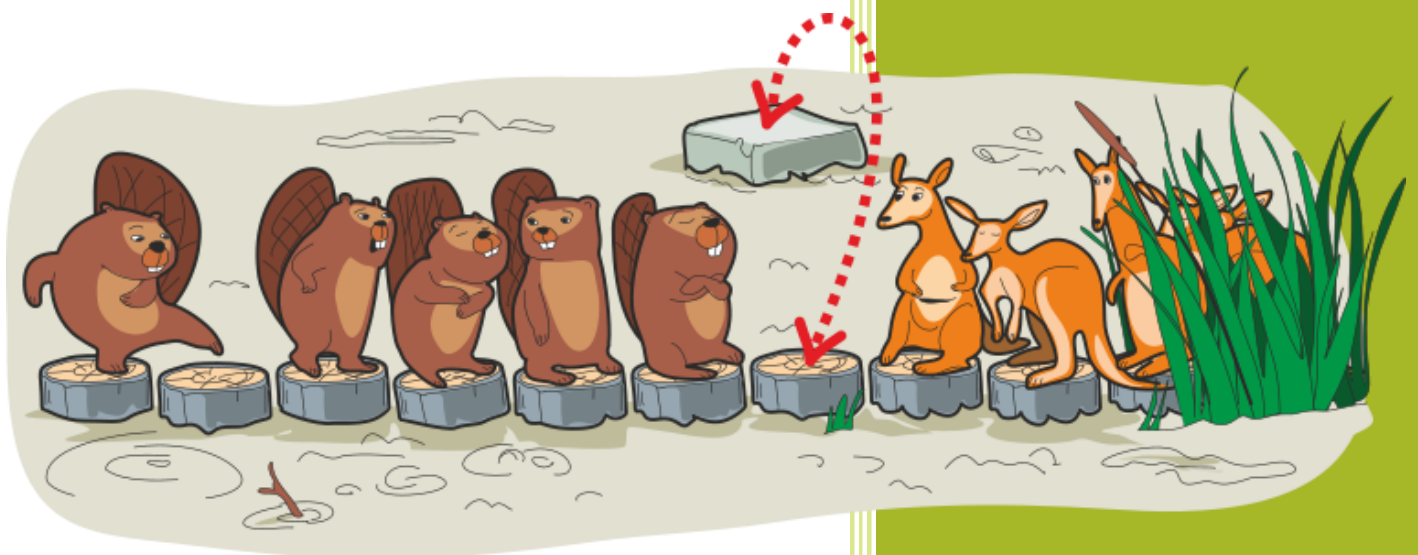


Bober 2020/21

Naloge in rešitve šolskega tekmovanja



Bebras

ACM Slovenija



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

Alenka Kavčič (UL FRI)

Anja Koron (OŠ Branik)

Janez Demšar (UL FRI)

Špela Cerar (UL PEF)

Razvoj tekmovalnega sistema:

Gašper Fele Žorž (UL FRI)

Gregor Jerše (UL FRI)

Avtor slike na naslovnici

Vaidotas Kinčius, Litva

Kazalo nalog

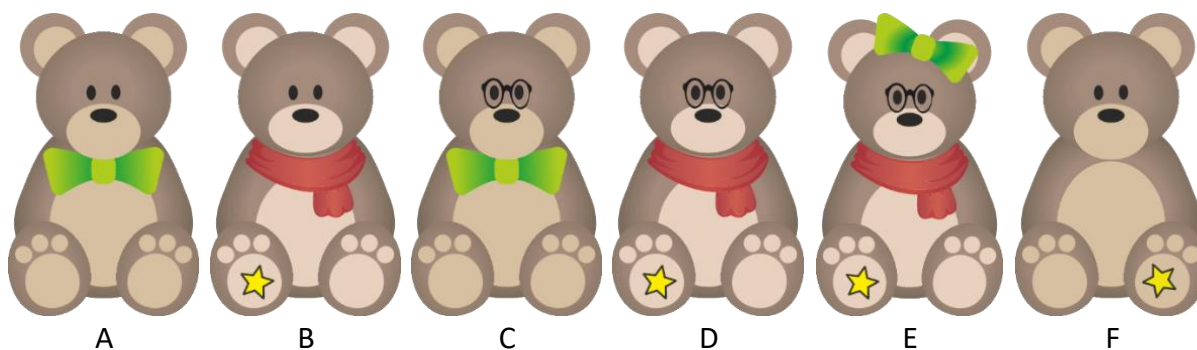
IZBERI MEDVEDKA	1	POSKOČNI KENGURU	29
BOBROV SALON	2	SESTAVLJANKA	30
SKRIVNA SPOROČILA	3	LJUBEŽENSKA PISMA	31
VESOLJSKI ZOO	4	EPIDEMIJA	32
ZNAMKE	5	TOPLOTNA KARTA	34
OGLED RAZSTAVE	6	VEDRA Z VODO	35
SKODELICE	8	ZMEDENI POŠTAR	36
NAJTEŽJA ŠKATLA	9	NALEPKE	37
OTOK ZAKLADOV	10	POTUJOČA PEKARNA	39
VISOKI STOLPI	11	PREPISOVANJE	40
POBRISANA TABELA	12	REZULTATI PREISKAV	42
BARVANJE HIŠE	14	TELEFONČEK	44
DREVESNA STRUKTURA	16	TRGOVINE	45
ZAPOMNI SI GESLO	18	RAČUNSKI STROJ	46
ZVEZDICE IN LUNE	19	ŠAH MAT	47
KNJIŽNICA	20	FRNIKOLE V ZABOJU	49
GLEDALIŠKA PREDSTAVA	21	TOVARNA AVTOMOBILOV	51
VEZENJE	22	ELEKTRIČNI AVTOMOBILI	53
DREVESNI SUDOKU	23	DREVESNI SUDOKU	54
VRATAR JAKA	25	PREGLED NALOG	57
NOVA HIŠA	26	AVTORJI NALOG	58
POLAGANJE PLOŠČIC	28		

Izberi medvedka

2. razred



REN BO V ŠOLI PREDSTAVIL SVOJEGA NAJLJUBŠEGA MEDVEDKA. TA IMA NA NOGI ZVEZDO, NOSI ŠAL ALI PENTLJO, VENDAR NIMA OČAL. KATEREGA BO PREDSTAVIL?



REŠITEV

REN BO PREDSTAVIL MEDVEDKA B, KI IMA ŠAL IN ZVEZDICO.

MEDVEDKA A NE BO PREDSTAVIL, KER NIMA ZVEZDE NA NOGI.

MEDVEDKOV C, D IN E NE BO PREDSTAVIL, KER IMAJO OČALA.

MEDVEDKA F NE BO PREDSTAVIL, KER NIMA NITI ŠALA NITI PENTLJE.

Računalniško ozadje

Pri programiranju moramo večkrat preveriti, ali drži več pogojev hkrati, zato da program, ki ga napišemo, pravilno deluje.



V BOBROVEM SALONU DELAJO ŠTIRI STILISTKE. V TABELICI LAHKO PREBEREŠ, KOLIKO ČASA POTREBUJEJO ZA KATERO STORITEV.

VSE SO PRAVKAR ZAČELE OPRAVLJATI SVOJO STORITEV, KOT KAŽE SLIKA. ČIM STILISTKA KONČA Z DELOM, NADALJUJE Z NASLEDNJO STRANKO.

STORITEV	ČAS
ČESANJE KOŽUHA	5 MINUT
OSTRENJE ZOB	8 MINUT
GLAJENJE REPA	15 MINUT



OBI JE EDINI V ČAKALNICI. KATERA STILISTKA BO ZGLADILA NJEGOV REP?

REŠITEV

PRVA BO Z DELOM ZAKLJUČILA ZORA, SAJ ČESANJE KOŽUHA TRAJA 5 MINUT. ČEZ 5 MINUT BO ZORA ZAČELA GLADITI OBIJEV REP. OSTALE STILISTKE PA BODO ŠE NAPREJ DELALE S SVOJIMI PRVIMI STRANKAMI.

Računalniško ozadje

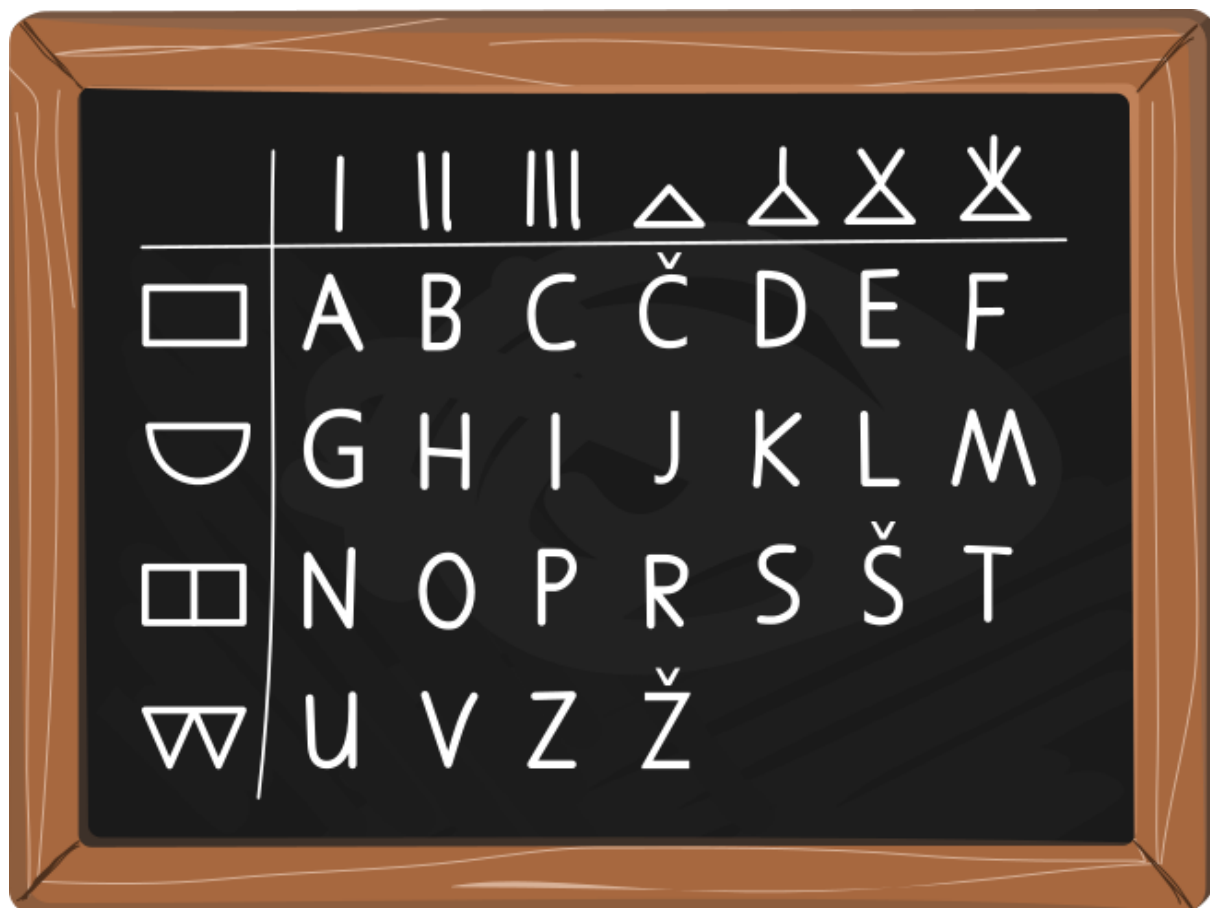
Tudi v računalniku istočasno teče več programov, operacijski sistem (na primer Windows, macOS, Linux) pa razporeja delo.

SKRIVNA SPOROČILA

2. razred



BOBRI SI POŠILJAJO SPOROČILA S SKRIVNO PISAVO.



KAJ PIŠE V TEM SPOROČILU?



REŠITEV

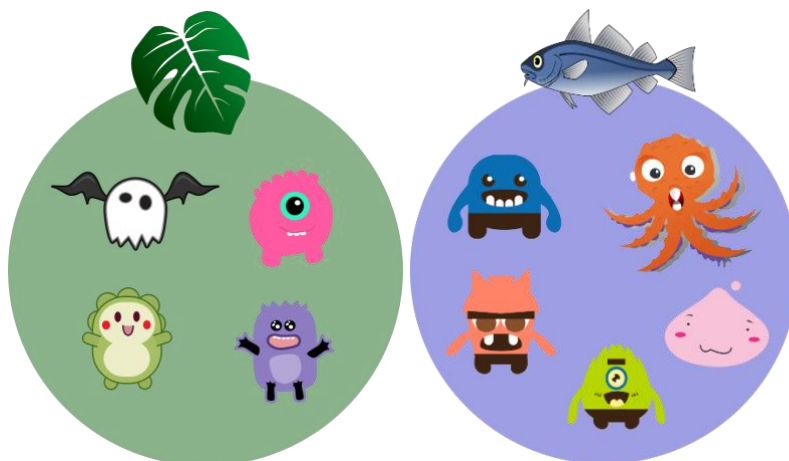
GREMO NA MORJE

Računalniško ozadje

Ko po internetu pošiljamo pomembna sporočila, ne želimo, da bi jih lahko vsi znali prebrati. Da bi ostala tajna, jih računalniki pred pošiljanjem zašifrirajo.



V VESOLJSKEM ZOO IMAJO RASTLINOJEDE IN RIBOJEDE VESOLJCE:



V VSAKI OGRADI JE VSAJ EN RASTLINOJEDEC. KATERIH OGRAD **NI** V VESOLJSKEM ZOO?



