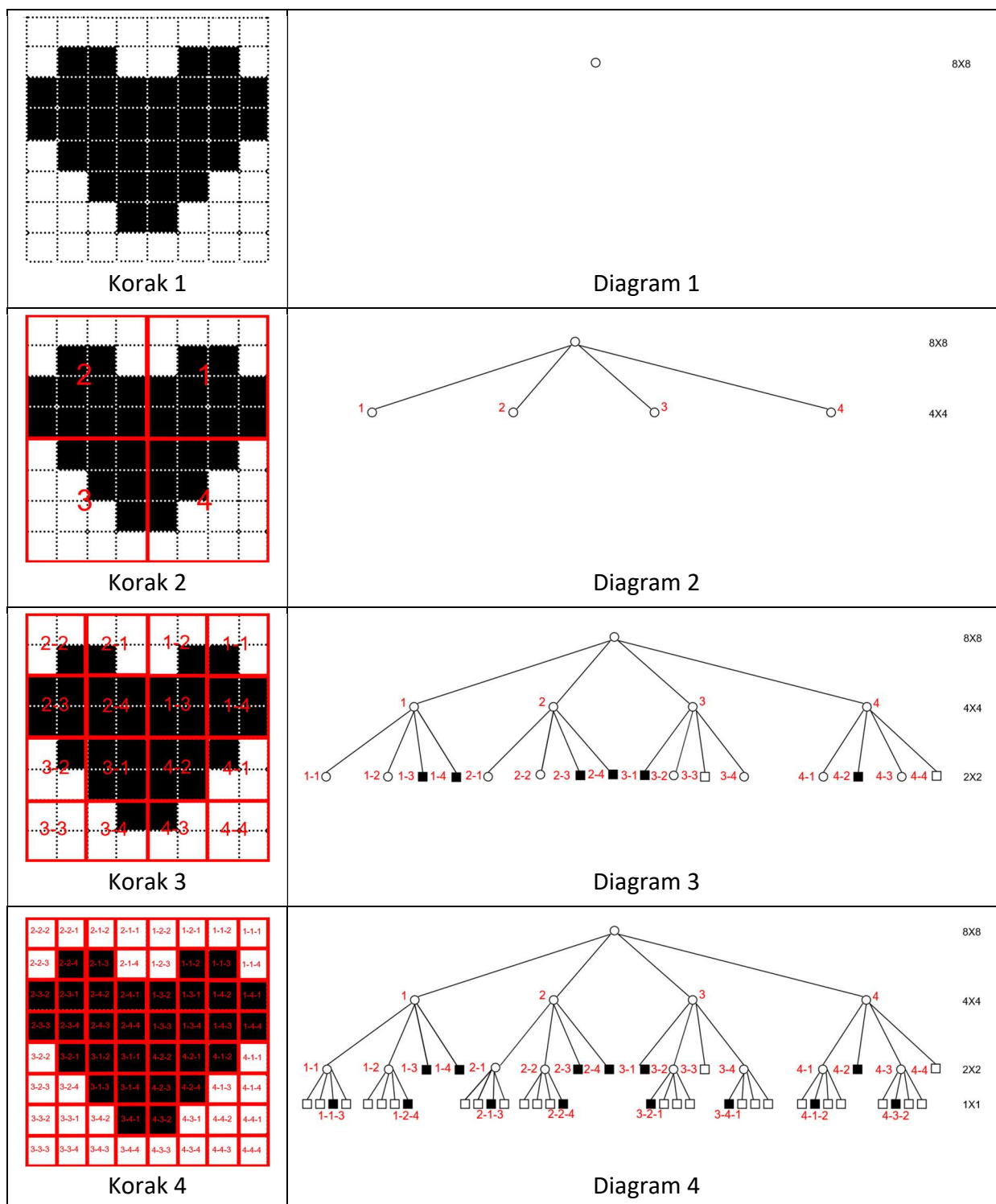
Pravilen odgovor je B.

V diagramu lahko označimo točke, ki predstavljajo območja velikosti 4 x 4, z oznakami 1 do 4, nato podtočke točke 1 z oznakami 1-1 do 1-4 in tako naprej. Ustrezne pozicije območij na zemljevidu in točk v diagramu prikazuje naslednja slika:



Točke 1-3, 1-4, 2-3 in 2-4 so v diagramu označene s črnim kvadratom, kar pomeni, da so vsa ta območja velikosti 2 x 2 na zemljevidu črna. Temu pogoju zadostuje le del zemljevida pod B.

Kos zemljevida pod B zapišemo v obliki diagrama na naslednji način:



Računalniško ozadje

V računalništvu take diagrame imenujemo *štiri-drevo* (angleško *Quadtree*). Štiri-drevo je drevesna podatkovna struktura, v kateri iz vsakega vozlišča vodijo natančno štiri veje. Uporabljamo jih za predstavitve slik (kot v naši nalogi), saj lahko na ta način slike učinkoviteje shranimo, posebej, če vsebuje večje površine enakih barv. Na podoben način deluje tudi stiskanje video posnetkov.



Diagnostična naprava v laboratoriju mora neprestano stresati vzorce krvi, ki jo vzamejo pacientom. Naprava deluje po navodilih računalniškega programa, ki je zapisan v oštevilčene vrstice. Naprava bere program vrstico za vrstico. Ko prebere eno vrstico, takoj izvrši v njej zapisan ukaz. Če vrstica vsebuje ukaz `pojdi na X`, naprava skoči na vrstico X ter nadaljuje branje vrstic in izvajanje ukazov.

Ukazi programa lahko shranijo število v A, prištejejo 1 številu, ki je shranjeno v A, ter primerjajo vrednost A z drugim številom.

Diagnostična naprava ima zapisan naslednji program:

```
1 nastavi A na 0
2 prištej 1 k A
3 pojdi na 6
4 če A enako 60, pojdi na 8
5 nastavi A na 0
6 prištej 1 k A
7 pojdi na 2
8 ponovi A krat: pretresi vzorce
9 konec
```

Kolikokrat bo naprava pretresla vzorce, če sledi zgornjemu programu?

- A. Vzorci ne bodo nikoli pretreseni.
- B. Vzorci bodo pretreseni enkrat.
- C. Vzorci bodo pretreseni 60-krat.
- D. Diagnostični postopek ne bo nikoli prenehal s pretresanjem vzorcev.