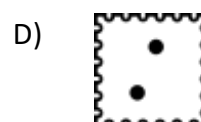
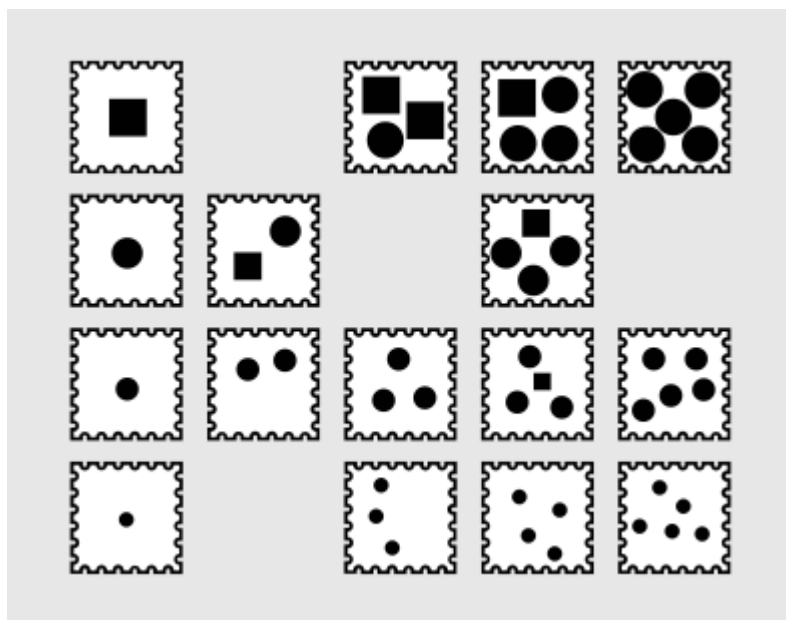
Rešitev

V PRVI IN TRETJI OGRADI SO SAMO RIBOJEDI VESOLJCI, ZATO TEH DVEH NI V VESOLJSKEM ZOO. V DRUGI OGRADI STA DVA RASTLINOJEDCA (BELI IN ROZA VESOLJEC). V ČETRTE OGRADI JE VIJOLIČNI RASTLINOJEDI VESOLJEC.

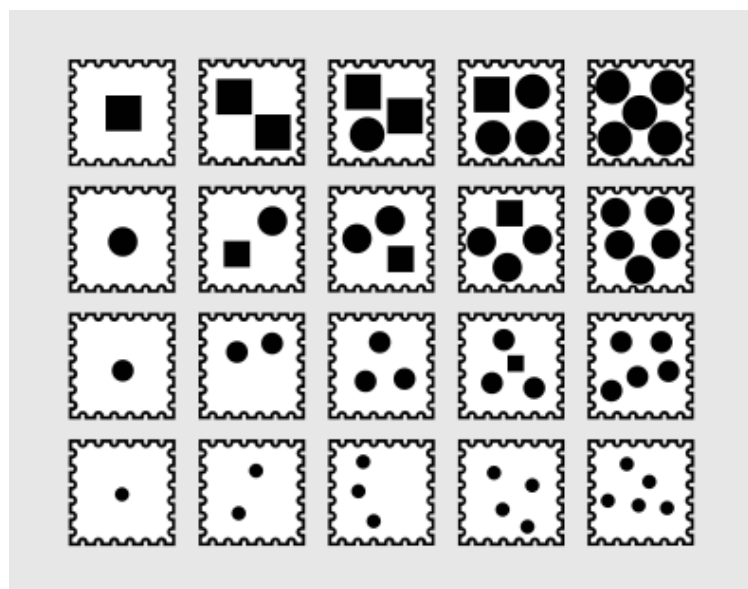
Računalniško ozadje

V računalništvu velikokrat preverjamo, če neka trditev drži ali ne. V tej nalogi smo preverjali, v katerih ogradah ni rastlinojedcev.

BOBER UREJA ZNAMKE. OSTALE SO MU ŠE ŠTIRI. POSTAVI JIH NA PRAVA MESTA.



REŠITEV



V TRETJEM STOLPCU SO NA VSEH ZNAMKAH PO TRIJE LIKI, ZATO SODI ZNAMKA A V TA STOLPEC.

ZNAMKA B SODI V PETI STOLPEC, KJER JE NA VSEH ZNAMKAH PO PET LIKOV.

ZNAMKI C IN D IMATA PO DVA LIKA, ZATO SODITA V DRUGI STOLPEC.

ZNAMKA D SODI V ČETRTO VRSTICO, SAJ SO V TEJ VRSTICI NA VSEH ZNAMKAH MAJHNI KROGCI.

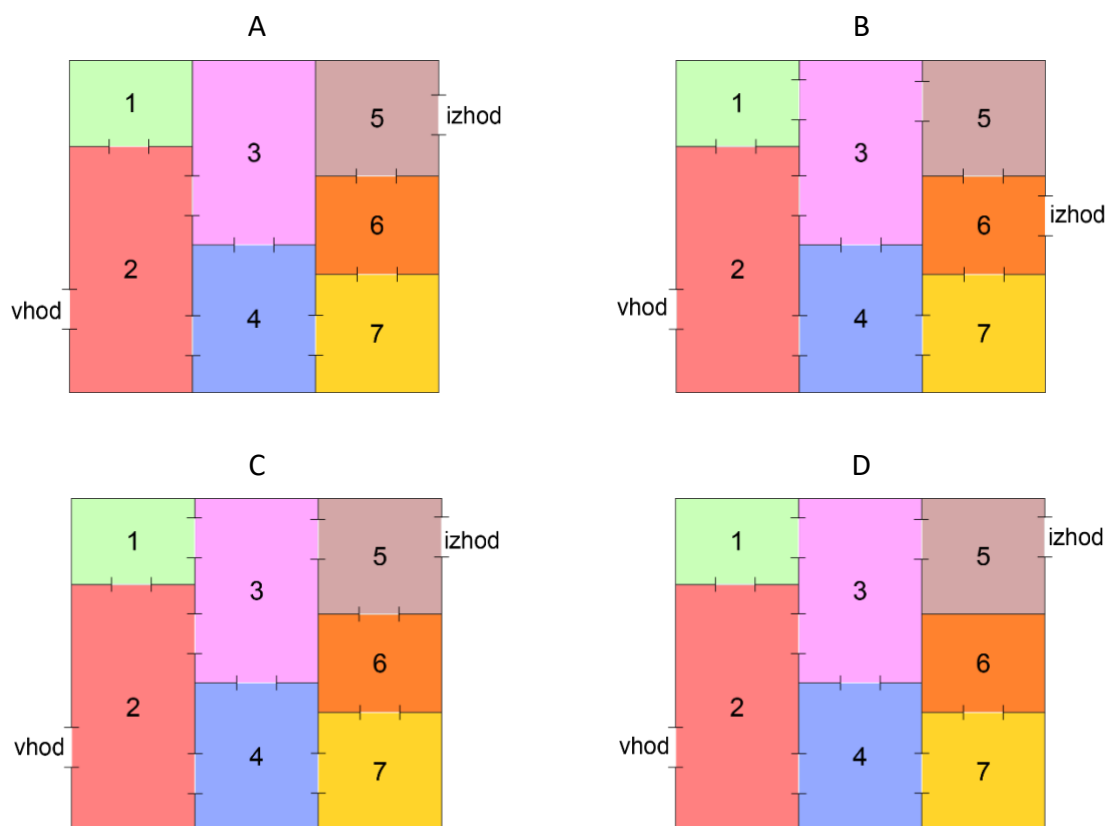
ZNAMKA C SODI V PRVO VRSTICO, SAJ SO V TEJ VRSTICI NA VSEH ZNAMKAH LE VELIKI LIKI.

Računalniško ozadje

V računalništvu pogosto urejamo podatke po različnih lastnostih. V nalogi so znamke v stolpcih urejene glede na število likov na njih in po vrsticah glede na velikost likov.



Za nov muzej so bile predlagane štiri različne prostorske razporeditve (glej spodnje slike). Kateri razpored omogoča obiskovalcem, da vstopijo v muzej, gredo skozi vsako sobo natančno enkrat (v nobeno ne vstopijo dvakrat!) in nato muzej zapustijo?



Rešitev

Le razporeditev prostorov v primeru C omogoča obiskovalcem ogled razstave tako, da ne obišejo nobenega prostora dvakrat. Zaporedje sob, s katerim to dosežejo, je 2, 1, 3, 4, 7, 6, 5.

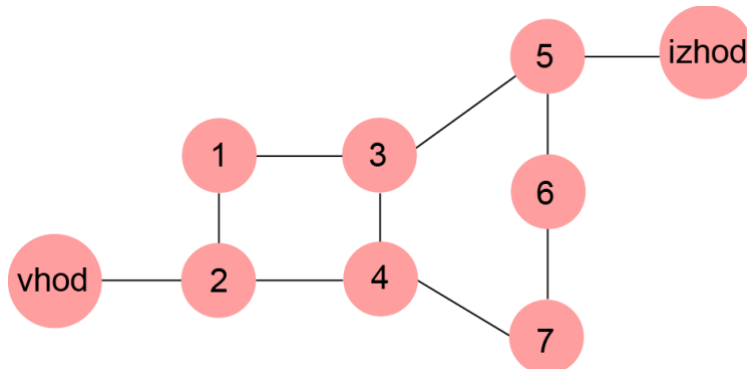
V splošnem tako predpisan ogled ni mogoč, če ima katera soba le en vhod, saj v tem primeru obiskovalec, ki v sobo vstopi, te ne more zapustiti drugače, kot da uporabi ista vrata, kar pomeni, da prejšnjo sobo obiše dvakrat.

Primer A ni ustrezen, saj vsebuje sobo 1, ki ima le en vhod. Prav tako ni ustrezen primer D, saj ima soba 6 le en vhod.

V primeru B lahko obiskovalci do sobe 6 pridejo iz sobe 5 ali 7. Če obiskovalec izbere sobo 5, potem lahko obiše sobo 7 le tako, da si ogleda razstavo v sobi 6 dvakrat, in obratno.

Računalniško ozadje

Pri iskanju ustrezne poti ogleda galerije je pomembno le, da vemo, katere sobe so med seboj povezane. Tako je mogoče razporeditev sob v primeru C predstaviti na naslednji način:

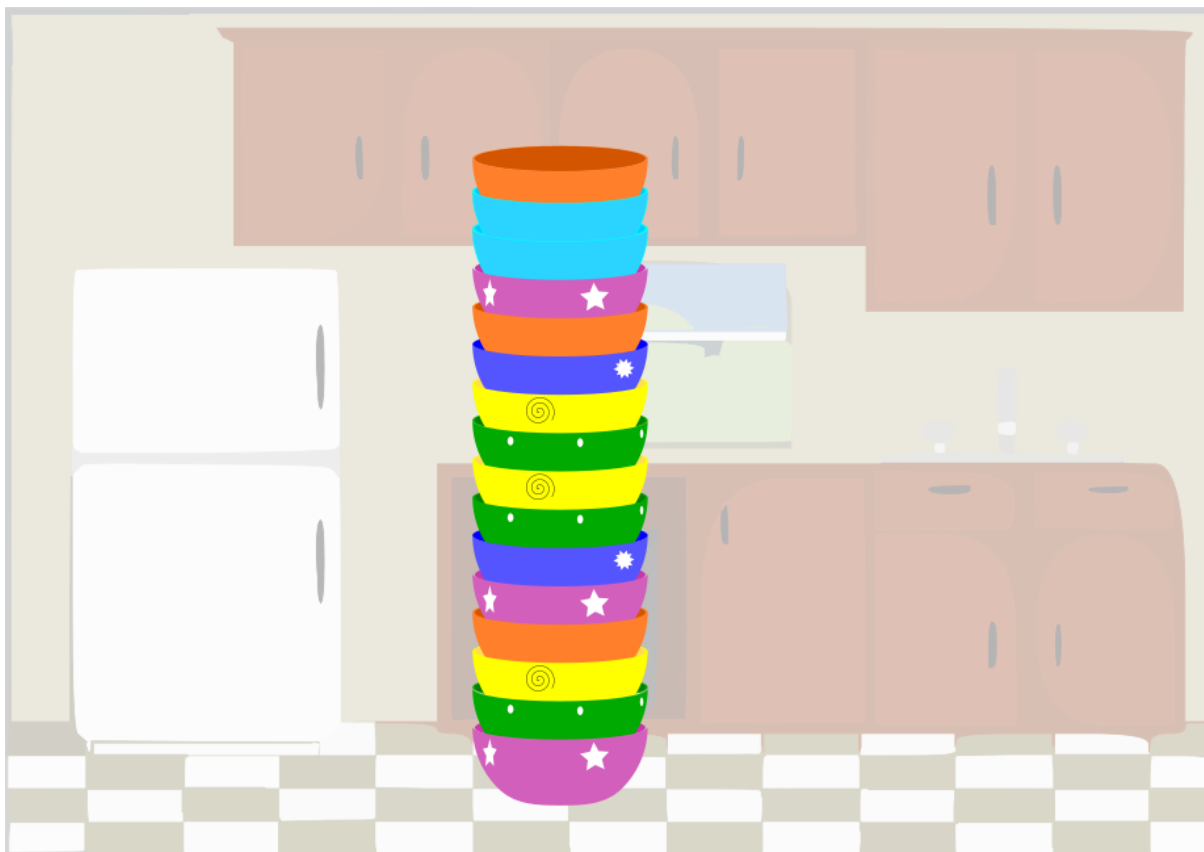


Krog predstavlja posamezno sobo, črta pa vrata. Tak model imenujemo graf. Obstaja veliko problemov, ki jih lahko računalniki rešijo s pomočjo grafov, kot so problem najkrajše poti, določitev prijateljev na družabnih omrežjih idr.

V modelu grafa iskanje poti ogleda ustreza iskanju poti vzdolž povezav, tako da pri tem obiščemo vse točke. Pri tej nalogi je število sob in vrat majhno in zato je naloga rešljiva tudi brez računalnikov. Če pa bi število sob in vrat povečali, bi se lahko zgodilo, da bi imeli pri iskanju ustrezne poti težave celo računalniki.



Tri sestre želijo jesti zajtrk iz enakih skodelic. Skodelice z različnimi vzorci so zložene na kup, kot kaže slika. Skodelice lahko jemljejo le z vrha kupa po eno naenkrat.



Najmanj koliko skodelic morajo vzeti s kupa, da bodo imele 3 enake?

Rešitev

Vzeti morajo trinajst skodelic, da bodo imele vse tri oranžne skodelice.

Računalniško ozadje

Tudi računalniki pogosto shranjujejo podatke tako, da imajo dostop le do zadnjega shranjenega podatka. Taki obliki podatkov rečemo sklad.