Rešitev

Pravilen odgovor je A.

Program neprestano preskakuje iz vrstice 3 v 6 ter iz vrstice 7 v 2. Razen na začetku, program tako izvaja le ukaze v vrsticah 2, 3, 6 in 7. Ukaz za pretresanje vzorcev je v vrstici 8, do katere program nikoli ne pride. To pomeni, da naprava nikoli ne pretrese ničesar. Pravilen odgovor je torej A. Poleg tega se tudi ukaz v vrstici 9 nikoli ne izvede, kar pomeni, da se program nikoli ne konča.

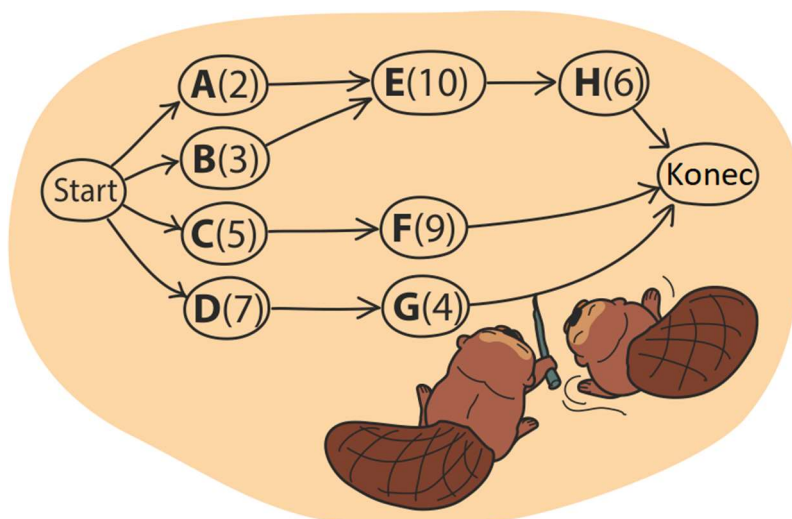
Računalniško ozadje

Prvi računalniški jeziki, ki so jih razvili v štiridesetih in petdesetih letih prejšnjega stoletja, so bili podobni našemu programu v nalogi. Imenujejo se *zbirni jeziki*. Vrstice programa so oštevilčene in s

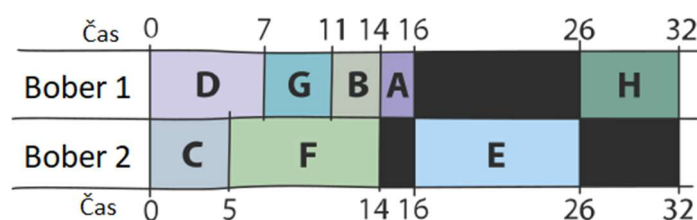
posebnim ukazom `pojdi na` (angleško `go to`) program preskoči na poljubno vrstico. Take programe, ki preskakujejo naprej in nazaj po vrsticah, je zelo težko brati in še težje v njih poiskati napako. Po drugi strani pa napake v njih zlahka naredimo. Ravno pogoste napake in njihovo težavno odstranjevanje so bili tudi glavni razlogi, da so v petdesetih letih začeli razvijati modernejšje programske jezike. Slednji niso orientirani po vrsticah in namesto stavkov tipa `pojdi na` vključujejo strukture, kot so zanke, procedure in izbire.



Dva bobra gradita jez. Za izgradnjo jezu morata opraviti osem nalog (podreti drevesa, odstraniti veje, naplaviti debla, zbrati debla in tako naprej), ki smo jih označili s črkami: A(2), B(3), C(5), D(7), E(10), F(9), G(4) in H(6). Številke v oklepajih povedo, koliko ur potrebuje bober, da opravi to nalogo. Nekaterе naloge morajo biti opravljene, preden lahko začnemo z drugo nalogo, kar je označeno s puščicami. Bobra delata sočasno, vsak svojo nalogo.



Bobra uporabita naslednji načrt dela: med vsemi nalogami, ki so pripravljene v določenem trenutku, izbereta največjo (ki traja najdlje). Tako lahko njuno delo zapišemo z naslednjim urnikom:



Kot vidimo na sliki urnika, bobra zaključita delo v 32 urah. Vendar pa bobra nista izbrala najboljšega načrta, saj bi lahko jez dokončala v krajšem času.

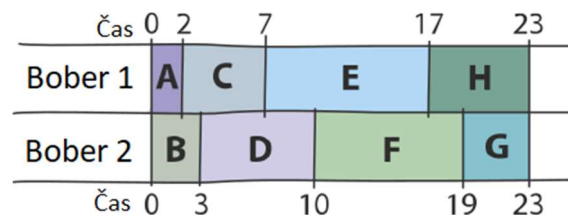
Kakšen je najkrajši čas, ki ga potrebujeta bobra za dokončanje jezu?

Rešitev

Pravilen odgovor je 23 ur.

Slika v nalogi prikazuje urnik dela obeh bobrov. V njem lahko opazimo, da prvi bober ne dela 10 ur, drugi bober pa 8 ur. Seveda bi naloge opravila hitreje, če bi delala cel čas.

Naša strategija iskanja rešitve je, da zagotovimo, da dve najdaljši nalogi, E(10) in F(9) ne opravi isti bober. Eden od možnih urnikov, ki to upošteva, je naslednji:



Vidimo, da bi s tem urnikom lahko jez zgradila v 23 urah.

Hkrati pa je to tudi najkrajši čas, ki je potreben za izgradnjo jezu, saj sta oba bobra zaposlena ves čas.

Računalniško ozadje

Za nekatere primere problema bi strategija obeh bobrov (»izberi največjo preostalo nalogo«) vodila v najkrajši čas izvajanja nalog. A za nekatere druge primere tega problema (kot je tudi ta v nalogi) se bolje obnese strategija delitve največjih nalog. Vendar pa lahko pri obeh pristopih najdemo primere, kjer izbrana strategija ne bo delovala tako dobro. To pa zato, ker lahko zagotovo najdemo razporeditev nalog, ki se zaključijo v najkrajšem času le tako, da preizkusimo vse možne kombinacije veljavnih razporeditev (razen v primeru, če imamo pri problemu še kakšne dodatne omejitve). To pa je v realnih situacijah seveda nepraktično, saj lahko za iskanje najboljše razporeditve nalog porabimo več časa, kot ga potrebuje en sam bober za izgradnjo jezu!