Najtežja škatla

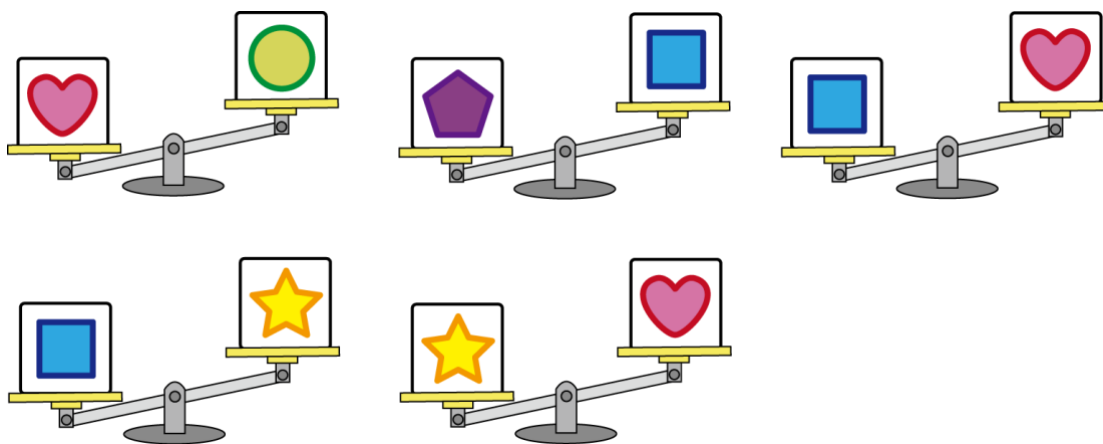
3. - 5. razred



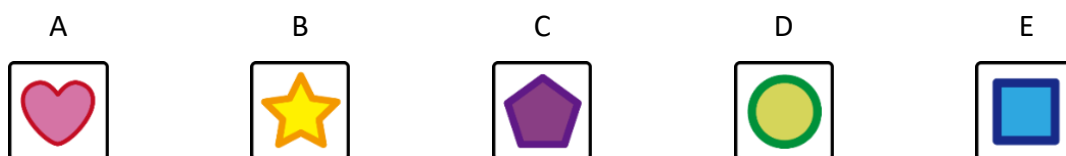
Imamo pet škatel z različni vzorci. S tehtnico lahko primerjaš teži dveh škatel.



Izvedenih je bilo pet meritev:



Katera škatla je najtežja?



Rešitev

Pravilni odgovor je C, petkotnik.

Krog je lažji od srca, srce je lažje od zvezde, zvezda od kvadrata, kvadrat od petkotnika.

Ne le, da smo izvedeli, kdo je najtežji – celo uredili smo jih od najlažjega do najtežjega, in to le s petimi tehtanji! Pa še od teh je eno nepotrebno. Katere?

Računalniško ozadje

Računalniki morajo kar pogosto poiskati informacije v veliki količini podatkov, kar je veliko lažje, če so le-ti urejeni. Poznamo kar veliko postopkov urejanja, ki se razlikujejo predvsem po tem, na kakšen način poskrbijo za to, da opravijo čim manj primerjav.

Otok zakladov

3. - 5. razred

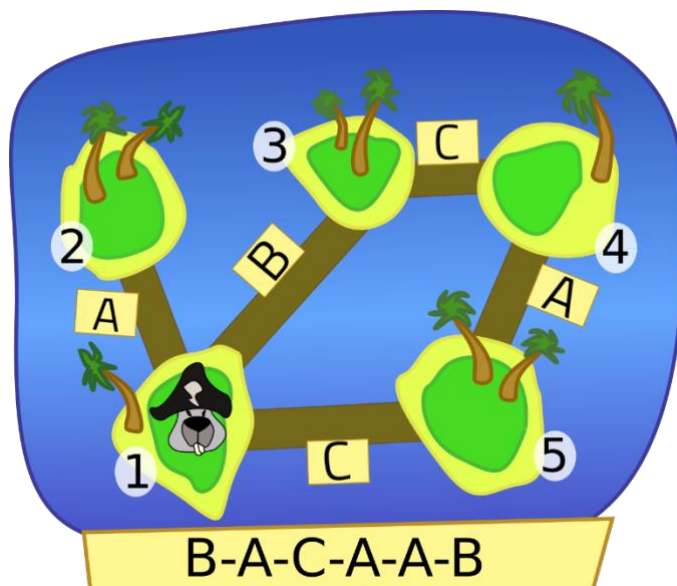


Pirat Peter živi na otoku 1, kot vidiš na zemljevidu. Dobil je skrivno kodo: B-A-C-A-A-B, ki mu sporoča, čez katere mostove mora iti, da pride do skritega zaklada.

Če je v kodi črka za ime mostu, ki ne vodi z otoka, na katerem je trenutno Peter, to črko izpusti in pogleda naslednjo.

Na primer, če je koda A-B-A, Peter prečka most A, z otoka 1 na otok 2. Ker mostu B na otoku 2 ni, ostane na otoku 2. Ker je naslednja črka A, gre čez most A z otoka 2 na otok 1.

Na katerem otoku se skriva zaklad, če je dobil kodo B-A-C-A-A-B?



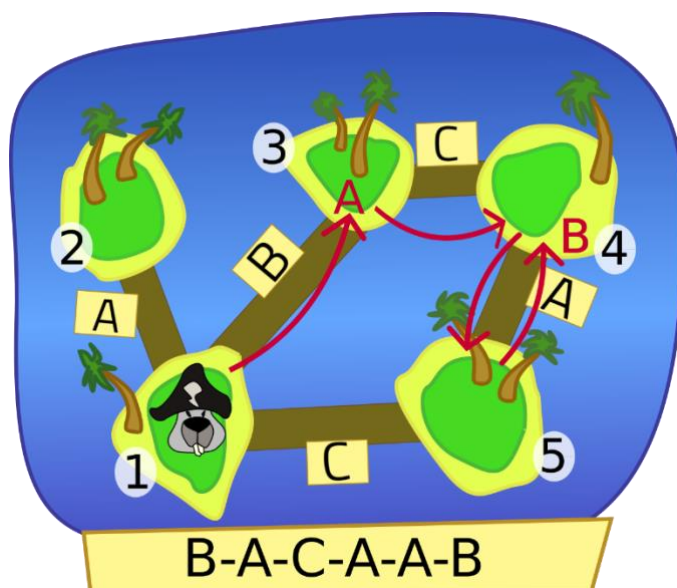
Rešitev

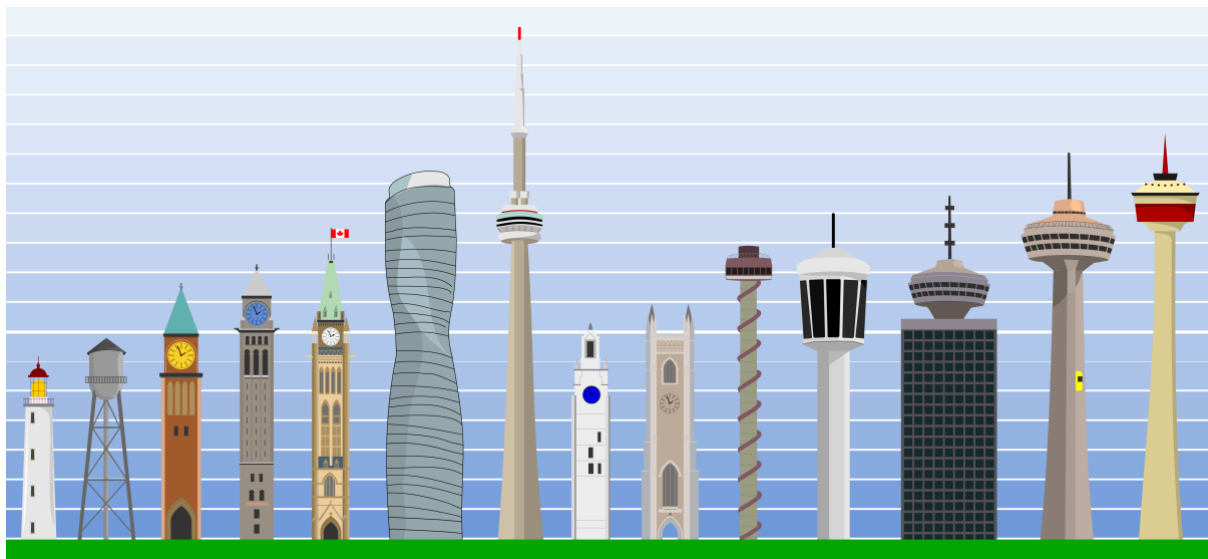
Skriti zaklad je na otoku 4.

Najprej prečka most B in je na otoku 3. Ker mostu A ni, ostane na otoku 3. Nato prečka most C na otok 4. Za tem gre čez most A na otok 5 in se vrne čez most A na otok 4. Ker mostu B z otoka 4 ni, ostane na tem otoku.

Računalniško ozadje

Piratska navodila so neke vrste program.





Na sliki je 14 stolpov. Kateri je deseti po vrsti, če jih razvrstimo od najnižjega do najvišjega?



Rešitev

Pravilni odgovor je B. Deseti najnižji stolp od štirinajstih je hkrati peti najvišji, saj je 9 stolpov nižjih od njega, 4 pa so višji. Na sliki so stolpi razdeljeni v dve zaporedji, v katerih višina narašča od leve proti desni. V našem primeru so štirje najvišji stolpi najvišja dva stolpa v levem in najvišja dva v desnem zaporedju. Primerjati moramo še tretji najvišji stolp v levem zaporedju s tretjim najvišjim v desnem zaporedju in ugotovimo, da je peti najvišji stolp tretji najvišji v desnem zaporedju.

Računalniško ozadje

Na podoben način deluje zelo hiter postopek urejanja podatkov – urejanje z zlivanjem.

Pobrisana tabela

3. - 7. razred



Bobri si pošiljajo skrivna sporočila. Ker jih rade berejo vidre, jih zapišejo s skrivno pisavo. Na žalost je bober Lovro pobrisal nekaj črk in znakov.



S pomočjo delno pobrisane tabele ugotovi, kaj je zapisano v sporočilu:



Obkroži pravilni odgovor.

- A) RAČUNALNIKI SO SUPERCA
- B) RAČUNALNIŠTVO JE ZAKON
- C) INFORMATIKA JE SUPERCA
- D) RAČUNALNIKI NISO ZAKON

Rešitev

Popolna tabela je na sliki na desni.

Ustrezen odgovor je: B)

RAČUNALNIŠTVO JE ZAKON

Črke v tabeli so očitno našteje po abecedi, torej manjkajoče črke preprosto vstavimo.

Nato opazimo, da so simboli sestavljeni iz oznak vrstic (spodnji del simbola) in stolpcev (zgornji del). Razberemo lahko, kako so videti pobrisane oznake prve vrstice in prvih dveh stolpcev.

	I	II	III	△	△	X	X
□	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			

Sedaj za določene črke že poznamo simbole, kar pomeni, da lahko le-te v zapisanem sporočilu nadomestimo s črkami. S poskušanjem lahko nato dopolnimo manjkajoče simbole, kar pa ni nujno potrebno, saj lahko z izločanjem pridemo do pravega odgovora.

Nalogo je mogoče rešiti tudi brez dopolnitve tabele: ugotoviti je potrebno, da prvi simbol ne ustreza črki I, kar izključi odgovor C, ter da deseti simbol ustreza črki Š, ki pa se pri odgovoru A in D razlikuje. Kar pomeni, da je edini možni odgovor B.

Računalniško ozadje

Zaščita podatkov je naloga, ki je starejša od 4000 let. Kriptografija, ki je ključni del informatike, se osredotoča na razvoj skrivnih pisav, s katerimi lahko zaščitimo zaupne podatke in sporočila.

Barvanje hiše

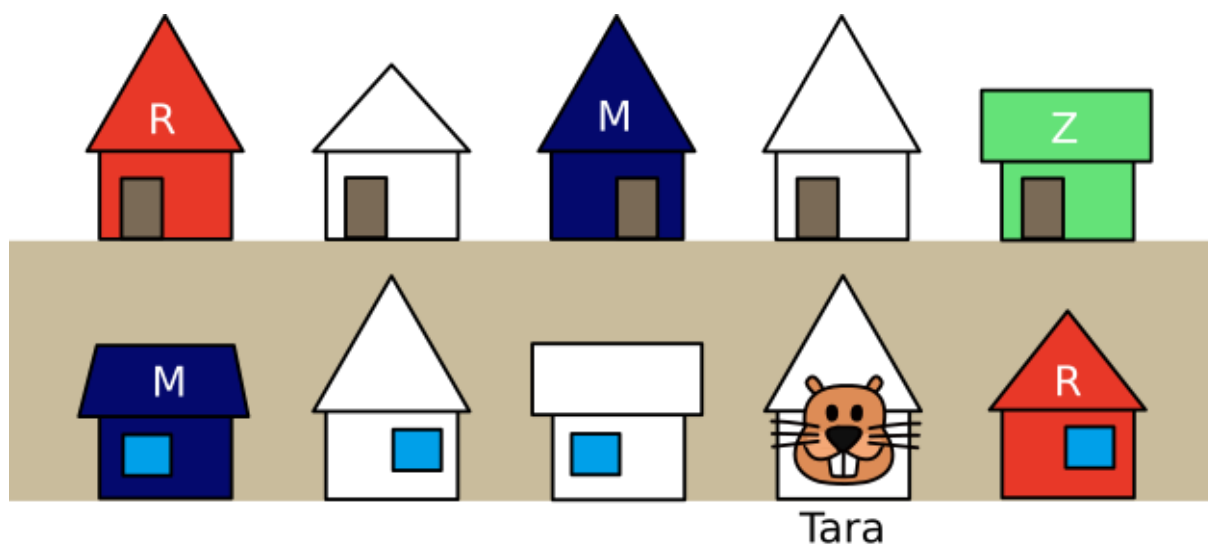
4. - 5. razred



Lastniki hiš v ulici, kjer živi Tara, so se odločili svoje hiše pobarvati po naslednjih pravilih:

1. Vsaka hiša mora biti pobarvana rdeče, modro ali zeleno.
2. Sosednji hiši ne smeta biti enake barve.
3. Hiša ne sme biti enake barve kot hiša čez cesto.

Nekaj hiš je že pobarvanih, kot kaže slika.

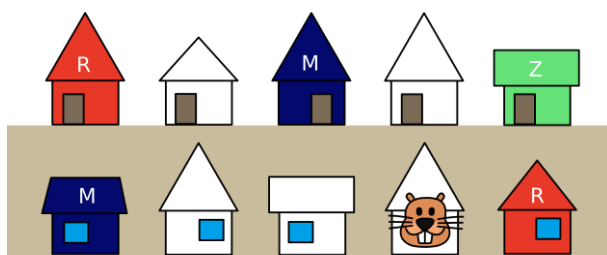


V katero bravo lahko Tara prebarva svojo hišo?

- A. Uporabi lahko samo rdečo.
- B. Uporabi lahko samo modro.
- C. Uporabi lahko samo zeleno.
- D. Uporabi lahko modro ali zeleno.