Samovozeči avto se ni vrnil domov, ampak je ostal nekje v mestu. Tik preden se mu je popolnoma izpraznila baterija, je parkiral na parkirno mesto in domov poslal pozicijo nekaterih objektov (a ne vseh) v njegovi okolici, ki jih je prepoznal senzor. Vsak objekt določata dve vrednosti:

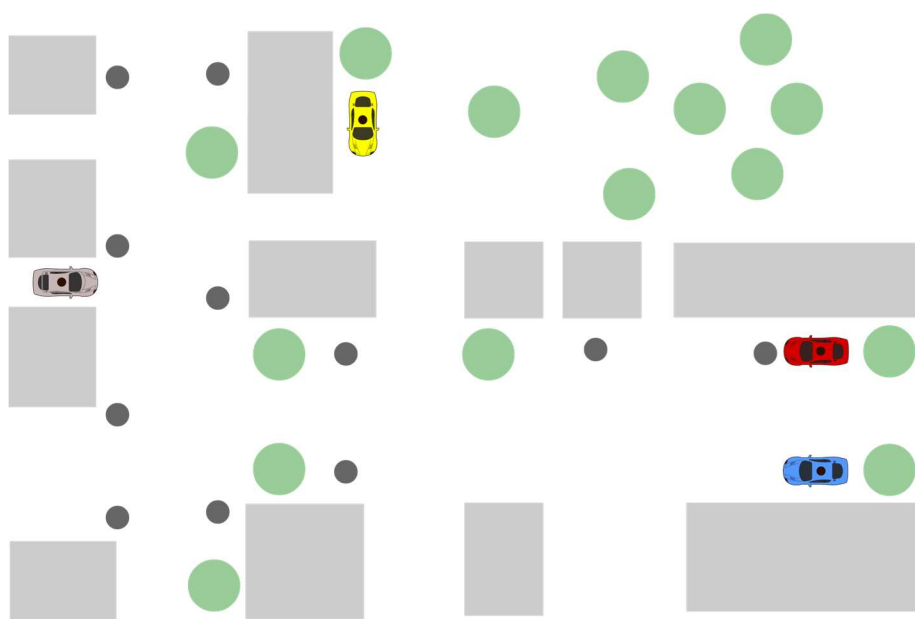
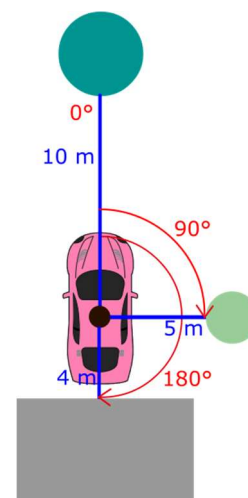
1. kot (glede na 360-stopinjski senzor na strehi avta; kot 0° je pred avtom, kot prikazuje desna slika) in
2. razdalja med senzorjem na avtu in objektom.

Za primer na sliki bi bile vrednosti naslednje: $[(0, 10), (90, 5), (180, 4)]$

Izgubljeni avto je poslal naslednjo predstavitev njegove okolice:

$$[(0, 5), (90, 4), (180, 5), (270, 12)]$$

Kateri avto na spodnji sliki je naš izgubljeni avto?



Rešitev

Pravilen odgovor je rdeči avto (na sredi slike na desni strani).

Izgubljeni avto ima pred seboj objekt na razdalji 5 metrov, na desni strani na razdalji 4 metre, zadaj na razdalji 5 metrov ter na levi strani na razdalji 12 metrov.

Sivi avto (levo) ne more biti naš izgubljeni avto, saj sta pri njem objekta na levi in desni strani enako oddaljena.

Tudi rumeni avto (zgoraj) ne more biti ta, ki ga iščemo, saj je objekt za njim (zelen krog) veliko bližje kot objekt pred njim (siv pravokotnik).

Modri avto (desno spodaj) ni izgubljeni avto, saj je objekt za njim (zelen krog) veliko bližje kot objekt pred njim (majhen črn krog).

Izgubili smo rdeči avto (desno, na sredini): objekta pred njim in za njim sta enako oddaljena, objekt na desni (siv pravokotnik) pa je veliko bližje kot objekt na levi strani (modri avto).

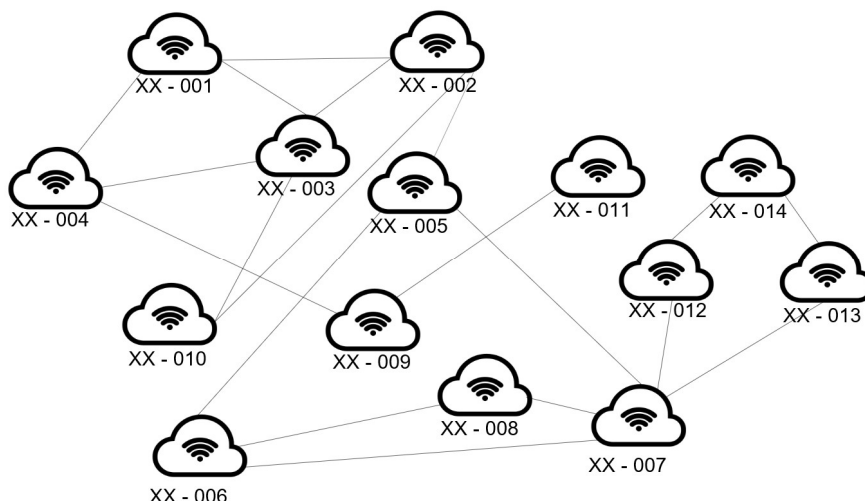
Računalniško ozadje

Samovozeči avtomobili uporabljajo lasersko tehnologijo LIDAR (kratica izvira iz angleških besed *light detection and ranging*) za skeniranje svoje okolice. Njihova programska oprema za navigacijo sestavi kompleksen trodimenzionalen model vseh objektov na razdalji do nekaj sto metrov.

V naši nalogi je bil model okolice avtomobila zelo poenostavljen in je zajemal le bližnje objekte. V splošnem je model abstrakcija realnega sveta in vključuje le tiste vidike realne okolice, ki so pomembni za določen namen. V primeru samovozečih avtomobilov se model uporablja za izogibanje oviram (in preprečevanju trkov). Zato moramo poznati relativne lokacije objektov v okolici avtomobila, medtem ko lahko druge značilnosti, kot je barva objekta, zanemarimo, ker niso pomembne pri izogibanju oviram.



Na sliki je brezžično omrežje manjšega podjetja, ki vključuje 14 intranetnih dostopnih točk.

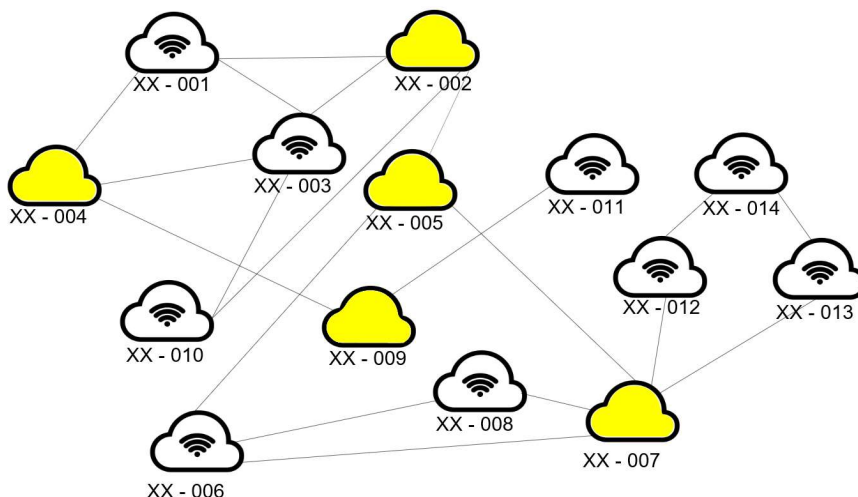


Nekatere dostopne točke v tem omrežju so ključne, kar pomeni, da v primeru njihove odpovedi (nedelovanja) tudi nekatere druge dostopne točke izgubijo povezavo z intranetom. Primer točke je XX-009. Če točka XX-009 ne deluje, potem tudi točka XX-011 nima dostopa do intraneta.

Katere dostopne točke na zgornji sliki so ključne?

Rešitev

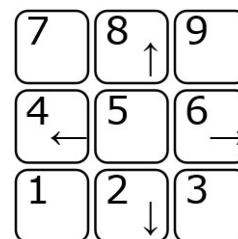
Pravilen odgovor je XX-002, XX-004, XX-005, XX-007, XX-009.



Računalniško ozadje

Omrežja pogosto predstavimo z uporabo grafa kot podatkovne strukture, ki je enostavno množica povezav. Naša naloga je bila v povezanem grafu poiskati vozlišča, ki delijo graf na dva (ali več) ločena dela. Taka vozlišča v angleščini imenujemo *articulation point*.

Za vnos številke v tabelo uporabljamo posebno numerično tipkovnico, s katero lahko vpisujemo številke ali pa premikamo pisalo, ki zapisuje te številke. Tako dvojno delovanje dosežemo tako, da tipkovnico preklapljamo med dvema različnima načinoma delovanja: vnosni in premikalni način.



V vnosnem načinu vsaka tipka zapiše (ali pa prepíše) ustrezno številko v trenutno celico tabele, nato pa se pisalo premakne eno celico na desno.

V premikalnem načinu tipke s puščicami premaknejo pisalo za eno celico v smeri puščice. Tipke brez puščic pa v tem načinu delovanja ne naredijo ničesar.

Tine je začel s prazno tabelo in je po vrsti pritiskal tipke z oznakami 1, 2, 3, 4, 5, 6, 7, 8 in 9. Pri tem je nekajkrat tudi zamenjaval način delovanja tipkovnice. Na koncu je dobil tabelo na desni.

1			9	
	6	7		

Katere tipke je pritisnil v vnašalnem načinu?

Rešitev

Pravilen odgovor je 1, 3, 6, 7 in 9.

Označimo relevantne celice tabele s črkami.

A	B		F	
	C	D	E	

Številka 1 v celici A pomeni, da smo začeli v vnašalnem načinu in nato pisalo premaknili v celico B. Tipka 2 v premikalnem načinu je premaknila pisalo v celico C. Tipka 3 je v celico C vpisala številko 3 in pisalo premaknila v celico D. Tipka 4 v premikalnem načinu je pisalo vrnila v celico C. Tipka 5 v premikalnem načinu ni naredila nič. Tipka 6 v vnašalnem načinu je zamenjala že vpisano 3 s številko 6 ter premaknila pisalo v D. Tipka 7 je vpisala številko 7 v celico D in premaknila pisalo v celico E. Tipka 8 v premikalnem načinu premakne pisalo v celico F. Tipka 9 v vnašalnem načinu pa vpiše število 9 v celico F.

Računalniško ozadje

Takšno »kravžljanje« možganov, kjer moramo na nek nenavaden način priti do nekega rezultata, so tipična vaja za hekerje.



Na plošči imamo 8 kvadratkov, ki so zaporedoma označeni s številkami od 1 do 8. V vsakem kvadratu je zapisan en od treh možnih premikov.

1. Premik v levo: npr. 2L pomeni dva kvadrata v levo:
2. Premik v desno: npr. 3D pomeni tri kvadratke v desno:
3. Brez premika: če je zapisano »0«, se s tega kvadrata ne premaknemo.

V kvadratih imamo zapisana naslednja pravila:

1D	3D	2L	0	3D	1D	3L	2L
1	2	3	4	5	6	7	8

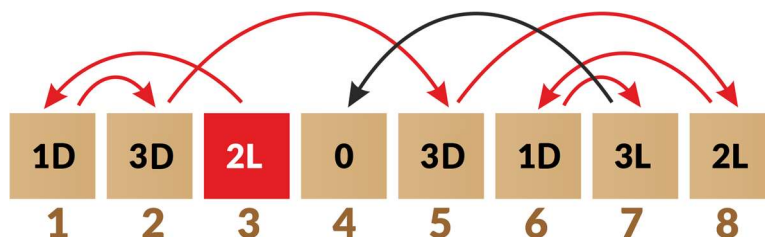
Na katerem kvadratu moramo začeti, da obiščemo vse kvadratke na plošči?