Rešitev

Tarina hiša je lahko le modra. Pravo barvo določimo na naslednji način:



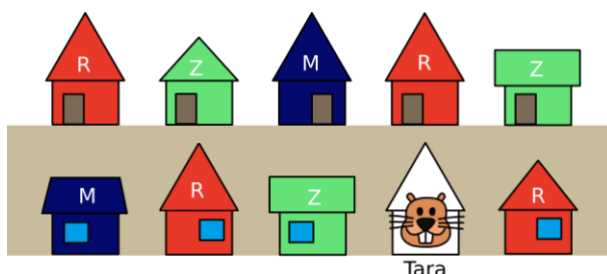
Začnemo s to sliko.



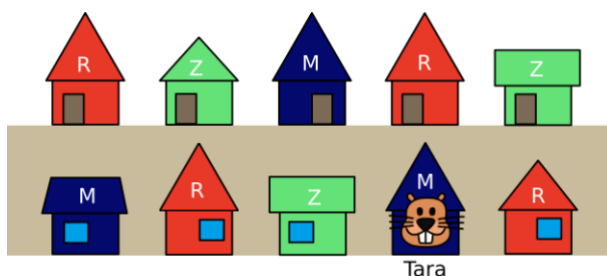
Najprej pobarvamo hiše, pri katerih poznamo barvi obeh sosednjih hiš.



Rdeče pobarvamo hišo, pri kateri poznamo barvo hiše sosedja (modra) in hiše čez cesto (zelena).



Zeleno pobarvamo hišo, pri kateri je barva hiše levega sosedja rdeča in hiše čez cesto modra.



Ker sta sosednji hiši zelena in rdeča, lahko Tara pobarva svojo le z modro bravo.

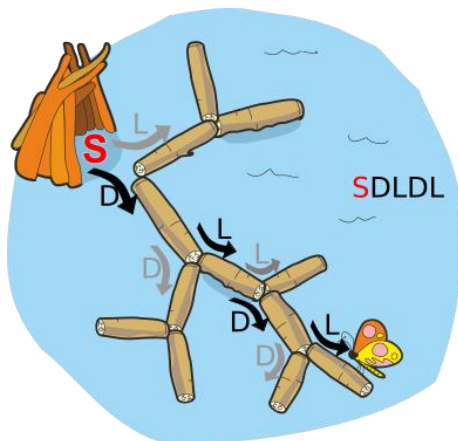
Računalniško ozadje

Problem nekoliko spominja na znani problem "barvanja grafov". Veliko problemov lahko rešimo tako, da jih prerišemo v graf (poglej, recimo, nalogo Ogled razstave, malo višje) in se jih lotimo s katerim od znanih postopkov za reševanje problemov na grafih.

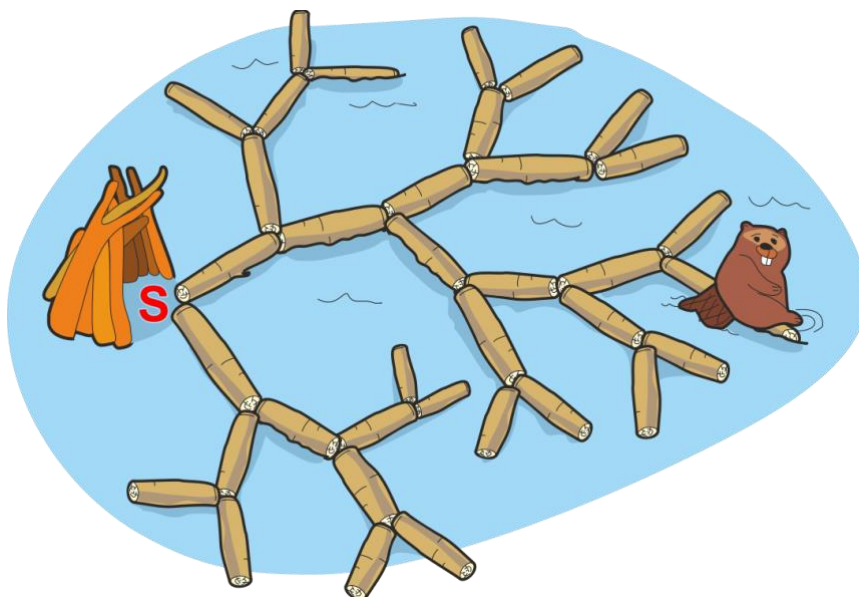


Bobri s hlodi zgradijo neverjetno strukturo, ki se začne pri njihovi koči **S**.

Pot do vsakega hloda opišejo z uporabo dveh ukazov **L** (levo) in **D** (desno). Na primer, pot do metulja se glasi: **S D L D L**



Opiši pot od koče **S** do sedečega bobra.

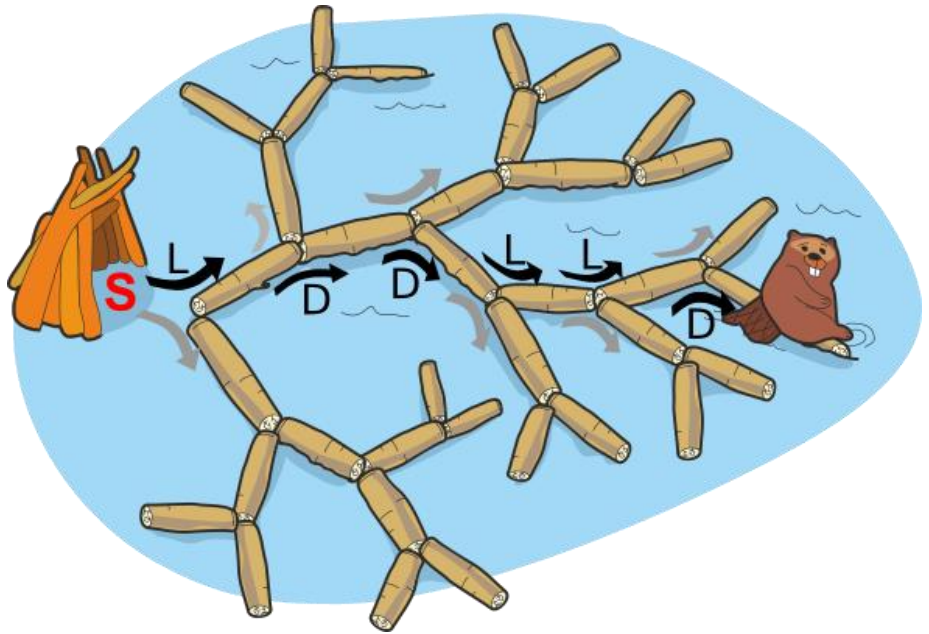


S									
----------	--	--	--	--	--	--	--	--	--

Rešitev

Pravilna rešitev je: **S L D D L L D**.

Kot je razvidno s slike, prvi premik v desno nikakor ne vodi do sedečega bobra, kar pomeni, da mora biti prvi premik obvezno v levo. Na naslednjem razcepu zopet pogledamo, na kateri strani se nahaja bobar, in ugotovimo, da moramo tokrat izbrati desno. Postopek nadaljujemo, dokler ne pridemo do bobra.



Računalniško ozadje

V računalništvu podatke pogosto uredimo v takšno obliko, da je vsak podatek povezan z dvema drugima. Temu rečemo "drevo". Drevesa so praktična predvsem zato, ker je po njih mogoče zelo hitro iskati.



Bober Fifikus želi ustvariti geslo, ki si ga bo lahko zapomnil, drugi pa ga ne bodo mogli hitro uganiti. Uporabil bo poseben postopek. Uporabil bo stavek: **Jaz rad jem golaž, ki ga naredi kuhar Andrej.**

To poved bo uporabil, da ustvari svoje geslo: **J3r3j3g5k2g2n6k5A6**

Uporabi isto metodo in ustvari geslo iz stavka: **Računalništvo me uči učiteljica Alenka.**

Rešitev

Bober je uporabil posebno metodo, pri kateri je vzel prvo črko vsake besede, k njej pa napisal, koliko črk ima beseda:

Jaz → J3

rad → r3

jem → j3

golaž → g5

ki → k2

ga → g2

naredi → n6

kuhar → k5

Andrej → A6

Po istem postopku dobimo novo geslo:

Računalništvo → R13

me → m2

uči → u3

učiteljica → u10

Alenka → A6

Geslo iz tega stavka je tako: **R13m2u3u10A6**

Velike začetnice besed se v geslu ohranijo kot velike črke, ločil ne upoštevamo.

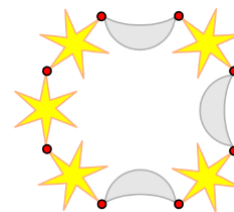
Računalniško ozadje

Kakšen pa je tvoj sistem za izmišljanje gesel? Saj menda nimaš povsod istega gesla, ne?

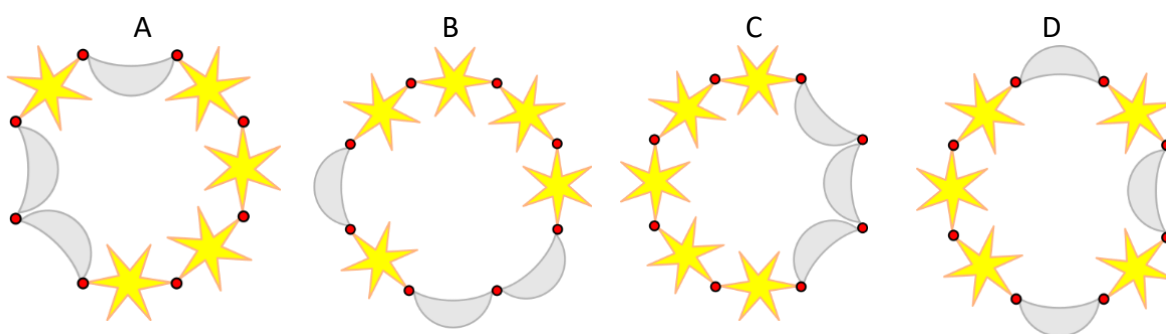


Maja si želi zapestnico, kot je na sliki na desni. Janezu je zato dala naslednja navodila:

- Vzemi zvezdico  in luno  in ju spni.
- Še dvakrat ponovi prejšnji korak.
- Vzemi tri narejene delčke in jih spni v en niz.
- Na vsako stran niza dodaj še dve zvezdici in spni oba konca niza, da nastane zapestnica.

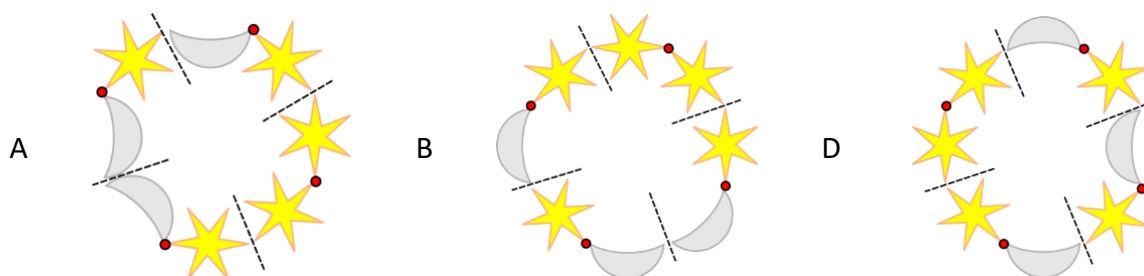


Žal mu ni dala slike, zato je lahko zapestnica, ki jo bo naredil, drugačna, četudi bo sledil navodilom. Pri kateri od spodnjih zapestnih pa **gotovo ni upošteval** Majinih navodil?



Rešitev

Ker mora biti zraven vsake lune zvezdica, ni mogoče, da bi imela zapestnica vse tri lune skupaj, kot je v primeru zapestnice C. Spodnje slike pa kažejo, kako lahko ostale zapestnice razdelimo na pare iz zvezdice in lune ter dveh zvezdic:



Računalniško ozadje

Da bo program pravilno deloval, mora programer napisati natančna navodila. Če so ta dvoumna, jih računalnik ne bo razumel ali pa izvedel na način, ki ga programer ni načrtoval.



Bobrovska knjižnica ima samo nekaj knjig. Ko si bober izposodi knjigo, knjižničar zapiše njegovo ime in mu poda knjigo z vrha kupa. Ko bober vrne knjigo, knjižničar zapiše njegovo ime in postavi knjigo na vrh kupa.

Na začetku tedna izgleda kup knjig tako:



V knjižničarjevem dnevniku za ta teden so ti podatki:



Katero knjigo si je izposodila Katja?

Rešitev

Izposodila si je knjigo Kolesar naj bo!, saj je bila ob njenem prihodu v knjižnico ta na vrhu kupa.

Knjižničar vedno izposodi knjigo, ki je v tistem trenutku na vrhu kupa:

- Anja si je izposodila knjigo Peter Nos, na vrhu ostane knjiga Kolesar naj bo!.
- Feliks si je izposodil knjigo Kolesar naj bo!, na vrhu kupa je knjiga Babica v supergah.
- Anja je vrnila knjigo Peter Nos, ki jo je knjižničar dal na vrh kupa.
- Martin si je izposodil knjigo Peter Nos, na vrhu kupa je Babica v supergah.
- Feliks je vrnil knjigo Kolesar naj bo!, ki je zdaj na vrhu kupa.
- Katja si izposodi knjigo Kolesar naj bo!.

Računalniško ozadje

Kup knjig predstavlja sklad. Za sklad je značilno, da je podatek, ki je bil zadnji shranjen na sklad, edini, ki je takoj dostopen, ostali podatki so dostopni, šele ko so novejši odstranjeni s sklada.

Igralci pravljичne predstave prihajajo in zapuščajo oder po vrstnem redu, prikazanem na sliki (od zgoraj navzdol). Predstava ima dve dejanji in med njima en odmor.

Katera izjava **ni resnična**?

- A) Princ in princesa sta bila skupaj na odru.
- B) Kralj in zmaj sta bila skupaj na odru.
- C) Princ je prišel na oder po odmoru.
- D) Princ in zmaj sta bila skupaj na odru.

Rešitev

Pravilni odgovor je B.

A) Izjava »Princ in princesa sta bila skupaj na odru« je pravilna, saj je po odmoru na oder prišel princ in zapustil zmaj ter nato je na oder zopet prišla princesa.

B) Izjava »Kralj in zmaj sta bila skupaj na odru« **ni pravilna**, zato ker je kralj zapustil oder, ravno preden je zmaj prišel na oder.