Rešitev

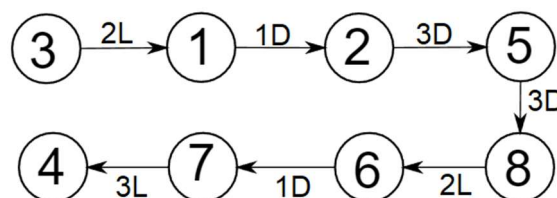
Pravilen odgovor je 3.

Rešitev lahko poiščemo s sledenjem skokom v obratni smeri, torej z začetkom v 0. Na tem kvadratu se

skakanje namreč zaključi. Četrty kvadrata (z zapisano 0) lahko dosežemo s sedmega kvadrata (je tri levo), tega pa lahko dosežemo s šestega kvadrata (je en desno), do katerega pridemo z osmega kvadrata (dva levo). Osmi kvadrata lahko dosežemo s petega kvadrata (tri desno), tega pa z drugega (tri desno) in slednjega s prvega kvadrata (en desno). Prvi kvadrata lahko dosežemo s tretjega (dva levo), ki je tudi naš začetni kvadrata.



Rešitev lahko poiščemo tudi tako, da najprej narišemo situacijo na plošči s pomočjo grafa, kjer so vozlišča kvadratki, povezave pa premiki med njimi. Graf lahko začnemo risati s katerimkoli kvadratom in je končan, ko narišemo vse kvadratke.



Računalniško ozadje

Naloga lepo kaže, kako lahko rešitev kakega zoprnega problema hitro vidimo, če problem predstavimo z grafom. Računalnikarji imamo radi matematike.



Ula, Zala in Žan redno igrajo loterijo, v kateri je lahko največ en zmagovalec. Nikoli ne povedo nobenemu (tudi ne drug drugemu), ali so dobili na loteriji ali ne. Namesto tega naredijo naslednje:

1. Ula in Zala na skrivaj vržeta kovanec.
2. Ula in Žan na skrivaj vržeta kovanec.
3. Zala in Žan na skrivaj vržeta kovanec.
4. Vsak od njih naznani, ali sta bila oba meta kovanca, katerima je prisostvoval, **enaka** ali **različna**. Pri tem tisti, ki ni zadel na loteriji, govori resnico. Tisti, ki je zadel na loteriji, pa se zlaže.

Tako bi v primeru, da so kovanke metali, kot prikazuje spodnja slika, in da je Ula zadela na loteriji, vsak od njih rekel, da so bili kovanci **različni**.



Če Žan reče **enaka**, Zala reče **enaka** in Ula reče **različna**, katera od naslednjih trditev je resnična?

- A. Lahko smo prepričani, da nihče ni dobil na loteriji.
- B. Lahko smo prepričani, da je nekdo dobil na loteriji, vendar ne vemo, kdo.
- C. Lahko smo prepričani, da je nekdo dobil na loteriji, in tudi vemo, kdo.
- D. Ne moremo vedeti, ali je kdo zadel na loteriji.

Rešitev

Pravilen odgovor je B: sicer ne vemo kdo, vendar je eden izmed njih, Ula, Zala ali Žan, zadel na loteriji.

Do rešitve lahko pridemo na različne načine. Tu smo jih opisali le nekaj.

Lahko opazimo, da če oseba laže, to pomeni, da je zadela na loteriji. Dobitek na loteriji torej naredi lažnivca. Sedaj lahko ugotovimo, da je pri odgovorih »enaka, enaka, različna« nekdo gotovo lagal. Do te ugotovitve lahko pridemo na dva načina:

1. Poiščemo protislovje. Recimo, da so vsi trije odgovori resnični. Ker Žan pravi enaka, potem sta meta ŽŽ (Žan – Zala) in ŽU (Žan – Ula) enaka. Ker tudi Zala reče enaka, potem sta tudi meta ŽŽ in ZU enaka. Potemtakem so vsi trije meti enaki. Vendar pa se to ne ujema s predpostavko, da so vsi trije govorili resnico. Torej je eden lagal in imamo dobitnika

loterije. To pa ne pomeni nujno, da je dobitnica loterije Ula; pomeni le, da ni res, da so vsi trije odgovori resnični.

2. Sestavimo tabelo. Lahko tudi sestavimo tabelo z vsemi možnimi izidi vseh treh metov (takih izidov je 8). V tabeli v prvih treh stolpcih zapišemo rezultate posameznih metov, v zadnje tri stolpce pa odgovore Ule, Zale in Žana, če bi vsi trije govorili resnico.

Meti			Resnične izjave		
UZ	UŽ	ZŽ	U	Z	Ž
cifra	cifra	cifra	enaka	enaka	enaka
cifra	cifra	grb	enaka	različna	različna
cifra	grb	cifra	različna	enaka	različna
cifra	grb	grb	različna	različna	enaka
grb	cifra	cifra	različna	različna	enaka
grb	cifra	grb	različna	enaka	različna
grb	grb	cifra	enaka	različna	različna
grb	grb	grb	enaka	enaka	enaka

V tabeli vidimo, da v nobeni vrstici nimamo zapisano »enaka, enaka, različna«, kar pomeni, da je nekdo lagal. Torej odgovora A in D nista pravilna.

Vemo, da je največ ena oseba lagala. Pa vemo tudi, kdo?

- Če je lagal Žan, bi bile resnične izjave: različna, enaka, različna.
- Če je lagala Zala, bi bile resnične izjave: enaka, različna, različna.
- Če je lagala Ula, bi bile resnične izjave: enaka, enaka, enaka.

Na tem mestu nimamo več nobene informacije, na podlagi katere bi lahko ločili med temi tremi primeri. Torej smo lahko prepričani le, da je nekdo lagal. Torej je nekdo zadel na loteriji, a ne vemo, kdo.

Računalniško ozadje

Protokol, ki ga uporabljajo Ula, Zala in Žan v nalogi, je poznan kot *Dining Cryptographers protocol*. To je zanimiv primer komunikacije, pri kateri tako pošiljatelj kot prejemnik sporočila ostaneta anonimna, saj ju ne moremo izslediti. Za anonimne komunikacijske mreže, ki temeljijo na tem problemu, se pogosto uporablja tudi izraz DC-mreže, kjer DC pomeni »dining cryptographers«. Dandanes je anonimnost zelo pomembna, čeprav včasih tudi kontroverzna, zadeva na internetu, kjer se osebni podatki pogosto zbirajo tudi brez privoljenja uporabnikov.