C) Izjava »Princ je prišel na oder po odmoru« je pravilna.

D) Izjava »Princ in zmaj sta bila skupaj na odru« je pravilna, saj je po odmoru na oder prišel zmaj in nato še princ.

Računalniško ozadje

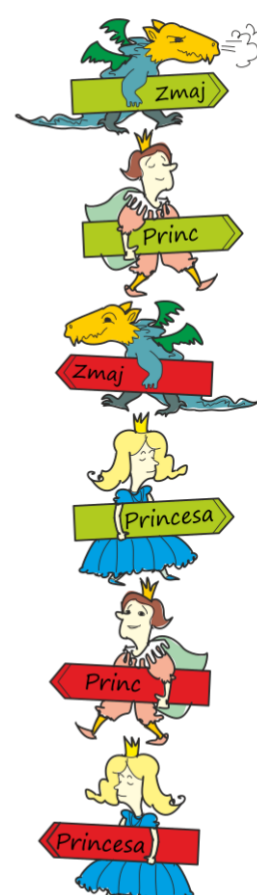
V informatiki pogosto grafično ponazorimo procese (dogodke, ki se zgodijo sčasoma v določenem vrstnem redu). Grafični vizualni prikazi informacij, podatkov ali znanja so običajno namenjeni hitri in jasni predstavitvi informacij. Te lahko pripomorejo k razvijanju sposobnosti prepoznavanja vzorcev. Razumevanje vizualnih predstav (ali diagramov) in sklepanje iz njih je pomembna veščina v računalništvu.

Začetek 1. dejanja



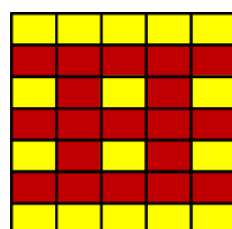
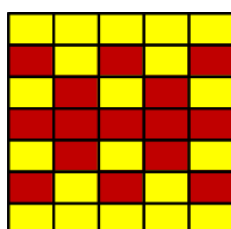
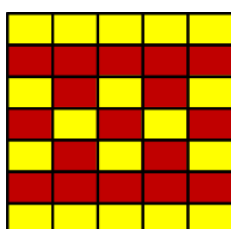
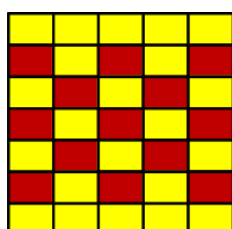
Odmor

Začetek 2. dejanja



Konec

Lini so vseč rdeče-rumene vezenine, ki jih lahko opiše s kodo iz zaporedja števil. Številke opisujejo vzorec od spodaj navzgor, od leve proti desni. Vzorec se ponavlja. Poveži kodo z vzorcem, ki ga opisuje.



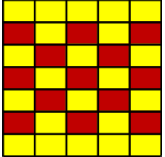
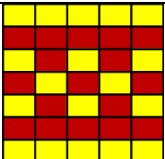
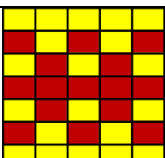
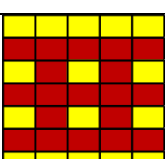
1.1.1.1.1.1.1
2.3.2

1.1.1.1.1.1.1
1.5.1

1.1.1.1.1.1.1
1.2.1.2.1

1.1.1.1.1.1.1
2.1.1.1.2

Rešitev

	1.1.1.1.1.1.1 2.1.1.1.2
	1.1.1.1.1.1.1 1.2.1.2.1
	1.1.1.1.1.1.1 2.3.2
	1.1.1.1.1.1.1 1.5.1

Če pogledamo vzorec spodnjega levega kota navzgor, opazimo, da se v vseh vezeninah v prvi, tretji in peti vrstici pojavlja izmenjevanje enega rumenega in enega rdečega kvadratka, zato je prva vrstica kode v vseh štirih vzorcih enaka 1.1.1.1.1.1.1. Kode se razlikujejo le v drugi vrstici.

2.1.1.1.2: 2 rumena kvadratka, nato 1 rdeč, 1 rumen, 1 rdeč in 2 rumena kvadratka v drugem in četrtem stolpcu.

1.2.1.2.1 pomeni 1 rumen, 2 rdeča, 1 rumen, 2 rdeča in 1 rumen kvadrata v drugem in četrtem stolpcu.

2.3.2 pomeni, da sta 2 rumena, 3 rdeči in 2 rumena kvadratka v drugem in četrtem stolpcu.

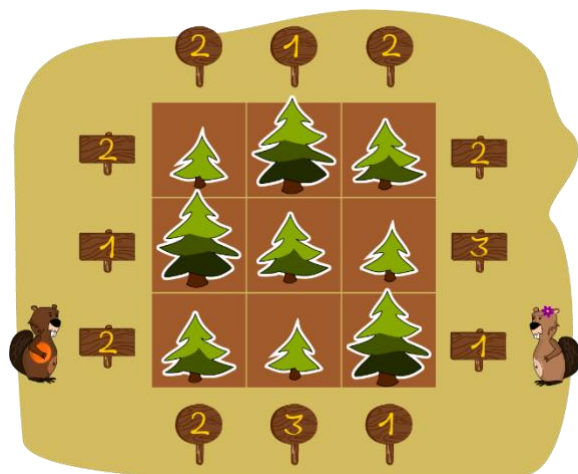
1.5.1 pomeni, da je 1 rumen, 5 rdečih in 1 rumen kvadrata v drugem in četrtem stolpcu.

Računalniško ozadje

Nekatere datoteke s slikami so shranjene na podoben način, zato da bi bile čim manjše.

Drevesni sudoku

5. - 7. razred



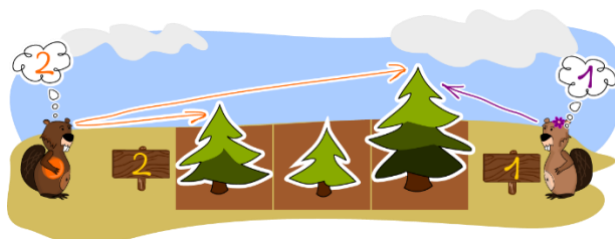
Bobri na polju vedno posadijo po 9 dreves skupaj, tako da so drevesa v 3 vrstah s po 3 drevesi. Imajo različne velikosti dreves: 1 (🌲),

2 (🌲) in 3 (🌲). V vsaki vrsti in v vsakem stolpcu je natanko po eno drevo vsake velikosti (to pomeni, da v nobeni vrsti in v nobenem stolpcu nista dve drevesi enake velikosti).

Okoli polja pa nato za vsako pozicijo (te so prikazane na levi sliki) pogledajo, koliko dreves se vidi s te pozicije, ter število zapišejo na

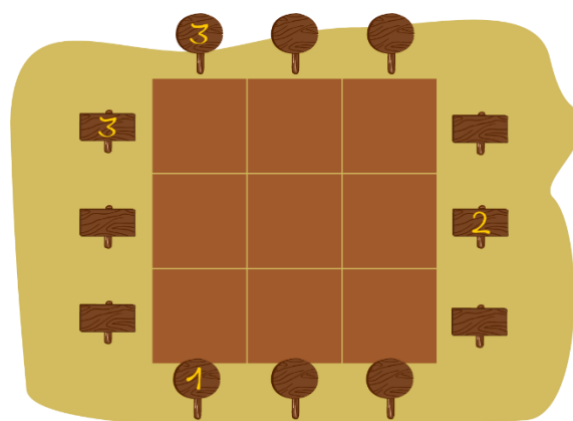
oznako (oznake 🌲 uporabijo za stolpce, oznake 🌲 pa za vrstice).

Ko bobri gledajo drevesa v ravni črti (kot kaže spodnja slika), ne vidijo manjših dreves, ki so skrita za večjim drevesom.



Če so bobri svoja opazovanja zapisali, kot kaže desna slika, katera drevesa so posajena v spodnji (tretji) vrstici polja (gledano od leve proti desni)?

- A. 1 (🌲), 2 (🌲) in 3 (🌲)
- B. 3 (🌲), 1 (🌲) in 2 (🌲)
- C. 3 (🌲), 3 (🌲) in 1 (🌲)
- D. 3 (🌲), 2 (🌲) in 1 (🌲)



Rešitev

Pravilni odgovor je 3 (🌲), 1 (🌲) in 2 (🌲).

Za podano polje je ena sama rešitev. Obe na oznake zapisani trojki zadoščata, da nedvoumno rekonstruiramo posaditev dreves na celotnem polju. Če biber z neke pozicije lahko vidi vsa tri drevesa, morajo biti ta drevesa urejena po vrsti od najmanjšega do

največjega (🌲🌲🌲). Torej lahko v polje nedvoumno postavimo prvo vrstico in prvi stolpec.



Ostala drevesa pa postavimo glede na pravila igre sudoku. Ker imamo lahko v vsaki vrstici in v vsakem stolpcu le po eno drevo enake velikosti, obstaja le en način, kako zapolniti preostala mesta. Na skrajno desnem mestu v srednji vrsti mora biti drevo velikosti 1 🌲, saj

je drevo velikosti 2 🌲 že na prvem mestu v tej vrsti, drevo velikosti 3 🌲 pa je v tretjem

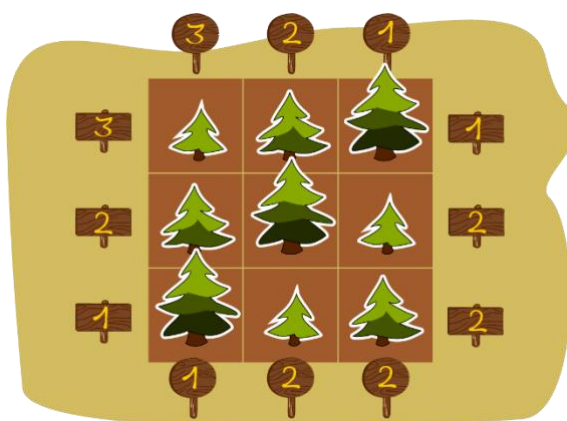
stolpcu. Tako imamo v tretjem stolpcu že drevesi velikosti 1 🌲 in 3 🌲, zato je v tretjem stolpcu spodaj lahko le še drevo velikosti 2 🌲. Podobno lahko sklepamo, da je v sredini

srednje vrstice lahko le drevo velikosti 3 🌲 (v tej vrstici že imamo drevesi velikosti 1 🌲

in 2 🌲). Na edinem še praznem mestu pa mora biti drevo velikosti 1 🌲, saj sta v tretji

vrstici že drevo velikosti 3 🌲 (v prvem

stolpcu) in drevo velikosti 2 🌲 (v tretjem stolpcu). Končna rešitev je na desni sliki.



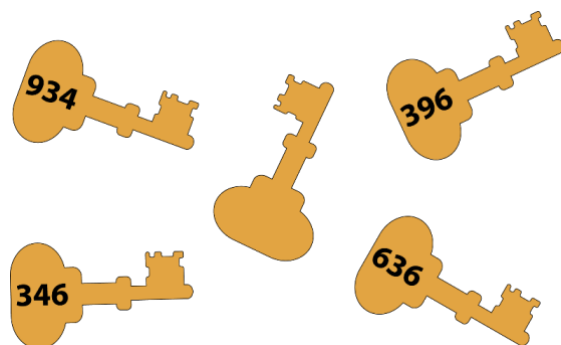
Računalniško ozadje

Poiskati smo morali rešitev problema, ki zadošča podanim omejitvam. Poleg tega pa je naloga od nas zahtevala sposobnost, da rekonstruiramo objekt na podlagi delnih informacij tako, da uporabimo znanje glede lastnosti tega objekta. Slednje lahko s pridom uporabimo pri stisnjeni predstavitvi objekta. Ta proces imenujemo brezizgubno stiskanje (angl. *lossless compression*), saj kljub stisnjeni obliki podatkov ne izgubimo nobene informacije, ko podatke ponovno razširimo.



Jaka je vratar bloka s petimi stanovanji. V vsakem stanovanju živi po en bober. Ko bobri odidejo v službo, ključke predajo Jaku. Da jih ne bi pomešal, hrani vsak ključ v svoji skrinjici. Na vsaki skrinjici so zapisane prve tri črke imena lastnika ključa.

Zaradi varnosti Jaka na ključke ne zapisuje imen, ampak trimestna števila, s katerimi označi lastnika. Posamezna črka je na vseh ključih predstavljena z isto številko.



Nekega dne so mu ključki padli na tla. Na štirih so nalepke 934, 346, 396 in 636, nalepka enega od njih pa se je izgubila. Kaj piše na manjkajoči nalepki?

Rešitev

Skrinjica BEB je edina, pri kateri se prva in zadnja črka ujemata. Kar pomeni, da skrinjico odpira ključ 636. Sedaj vemo, da črko B zapišemo s številko 6 in črko E s številko 3. Skrinjica AER je edina, ki se konča na črko R. Iz tega sledi, da črko R zapišemo s številko 4 in da ključ 934 odpre skrinjico AER. Na podlagi tega, črka A ustreza številki 9.

Sedaj, ko poznamo vse številke, brez težav ugotovimo, da ključ 396 odpira skrinjico EAB in ključ 346 skrinjico ERB. Ostane nam še skrinjica RAB, ki jo odpre ključ 496, katerega oznaka tudi manjka.

Računalniško ozadje

Šifriranje in dešifriranje sta temeljna pojma kriptografije. Tu smo vsaki številki dodelili neko črko. Vsako črko smo uporabili le enkrat. Pri kodiranju posamezno črko nadomestimo z ustrežno številko, pri dekodiranju prejetega sporočila pa številko zamenjamo z ustrežno črko. Tak način šifriranja ni posebno varen, saj je takšno kodo prelahko uganiti.



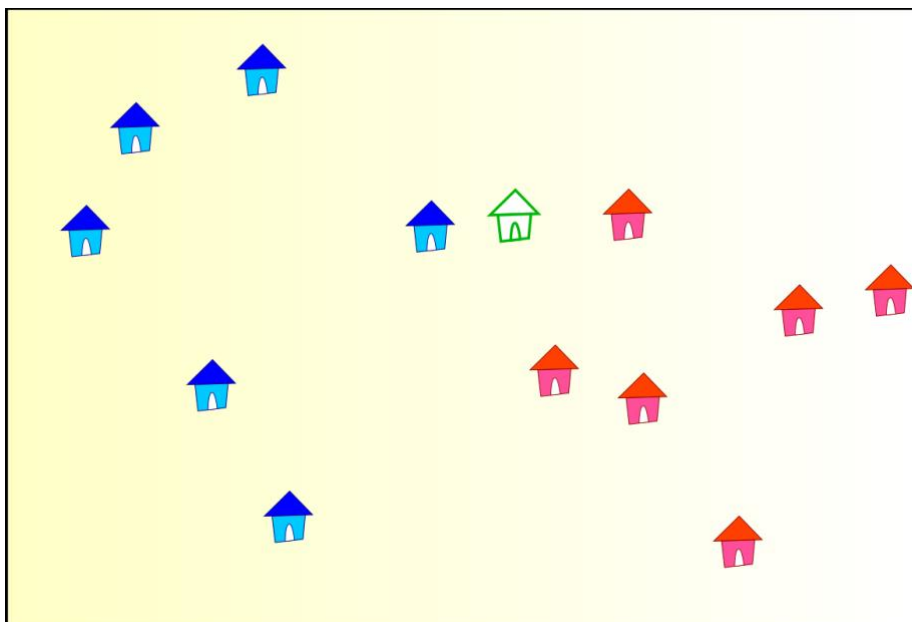
V vasi Bobrovnik so zgradili novo hišo. V vasi velja naslednje pravilo, po katerem določijo, katere barve mora biti nova hiša:

Barva nove hiše mora biti enaka barvi večine k najbližjih hiš. Če je rezultat neodločen, povečajo k za 1 in postopek ponovijo.

Začetne vrednosti števila k ne poznamo.

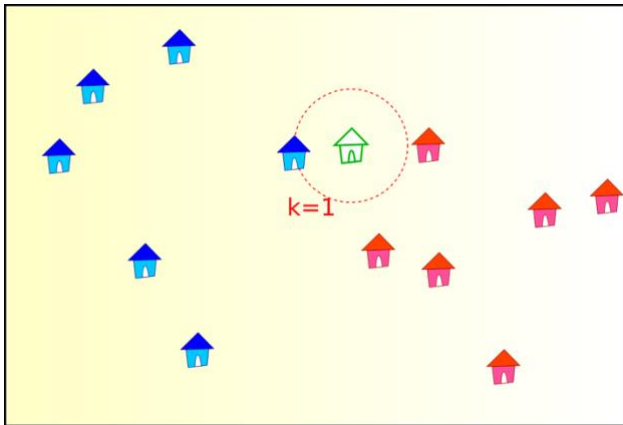
Spodaj je zemljevid Bobrovnika. Nova hiša je na zemljevid vrisana z zeleno barvo. Kot vidimo, so vse hiše levo od nove modre, vse desno od nje pa rdeče barve.

Pravijo, da bo barva nove hiše rdeča. Kakšna je najmanjša začetna vrednost k , pri kateri je to res?

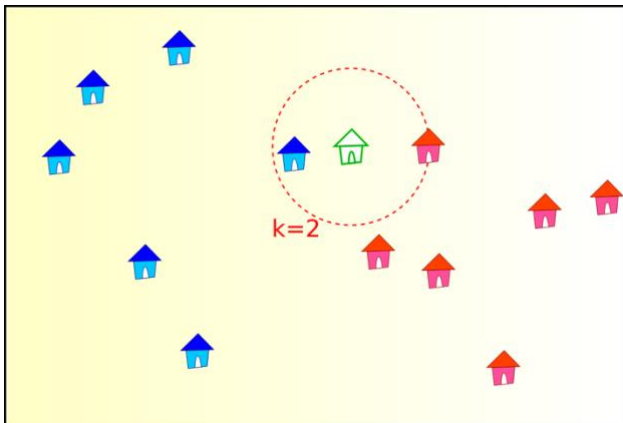


Rešitev

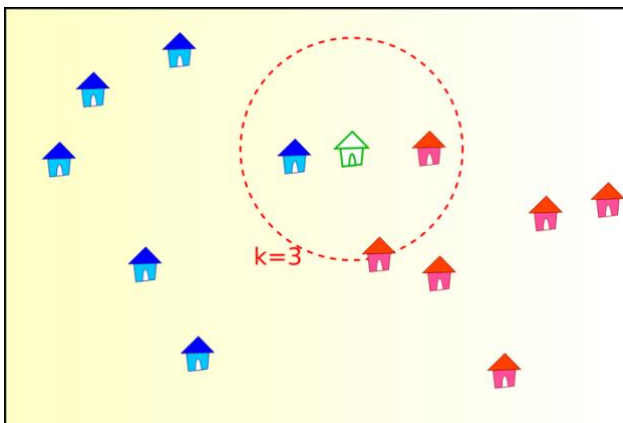
Vrednost k mora biti najmanj dva.



Če je $k = 1$, bo nova hiša modra, tako kot njej najbližja hiša.



Če je $k=2$, potem sta najbližji hiši ena modra in ena rdeča. Ker je rezultat neodločen, povečamo k za ena. Torej je $k=3$ in pogledamo še tretjo najbližjo hišo, ki pa je rdeča. Torej je večina hiš rdeča. Zato je pravilen odgovor dva.



Če je $k=3$, potem sta najbližje hiše ena modra in dve rdeči, zato bo hiša rdeča. A ker je že pri $k=2$ enak rezultat, je pravilni odgovor 2.

Računalniško ozadje

Algoritem k najbližjih sosedov je eden od najstarejših postopkov iz strojnega učenja, veje umetne inteligence. Uporablja se za napovedovanje različnih izidov, diagnosticiranje bolezni in podobno. Deluje tako, da v množici zbranih podatkov z znanim izidom poišče nekaj najpodobnejših in napove izid, ki se med njimi pojavi največkrat.

Polaganje ploščic

6. - 9. razred



Bober Bitaro polaga osem tipov ploščic (slika desno). Polaga jih od leve proti desni upoštevajoč naslednja pravila:

- prva ploščica je vedno tipa 1, druga ploščica pa tipa 2.
- naslednjo ploščico določi tako, da sešteje številki tipa prejšnjih dveh ploščic. Če je vsota več kot 7, uporabi sivo ploščico tipa -2.

Primer: tretja ploščica bo rdeča tipa 3, saj sta prejšnji dve ploščici tipov 1 in 2.



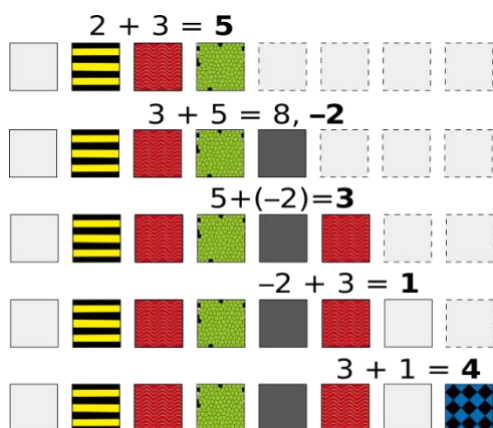
	1
	2
	3
	4
	5
	6
	7
	-2

Kateri tip ploščic bo Bitaro položil na osmo mesto?

Rešitev

Bitaro bo na osmo mesto položil ploščico tipa 4.

Na sliki lahko vidite postopek reševanja:



Računalniško ozadje

Pravilo, ki ga Bitaro uporablja za postavljanje ploščic, je algoritem, ki ga uporabljamo za izračun Fibonaccijevega zaporedja, s to razliko, da v primeru, ko je število večje od sedem, tega nadomestimo z -2.

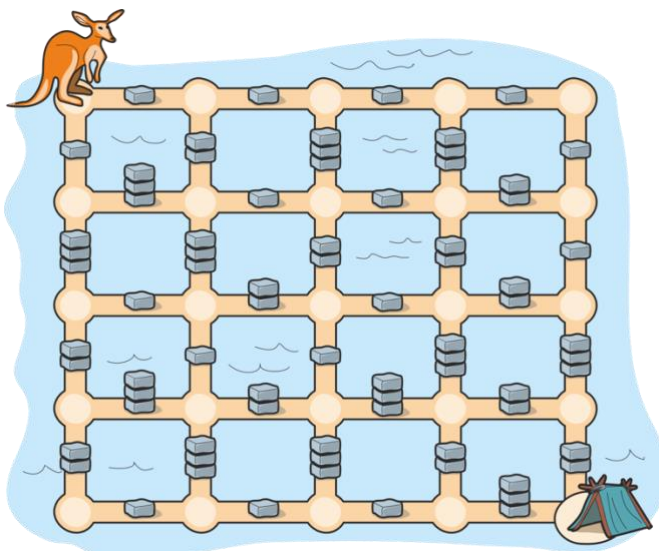
Poskočni kenguru

6. - 9. razred



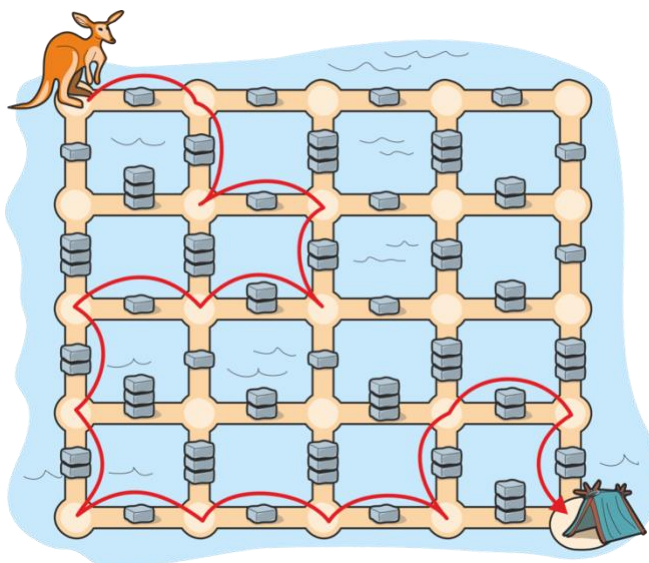
Kenguru skače domov. Skače lahko le po poti, to je vodoravno (levo – desno) ali navpično (gor – dol). Na poti lahko preskoči oviro visoko do dve opeki. Ovire s tremi opekami so zanj še previsoke.

Kolikokrat mora skočiti, da bo domov prišel po čim krajši poti?



Rešitev

Kenguru mora skočiti 14-krat, kot kaže slika.

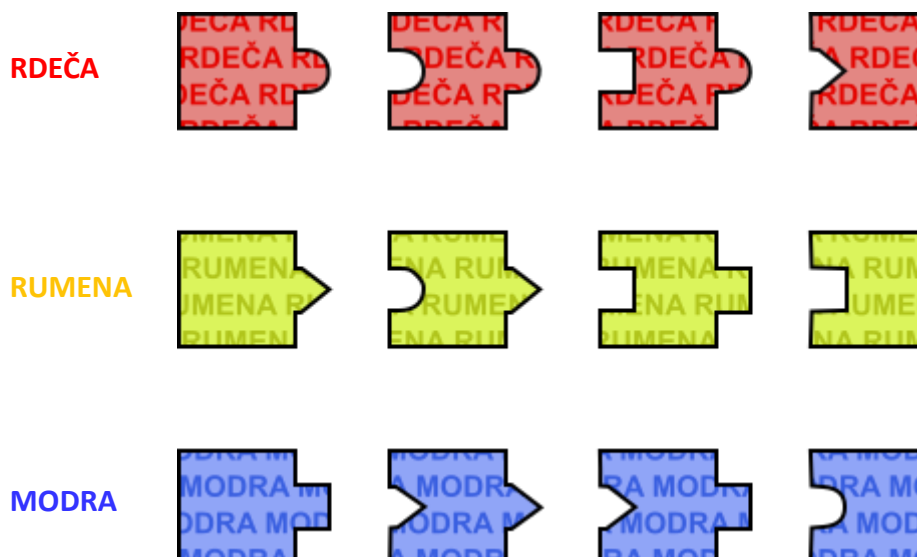


Računalniško ozadje

Iskanje najkrajše poti je eden izmed znanih problemov, ki jih rešujemo v računalništvu, še posebej pogosto pa nanj naletimo na tekmovanju Bober.

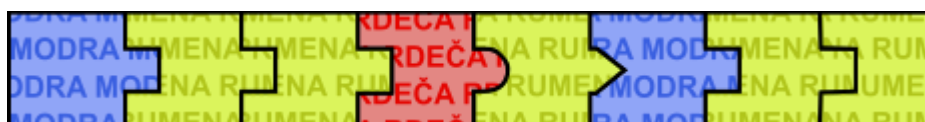


Bober David ima neomejeno število koščkov sestavljanke. Sestavlja lahko 12 različnih oblik koščkov, ki so v treh različnih barvah.



S temi delčki lahko sestavi različna barvna zaporedja, na primer:

MODRA → **RUMENA** → **RUMENA** → **RDEČA** → **RUMENA** → **MODRA** → **RUMENA** → **RUMENA**



Zaporedje se mora začeti s koščkom, ki ima raven levi rob, in zaključiti s koščkom, ki ima raven desni rob. Vmesni delčki se morajo ujemati.

Katerega od naslednjih barvnih zaporedij **ne more** sestaviti s svojim naborom delčkov?

- A) **RUMENA** → **MODRA** → **MODRA** → **RDEČA** → **RDEČA** → **RDEČA** → **MODRA**
- B) **MODRA** → **RUMENA** → **RDEČA** → **RUMENA** → **RDEČA**
- C) **RDEČA** → **RDEČA** → **RUMENA** → **MODRA** → **MODRA** → **MODRA**
- D) **MODRA** → **RDEČA** → **RUMENA** → **MODRA** → **RDEČA** → **RUMENA** → **RDEČA**

Rešitev

Pravilni odgovor je C, saj David ne more združiti nobenega modrega koščka z modrim zaključnim koščkom.

Računalniško ozadje

Naloga spominja na način, na katerega opišemo sintakso računalniškega jezika.



Maja in Miha si med poukom pošiljata skrivna sporočila. Ker ne želita, da bi kdo drug prebral vsebino njunih sporočil, sta si izmislila šifro, s katero bodo njuni pogovori ostali skrivni. Miha napiše črke sporočila v kvadrat po diagonali in jih prebere po vrsticah. Njegovo sporočilo KDAJSESPETDOBIVA se torej preoblikuje v SJDKDPSAIOEEAVBT.

K	D	A	J	S	E	S	P	E	T	D	O	B	I	V	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

S	J	D	K
D	P	S	A
I	O	E	E
A	V	B	T

S	J	D	K	D	P	S	A	I	O	E	E	A	V	B	T
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Maja mu odgovori s sporočilom ROBSVLEOOOIAHBTRČŠVTINAČS. Kdaj se bosta dobila?