Borut se je začel učiti angleščino in ga zanima, kako zvenijo različne besede. Zato je vsako besedo pretvoril v štirimestno kodo z uporabo naslednjega postopka:

1. Prvo črko besede ohrani.
2. Pobriši vse pojavitve črk A, E, I, O, U, H, W, Y.
3. Preostale črke spremeni v številke po preslikovalni tabeli na desni.
4. Dve ali več enakih zaporednih številke zamenjaj za eno samo.
5. Uporabi le prve štiri znake tako pretvorjene besede (po potrebi na koncu dodaj ničle).

Črka	se preslika v številko
B, F, P, V	1
C, G, J, K, Q, S, X, Z	2
D, T	3
L	4
M, N	5
R	6

Nekaj primerov:

Beseda	Koda
BOB	B100
BEAVER	B160
HEILBRONN	H416
ESSAY	E200

Kakšno kodo dobimo iz besede HILBERT?

Rešitev

Pravilen odgovor je H416.

1. Obdržimo prvo črko besede (H).
2. Brišemo I in E (HLBRT).
3. Črke zamenjamo s številkami (H4163).
4. Obdržimo le prve štiri znake (H416).

Računalniško ozadje

Postopek, ki ga opisuje naloga, se imenuje Soundex.

Soundex je fonetični algoritem za indeksiranje imen z glasovi, kot se izgovarjajo v angleščini. Taki algoritmi so zelo uporabni pri iskanju. Enako tehniko lahko uporabimo tudi za fonetične popravke, kadar uporabnik napačno zapiše besedo v povpraševanju (zaporedje besed), ker zveni enako kot beseda, ki jo želi uporabiti.

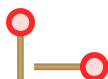


Jana in Rok se igrata na travniku z modelom letala. Po vsakem pristanku ga morata poiskati v visoki travi. Zato se Jana povzpne na manjšo vzpetino, od koder ima pregled nad travnikom ter lahko vidi letalo in usmerja Roka, ko ga gre pobrat. Jana Roka usmerja s pomočjo signalnih loparčkov. Skupaj sta določila naslednjo kodo za sporočanje pozicije letala:

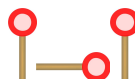
Levo



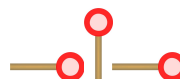
Desno



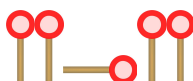
Proti hribu



Stran od hriba



Žal pa je v tej kodi napaka, saj je lahko dvoumna, če signale pošiljamo brez prekinitve. Tako lahko na primer naslednje zaporedje signalov razumemo na dva načina:



Zaporedje lahko razumemo kot »Levo, Proti hribu, Levo« ali pa tudi kot »Levo, Desno, Levo, Levo«. Jana in Rok bosta morala ponovno pregledati dogovorjene signale, da rešita ta problem.

Le ena od spodnjih kod je taka, da zaporedja signalov nikoli ne moremo razumeti na dva različna načina. Katera?

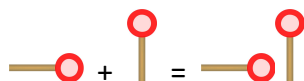
- | | Levo | Desno | Proti hribu | Stran od hriba |
|----|-------------|--------------|--------------------|-----------------------|
| A. | | | | |
| B. | | | | |
| C. | | | | |
| D. | | | | |

Rešitev

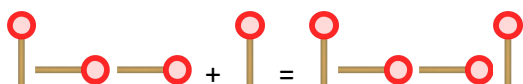
Pravilen odgovor je C.

Do rešitve lahko pridemo tudi tako, da izločimo ostale možne odgovore.

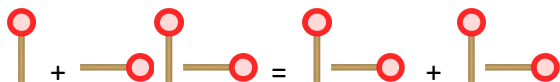
Odgovor B ni pravilen, saj je zaporedje signalov za Levo in Desno enako signalu Proti hribu:



Odgovor D tudi ni pravilen, saj je signal za Levo, ki mu sledi signal za Stran od hriba, enak signalu Proti hribu:



Tudi za odgovor A lahko ugotovimo, da ni pravilen. Signal za Levo, ki mu sledi signal za Stran od hriba, ima enako kodo kot dvakrat Desno:



In kako vemo, da je C res pravilna koda in da nikoli ne bomo imeli dveh različnih zaporedij ukazov, ki bosta imeli enako zaporedje signalov? Vidimo lahko, da je vsak ukaz zakodiran kot pokončen loparček, ki mu sledi nič ali več vodoravnih loparčkov. Torej, vsakič ko Rok vidi pokončen loparček, ve, da se je začela koda novega ukaza. Za dekodiranje sporočila mora tako le prešteti število vodoravnih loparčkov med dvema navpičnima.

Računalniško ozadje

Tudi računalniki si izmenjujejo sporočila in ukaze s pomočjo zaporedja signalov, pa naj komunicirajo preko žice ali pa preko zraka v brezžičnem omrežju. Običajno imamo za vsak signal dve možnosti, navadno ju označimo z 1 in 0, podobno kot signalni loparčki Jane in Roka.

Način, kako sporočila in ukaze prevedemo v signale, se imenuje (binarna) *koda*. Če za kodiranje posamičnih ukazov uporabljamo različno število signalov, je to *koda spremenljive dolžine*.

Pri tem je zelo pomembno, da lahko prejemnik zakodiranega sporočila signale brez napak pretvori nazaj v prvotno sporočilo. Zato moramo biti zelo previdni pri izbiri kodiranja (podobno kot Jana in Rok).

Posebna vrsta kode, ki zagotavlja enolično dekodiranje sporočila, je *prefiksna koda*. Pri prefiksni kodi se nobena kodna beseda ne začne z zaporedjem signalov kake druge kodne besede. Primer prefiksne kode je naslednji:



Prefiksne kode imajo zelo lepe lastnosti: lahko so zelo kratke in jih je zelo lahko dekodirati. Zato jih ne uporabljamo le za komunikacijo temveč tudi v različnih algoritmih za stiskanje podatkov (pa tudi v nalogah Bobra).

A cartoon illustration of a green crocodile with a row of white, stylized teeth along its back. The crocodile is green with a lighter green underbelly and has large, expressive eyes. The teeth are white and have various shapes, some resembling the letters 'D', 'E', 'K', 'A', 'E', 'D', 'A', 'D', 'A', 'D'.

98



V baletni šoli pripravljajo predstavo, v kateri bodo vse balerine nastopile tudi v duetu. V šoli je 6 balerin: Anja, Barbara, Cirila, Dani, Ema in Fani. Nastopile bodo v naslednjih duetih:

- Anja - Barbara
- Ema - Dani
- Anja - Ema
- Barbara - Cirila
- Dani - Anja
- Fani - Barbara
- Cirila - Ema
- Barbara - Dani
- Dani - Fani
- Fani - Ema

Njihova učiteljica želi razvrstiti vse duete tako, da se bo po vsaki točki na odru zamenjala le ena balerina. Če bi na primer prvi nastopili Anja in Ema, bi lahko za njima prišel duet Cirila in Ema. Poleg tega nobena balerina ne more odplesati več kot dveh točk zapored, saj bi bilo to preveč utrudljivo. Vsak duet seveda nastopi le enkrat.

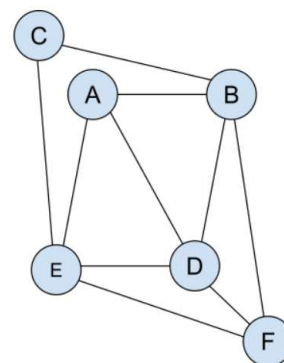
Katera balerina **ne more** nastopiti v prvem duetu ne glede na razpored, ki ga pripravi učiteljica?

Rešitev

Pravilen odgovor je Cirila.

Poglejmo diagram na sliki.

V diagramu je balerina predstavljena s krogom. Dva kroga sta povezana, če dve balerini plešeta v duetu. Razpored duetov na predstavi, ki sledi postavljenim pravilom, ustreza poti skozi ta diagram: začnemo z nekim krogom in sledimo povezavam med krogi, dokler vsake povezave ne obiščemo natanko enkrat.



Na tej poti moramo vsak krog, do katerega pridemo, tudi zapustiti, kar pomeni, da mora imeti vsak krog sodo število povezav. Izjemi sta le prvi in zadnji krog na naši poti. Torej je lahko prvi krog na naši poti le A ali F.

Ne glede na vrstni red duetov, ki ga izbere učiteljica, mora prva nastopiti ali Anja ali Fani. Edina balerina, ki ne pleše v duetu z Anjo in s Fani, pa je Cirila. Zato na predstavi ne more nastopiti v prvem duetu.

Priznati moramo še, da smo pri nalogi na tekmovanju pozabili dopisati stavek, da sme balerina odplesati največ dve točki zapored. Če te omejitve ni, lahko začnemo tudi s Cirilo. (Naloge zato seveda nismo upoštevali pri točkovanju. Nihče ni nezmotljiv – in nikomur naj ne bo nerodno, ko ugotovi, da se je zmotil.)

Računalniško ozadje

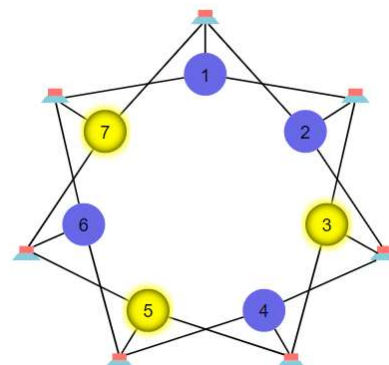
Diagram, ki smo ga narisali pri iskanju rešitve naloge, je graf. Pot, ki smo jo iskali v grafu, pa je Eulerjeva pot.

Graf je zelo prikladen za izdelavo modela problema, saj nam omogoča, da vidimo njegovo strukturo. Graf sestavljajo vozlišča (krogi) in povezave, ki povezujejo dve vozlišči. Vozlišča so v naši nalogi predstavljala balerine, povezave pa plesni duet. Pri reševanju problema smo morali uporabiti vse duete (vse povezave) tako, da smo vsak duet (vsako povezavo) uporabili natanko enkrat. Euler je dokazal, da je to možno le v primeru, če začnemo in tudi končamo v vozlišču, ki ima liho število povezav. Zato tako pot imenujemo Eulerjeva pot.

Grafe uporabljamo tudi za načrtovanje poti in razvrščanje.

Luči so povezane s stikali, kot prikazuje slika. Ko prekllopimo stikalo, vse tri luči, ki so povezane s tem stikalom, spremenijo stanje: če je luč prižgana, se ugasne in če je ugasnjena, se prižge.

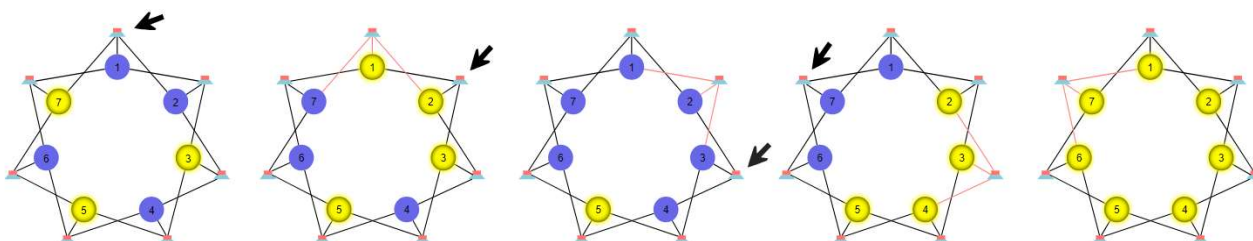
Pri katerih lučeh moramo prekllopiti stikala, da bodo vse luči prižgane?