Rešitev

Pravilen odgovor je v soboto ob treh. Sporočilo zapišemo v tabelo po vrsticah od leve proti desno, od zgoraj navzdol. Odgovor pa preberemo diagonalno, začnši v zgornjem desnem kotu (številke v levi tabeli povejo zaporedje črk v odgovoru).

11	7	4	2	1	R	O	B	S	V
16	12	8	5	3	L	E	O	O	O
20	17	13	9	6	I	A	H	B	T
23	21	18	14	10	R	Č	Š	V	T
25	24	22	19	15	I	N	A	Č	S

Maja in Miha se bosta dobila v soboto ob treh v slaščičarni.

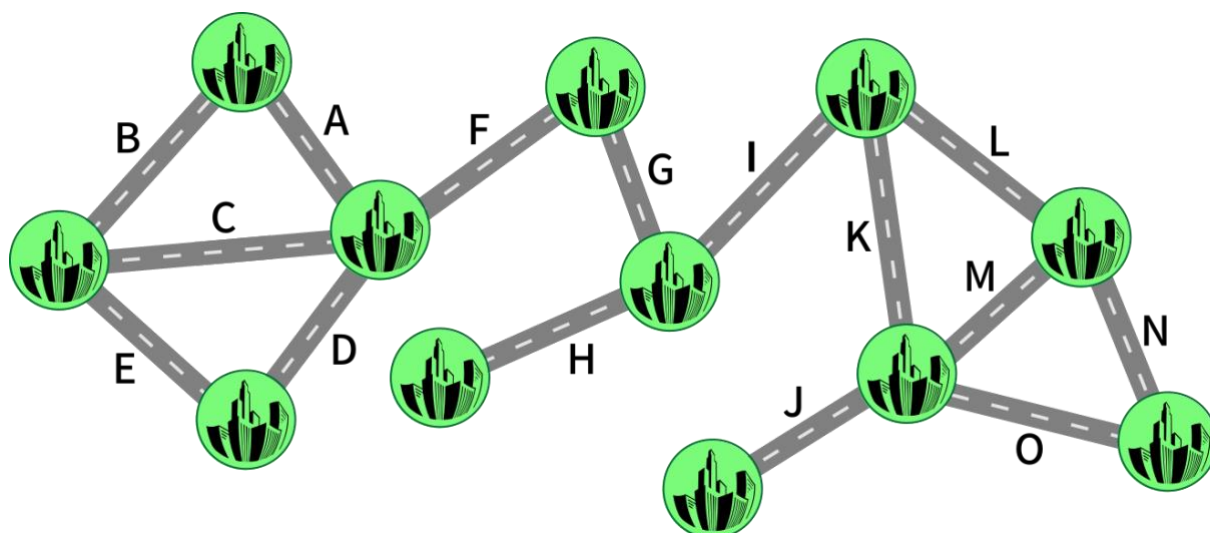
V	S	O	B	O	T	O	O	B	T	R	E	H	V	S	L	A	Š	Č	I	Č	A	R	N	I
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Računalniško ozadje

Kriptografija je veda, ki se ukvarja z zaščito podatkov s pomočjo šifriranja. Pomembna je tako v računalništvu kot tudi drugod. Uporablja se jo predvsem zaradi varnosti, saj se na takšen način lahko zaščitimo pred vsiljivci, ki želijo ukrasti naše podatke ali prestreči komunikacijo.



V državi je 12 mest, ki so med seboj povezana z avtocestami (označenimi s črkami A do O), kot kaže slika.



Mesta, ki so neposredno ali posredno povezana z eno ali več avtocestami, sestavljajo ekonomsko skupnost. Trenutno vseh 12 mest sestavlja eno skupnost.

Vlada se je zaradi izbruha epidemije odločila omejiti potovanja po državi tako, da zapre dve avtocesti. Njihov cilj je razdeliti državo na tri ločene ekonomske skupnosti. Da minimizirajo motnje v ekonomiji, mora po zaprtju avtocest najmanjša skupnost obsegati čim več mest.

Kateri dve avtocesti naj zaprejo?

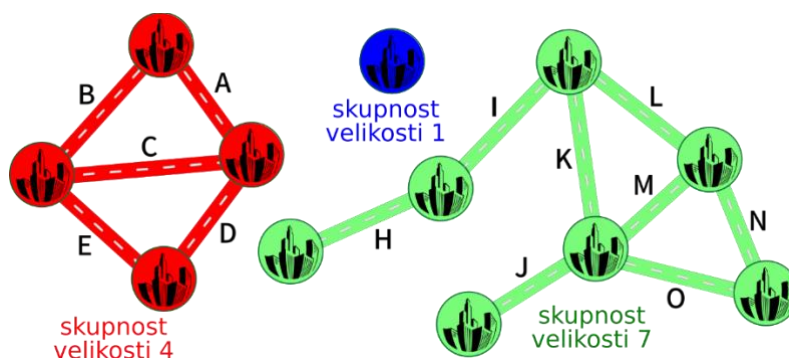
Rešitev

Zaprejo naj avtocesti F in I.

Ker je potrebno državo razdeliti na 3 dele, je smiselno upoštevati le avtoceste, ki po zaprtju razdelijo državo tako, da po drugi cesti ni možen dostop iz enega v drug del. Na primer, če zaprejo avtocesto C, lahko ljudje še vedno med mestoma potujejo preko avtocest A in B, zato te avtoceste ni smiselno zapreti. Edine ceste, ki po zaprtju razdelijo državo na ločene dele so ceste F, G, H, I in J, saj je sicer za ločitev skupnosti na ločene dele potrebno zapreti vsaj 2 cesti, mi pa moramo z zaprtjem le dveh avtocest ločiti državo na 3 skupnosti.

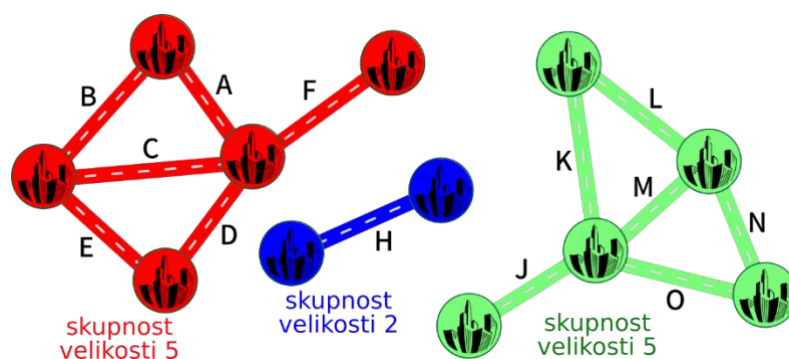
Zdaj je potrebno preveriti vse možne kombinacije zaprtja dveh cest in preveriti, koliko mest je povezanim v najmanjši skupnosti.

Če na primer zapremo avtocesti F in G, je najmanjša skupnost velikosti 1:

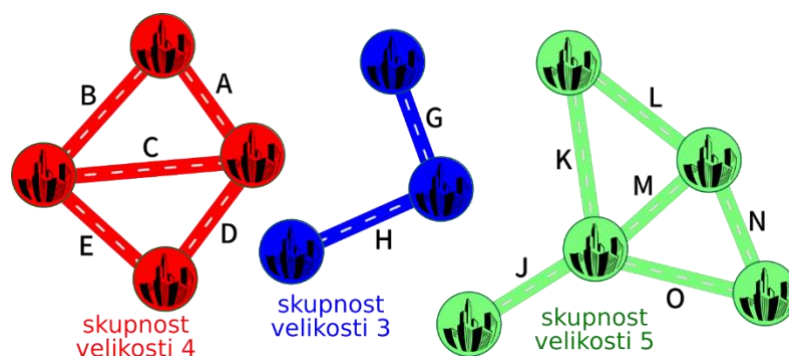


Skupnost velikosti 1 dobimo tudi če zapremo naslednje kombinacije avtocest: F in H, F in J, G in H, G in J, H in I, H in J ter I in J

Preverimo še kombinacijo zaprtja avtocest G in I. V tem primeru državo razdelimo na 3 dele, najmanjša skupnost je velikosti 2, v ostalih dveh pa je v vsaki po 5 mest.



Ostane še kombinacija zaprtja avtocest F in I. V tem primeru ostanejo skupnosti velikosti 3, 4 in 5. Ker je najmanjša skupnost v tem primeru 3, kar je več kot v vseh ostalih primerih, je to pravilna rešitev.



Računalniško ozadje

Naloga je povezana z uporabo grafov. Mesta predstavljajo vozlišča, avtoceste pa neusmerjene povezave med njimi. V nalogi iščemo mostove, to so povezave, ki povečajo število nepovezanih komponent v grafu, če jih odstranimo.



Stroj za razpoznavo lahko razpozna naslednjih pet slik, ki predstavljajo črke I, T, O, C in L, po vrsti.



Pri razpoznavanju črk stroj uporablja toplotne karte. Na toplotni karti slike je za vsak piksel označena redkost danega piksla na dani poziciji. Svetlejša kot je barva, bolj redek je ta piksel:

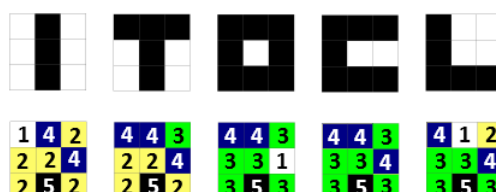
- 1** Edini primerek. Nobena druga slika nima na tem mestu piksla enake barve.
- 2** Redek. Le ena druga slika ima na tem mestu piksel enake barve.
- 3** Pogost. Dve drugi sliki imata na tem mestu piksel enake barve.
- 4** Zelo pogost. Tri druge slike imajo na tem mestu piksel enake barve.
- 5** Običajen. Vse ostale slike imajo na tem mestu piksel enake barve.

Tako ima slika črke  toplotno karto . Katera slika pa ima toplotno karto .

Rešitev

Pravilen odgovor je B, slika črke O: . Podana toplotna karta ima bel piksel na 3. poziciji v drugi vrsti. To pomeni, da mora biti piksel na tej poziciji na pripadajoči sliki unikatni.

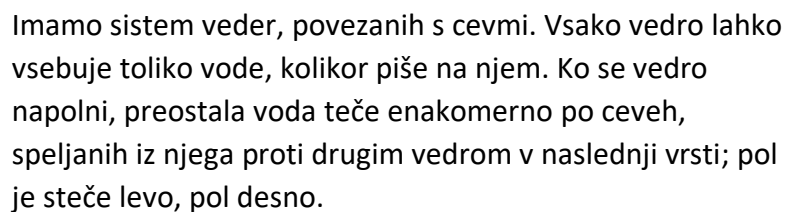
Slika  je edina, ki ima na tej poziciji piksel črne barve. Torej je to tudi edina slika, ki ima na tej poziciji edini primerek takega piksla. Toplotne karte ostalih črk prikazuje spodnja slika.



Računalniško ozadje

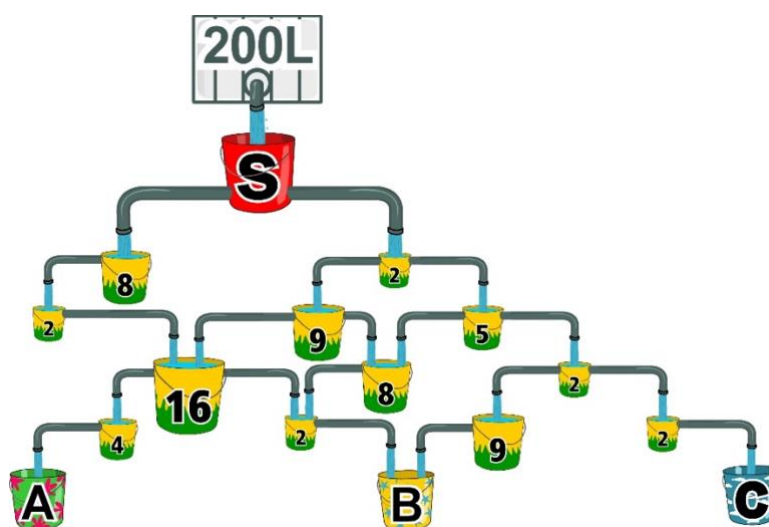
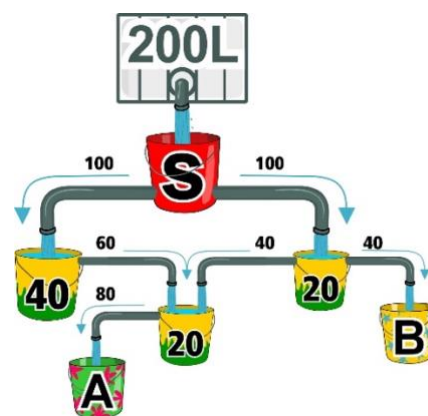
Toplotna karta (angl. *heat map*) je grafična predstavitev podatkov, kjer so individualne vrednosti iz matrike predstavljene z barvami. Verjetno ste toplotno karto že videli v vremenski napovedi pri prikazu temperatur na različnih območjih. Toplotne karte (kot je ta v naši nalogi) se uporabljajo tudi pri razpoznavanju slik. Ideja je, da vsakemu pikslu pripišemo »zaznavno pomembnost« in tako se globoke učne mreže lahko osredotočijo le na določene dele slike.

6. - 9. razred, srednja šola



Rdeče vedro z oznako S ne drži vode. Vanj zlijemo 200 litrov vode, ki steče po sistemu navzdol. V sistemu na desni sliki bi se 40 litrov vode nabralo v vedru B.

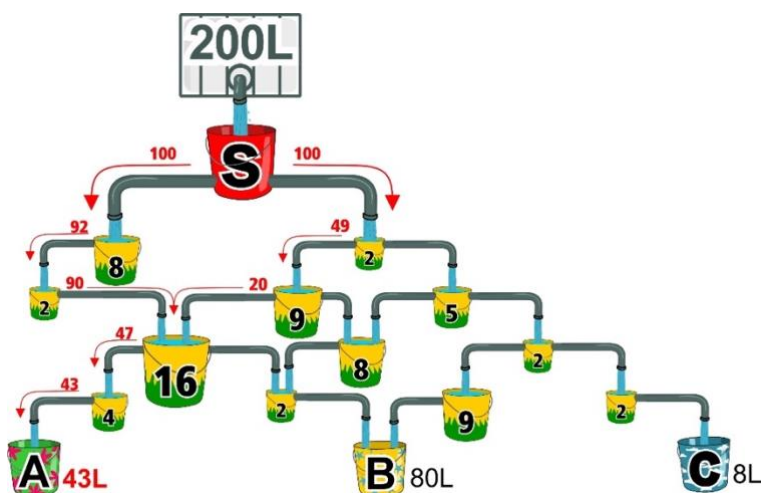
Koliko vode se nabere v vedru A v spodnjem sistemu?