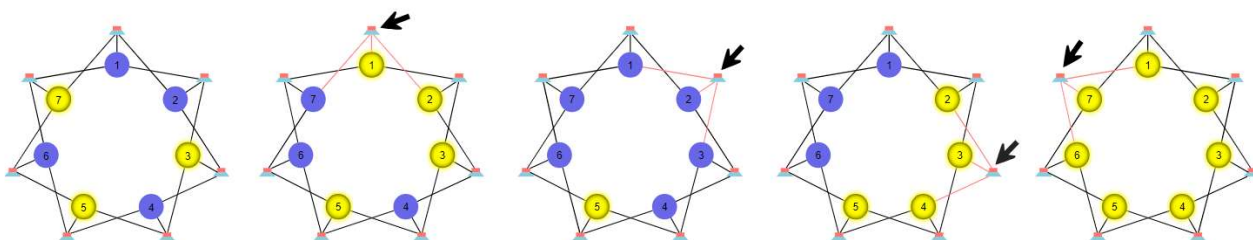
Rešitev

Pravilen odgovor je stikala 1, 2, 3 in 7.

Da prižgemo vse luči, lahko uporabimo stikala ob lučeh 1, 2, 3 in 7. Vrstni red ni pomemben. Posamezna stanja prižganih oz. ugasnjenih luči, če najprej pritisnemo 1, nato 2, 3, in 7, prikazuje spodnja slika.



Do rešitve najlažje pridemo, če sklepamo nazaj od končnega stanja, kjer so vse luči prižgane. Če želimo, da uporaba zadnjega stikala prižge vse luči, morajo biti luči, ki so povezane s tem stikalom, ugasnjene. Zato moramo najprej ugasniti nekaj luči. Z uporabo stikala pri 1 in nato še stikala pri 2 lahko ugasnemo luči 1, 2, 3 in 7. Tako gori le še luč 5. Sedaj lahko uporabimo stikali pri 3 in 7, da prižgemo še vse preostale luči.



Pri iskanju rešitve smo si pomagali z nekaj lastnostmi našega omrežja:

- Če dvakrat uporabimo stikalo, ne naredimo nič: ko drugič uporabimo isto stikalo, izničimo učinek prve uporabe stikala.
- Učinek uporabe katerega koli para stikal X in Y (v tem vrstnem redu), je enak učinku uporabe stikala Y pred stikalom X.

Računalniško ozadje

Pri veliko problemih natančno poznamo cilj in tudi začetno stanje, iz katerega izhajamo. V tej nalogi je bil cilj, da v omrežju prižgemo vse luči, v začetnem stanju pa so bile prižgane le luči 3, 5 in 7 (vse ostale pa so bile ugasnjene).

V začetnem obdobju uporabe umetne inteligence je reševanje problemov zajemalo iskanje zaporedja akcij, ki nas pripeljejo od začetnega stanja do želenega ciljnega stanja. Tak pogled je razumljiv, če imamo končno množico vseh možnih akcij in tudi končno število objektov, ki jih lahko manipuliramo s temi akcijami. Pri reševanju te naloge smo uporabili pristop dvosmernega iskanja, pri katerem rešitev iščemo v dveh smereh: iz začetnega stanja in iz končnega stanja hkrati.



Na kupu je pet popularnih knjig za srednješolce:

- Zločin in kazen
- Bratje Karamazovi
- Vojna in mir
- Tihi Don
- Ana Karenina

Radi bi jih uredili po abecedi, tako da bo na vrhu kupa knjiga, ki je po abecedi prva po vrsti. To bomo počeli tako, da bomo vzeli določeno število knjig (lahko tudi vse) z vrha kupa (vedno z vrha!) in jih zložili po eno in eno nazaj. Če vzamemo in preložimo, recimo, tri knjige, se bo vrstni red spremenil v:

- Vojna in mir
- Bratje Karamazovi
- Zločin in kazen
- Tihi Don
- Ana Karenina

S tem seveda nismo nič bližje rešitvi. Morda smo celo dlje, zato se vrnimo k prvemu razporedu.

Urediti jih želimo s čim manj koraki – pri čemer je en korak eno jemanje knjig z vrha kupa in zlaganje nazaj. Koliko korakov potrebujemo, da bodo urejene po abecedi?

Rešitev

Na prvi (a napačni) pogled jih potrebujemo pet.

Za lažje razmišljanje si predstavljajmo, da urejamo zaporedje števil 5 2 3 4 1. Strategija je lahko, da v vsakem koraku spravimo največje število, ki še ni na svojem mestu, na začetek in od tam na mesto, kamor sodi.

5 2 3 4 1 → 1 4 3 2 5 → 4 1 3 2 5 → 2 3 1 4 5 → 3 2 1 4 5 → 1 2 3 4 5

Pred vsako potezo so z mastnim tiskom izpisana števila, ki jih preobrnemo v naslednjem koraku. 5 je že na začetku, tako da ga le prestavimo na konec. V naslednjih dveh korakih spravimo 4 na začetek in od tam na konec. Nato spravimo 3 na začetek in od tam na konec. Ker imamo srečo, sta 1 in 2 že urejeni.

Gre pa hitreje: v resnici zadoščajo štirje koraki.

5 2 3 4 1 → 1 4 3 2 5 → 2 3 4 1 5 → 4 3 2 1 5 → 1 2 3 4 5

Računalniško ozadje

Problem, ki ga rešujemo v tej nalogi, se imenuje problem urejanja palačink: imamo kup različno velikih palačink, ki jih želimo urediti po velikosti tako, da v vsakem koraku obrnemo določeno število palačink na vrhu. Problem je zanimiv iz več razlogov.

Prvi je, da se ga je domislil nek matematik (očitno!), Jacob E. Goodman, ko je zlagal sveže oprane brisače. Ker so bile različno velike, je hotel kup urediti, vendar je imel premalo prostora, da bi jih razložil po kaki mizi in polici, zato je bil prisiljen le obračati tiste na vrhu.

O tem je napisal članek, vendar se mu je zdelo primerneje govoriti o palačinkah, poleg tega pa se je bal, da ga kolegi zaradi ukvarjanja z neresnimi problemi ne bodo imeli za resnega matematika, zato je članek napisal pod psevdonimom.

Nadalje je problem zanimiv zato, ker še vedno ne vemo, koliko prestavljanj potrebujemo za najbolj zoprno namešan kup n palačink. Trenutno se ve, da za n palačink potrebujemo vsaj $15/14 n$, vendar ne več kot $18/11 n$ urejanj. To vemo od leta 2008, najboljša ocena zgornje meje pred tem pa je bila $5(n + 1)/3$. Leta 1979 je ni odkril nihče drug kot Bill Gates, kasnejši soustanovitelj Microsofta.