V vedru A se nabere 43 l vode.
Ker nas zanima le vedro A,
zadošča že, da izračunamo
količine vode na levi strani
sistema (to je 8 števk).

Računalniško ozadje

Naloga nekoliko spominja na sestavljanje optimalna iskalna drevesa, podatkovne strukture, namenjene hitremu iskanju podatkov.





Plavajočo vas na jezeru Titicaca sestavlja 7 plavajočih otokov. Na 6 otokih so hiše, v katerih lahko živi ena ali pa tudi več oseb. Na sedmem otoku pa je pošta, kamor je na novo delovno mesto ravno prispel novi poštar, ki je zadolžen za raznašanje pošte znotraj te vasi. Poštar je na začetku nekoliko zmeden, saj ne pozna nobenega prebivalca vasi. Zato si zamisli naslednjo strategijo za dostavo pisem:



1. Za vsako pismo si v beležko zapiše ime in naslov pošiljatelja.
2. Nato pogleda v svojo beležko in če je v njej zapisano ime naslovnika, dostavi pošto naslovniku.
3. Če pa naslovnikovega imena ni v njegovi beležki, naredi kopije tega pisma in jih dostavi na vse naslove v vasi, razen na naslov pošiljatelja.
4. V vsakem primeru, pravi naslovnik vedno odgovori na prejeto pismo še isti dan. Tako si poštar zapiše njegovo ime in naslov ter dostavi pismo z odgovorom.

Prvi dan pošlje Eva pismo Greti in Miha pošlje pismo Evi. Drugi dan Sandi pošlje pismo Neži.

Koliko pisem je poštar razdelil v prvih dveh dneh svoje službe?

Rešitev

Poštar v prvih dveh dneh razdeli skupaj 14 pisem.

Najprej Eva pošlje pismo Greti. Ker poštar ne pozna Gretinega naslova, naredi kopije pisma in jih dostavi na vseh 5 naslovov (šesti naslov je pošiljateljičin), torej dostavi 5 pisem. Nato Greta odgovori na Evin pismo in poštar dostavi ta odgovor (še 1 pismo). Nato Miha pošlje pismo Evi. Ker poštar že ve, kje živi Eva, dostavi pismo na njen naslov (še 1 pismo). Eva odgovori še isti dan in poštar dostavi pismo Mihi, saj ima njegov naslov že zapisan v svoji beležki (še 1 pismo). Torej je prvi dan dostavil skupaj 8 pisem.

Drugi dan pa Sandi pošlje pismo Neži. Ker poštar ne pozna Nežinega naslova, naredi kopije pisma in jih dostavi na vseh 5 naslovov (še 5 pisem). Neža odgovori še isti dan in poštar dostavi to pismo na že poznani mu naslov (še 1 pismo). Drugi dan je tako dostavil 6 pisem, skupaj v prvih dveh dneh pa 14 pisem.

Računalniško ozadje

Stikala – vsaj eno imate skoraj gotovo tudi doma – v računalniških omrežjih si na podoben način zabeležijo, v katero vtičnico je priključena katera naprava. Ta podatek potrebujejo zato, da lahko prispele podatke pošljejo pravemu naslovniku v omrežju.

Bobi se igra z nalepkami. Ima zelo veliko nalepk, a le štiri različne vrste: s črkami ABBA, GAGA, GIBB in IGGY. Z njimi sestavi besedo tako, da na prazen list papirja nalepi nalepke. Če nalepko nalepi na določeno mesto, ta prekrije štiri črke od tega mesta naprej.

Tako na primer lahko sestavi besedo GIABIGGYGA (* pomeni še prazno mesto):

1. GIBB nalepi na mesto 1: GIBB*****

2. ABBA nalepi na mesto 3: GIABBA****

3. GAGA na mesto 7: GIABBAGAGA

4. IGGY na mesto 5: GIABIGGYGA

Katere izmed naštetih besed Bobi ne more sestaviti s svojimi nalepkami?

- A. AGGIBBAGGAGABABGA
- B. AGGIBBAGAGGABABGA
- C. AGIBBGAGAGGYBAYBB
- D. AGGIBBAGAGGYBAGGY



Rešitev

Pravilna rešitev je B. Rešitev bomo lažje poiskali, če bomo začeli od zadaj, torej od podane besede odstranjevali nalepke, dokler ne pridemo do praznega lista. Na vsakem koraku poiščemo nalepko, ki je bila uporabljena (tj. podniz trenutne besede, ki ustreza besedi na nalepki), ter jo odstranimo s papirja. Ko nalepko odstranimo, nas ne zanimajo dejanske črke, ki so se skrivale pod to nalepko, saj so bile v končni besedi tako ali tako prekrite. Torej v naslednjih korakih upoštevamo uporabo nalepk, ki se prekrivajo s katerokoli predhodno uporabljeno nalepko (torej imajo črke na tem prekrivajočem se delu).

Na tak način lahko sledimo v obratni smeri za vse štiri podane besede (trenutna nalepka je zapisana krepko in podčrtano, po odstranitvi nalepke pa na mesto črke zapišemo zvezdico):

A.
AGGIBBAGGAGABABGA
AG****AG****BABGA
AGAGAGAGA*ABBABABGA
A*****BGA
ABBA*****GIBBGA
*****GA
*****GAGA

Prišli smo do praznega papirja!

B.
AGGIBBAGAGGABABGA
AG****AGAGGABABGA
AGAGAGAGAAGGABABGA
A*****GGABABGA
ABBA*****GGABABGA
*****GGABABGA

Nimamo več nalepk, torej rešitev ne obstaja!

C.
AGIBBGAGAGGYBAYBB
A*****GGYBAYBB
ABBA*****IGGYBAYBB
*****BAYBB
*****ABBAYBB
*****YBB
*****IGGYBB
*****BB
*****GIBB

Prišli smo do praznega papirja!

D.
AGGIBBAGAGGYBAGGY
AG****AGAGGYBAGGY
AGIBBGAGAGGYBAGGY
A*****GGYBAGGY
ABBA*****IGGYBAGGY
*****BAGGY
*****ABBAGGY
*****GGY
*****IGGY

Prišli smo do praznega papirja!

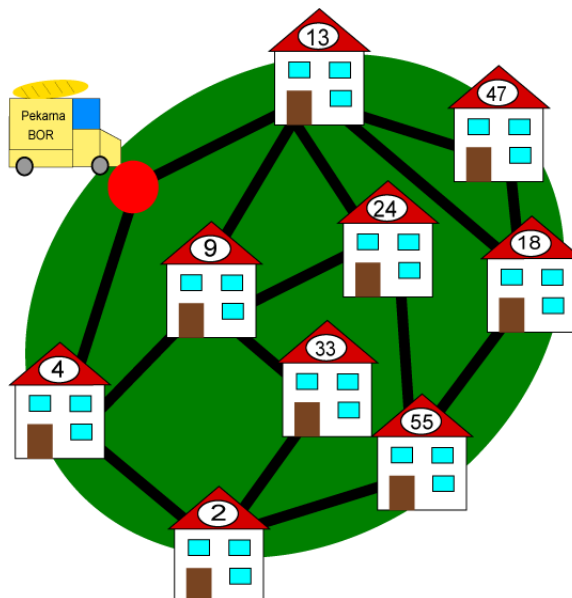
Računalniško ozadje

Naloga je primer problema, ki ga lažje analiziramo z uporabo sledenja nazaj, ki je v računalništvu pogosto uporabljena metoda reševanja problemov. Sledenje nazaj je proces sklepanja, obrnjen v času: začnemo pri končnem stanju problema (v naši nalogi pri sestavljeni besedi) in poskušamo ugotoviti akcijo, ki je pripeljala do tega stanja (v naši nalogi uporaba določene nalepke), ter s tem določiti možno predhodno stanje. To ponavljamo, dokler ne pridemo do začetnega stanja problema (v naši nalogi do praznega lista). Veliko problemov je takih, da imamo v tem procesu več možnosti za izbiro ustrezne akcije in v takem primeru izberemo eno od ustreznih akcij in če nas ta ne pripelje do cilja, postopek ponovimo z drugo možno akcijo.

Pekarna Bor dostavlja kruh na dom. Pri tem želi prevoziti kar najmanj kilometrov. Zato Bor pred dostavo vedno pregleda zemljevid z označenimi hišami strank in pripravi načrt poti, ki ga vodi do hiš vseh strank, ne da bi katero od ulic prevozil večkrat. Pot mora začeti in tudi končati v pekarni, ki je na zemljevidu označena z rdečim krogom.

V kakšnem vrstnem redu bo Bor obiskal hiše pri dostavi kruha?

- A. 13-47-18-55-24-9-33-2-4
- B. 4-9-33-2-55-18-47-13-9-4
- C. 13-9-24-47-18-55-2-33-4
- D. 4-2-55-18-47-13-24-9-33



Rešitev

Pravilen odgovor je A: 13-47-18-55-24-9-33-2-4.

Pri odgovoru B (4-9-33-2-55-18-47-13-9-4) izpusti obisk hiše 24, hiši 9 in 4 pa obiše dvakrat, kar ne ustreza zapisanim zahtevam poti.

Pot pri odgovoru C (13-9-24-47-18-55-2-33-4) ni možna, saj hiši 24 in 47 (ter tudi 33 in 4) nista povezani z ulico, zato ju ne more zaporedoma obiskati.

Pri odgovoru D (4-2-55-18-47-13-24-9-33) pa se pot konča pri hiši 33, od katere nobena ulica ne vodi neposredno do pekarni, kamor se mora Bor vrniti po opravljeni dostavi.

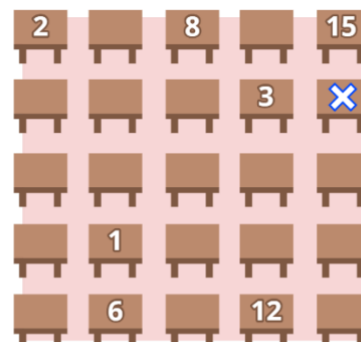
Računalniško ozadje

V računalništvu pogosto srečamo probleme, ki so povezani s teorijo grafov, saj se grafe pogosto uporablja za predstavitev podatkov. V naši nalogi hiše (in pekarna) predstavljajo vozlišča v grafu, ulice pa povezave med vozlišči. V nalogi smo v grafu iskali Hamiltonov cikel, to je pot, ki obiše vsa vozlišča natanko enkrat, začne in konča pa se v istem vozlišču. Takšen cikel ne obstaja vedno: Boru se utegne zgoditi, da se bo moral kakšen dan tudi večkrat peljati po isti ulici. Poleg tega iskanje Hamiltonovega cikla ni vedno preprosta naloga.



Luka se je prepozno spravil k učenju za prihajajoče kratko preverjanje znanja pri matematiki, zato je sedaj v velikih težavah. Na pomoč so mu priskočili njegovi sošolci, s katerimi se je dogovoril, da mu bodo med testom posredovali rešitve.

Sedem njegovih sošolcev, ki obvladajo matematiko, sedi v klopih, kot prikazuje slika (klopi s števkami označujejo klopi sošolcev, v klopi X pa sedi Luka).



Vsak od sošolcev potrebuje določen čas, da reši nalogo, za katero se je z Luko dogovoril, da mu posreduje rešitev. Ta čas je zapisan na njegovi klopi, kot je npr. **5**.

Luka ima en sam list papirja, na katerega bodo sošolci zapisali svoje rešitve. Za podajo papirja med dvema sosednjima klopema (vodoravno ali navpično, podaja diagonalno ni možna) potrebujejo eno minuto.

Ko papir prispe do klopi, v kateri je učenec že našel rešitev, to pripiše na list papirja. Papir lahko da v obtok katerikoli sošolec (torej lahko začne krožiti iz katerikoli klopi), a mora svojo pot zaključiti v klopi, kjer sedi Luka, to je **X**, najkasneje po 16 minutah, ko se test zaključí.

Če bi vsi kar najhitreje poslali papir naprej, koliko rešitev bi lahko Luka dobil na koncu 16-minutnega testa?

Rešitev

Luka lahko dobi največ 5 rešitev.

Skupaj je 7 rešitev, vendar je nemogoče, da bi zbrali vseh sedem. Poglejmo le dve klopi, v katerih priprava rešitve traja najdlje (to sta 12 in 15). Če v klopi 12 poberemo odgovor po 12. minuti, ta odgovor potrebuje še 5 minut, da pride do klopi 15 (rešitve v klopi 15 ne moremo pobrati prej kot minuto pred koncem testa). Torej v roku 16 minut ne moremo pobrati obeh rešitev 12 in 15.

Tudi 6 rešitev ne moremo zbrati pravočasno. Ugotovili smo že, da ne moremo imeti obeh rešitev 12 in 15. A če želimo zbrati 6 rešitev, morajo te vključevati ali 12 ali 15. Vsako od njiju lahko spravimo pravočasno do klopi X, saj je klop 12 oddaljena 4 minute od X, klop 15 pa 1 minuto.

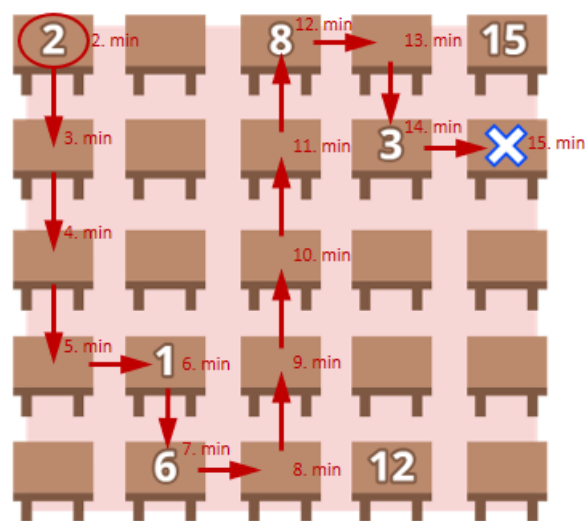
Recimo, da vključimo rešitev iz klopi 12. Naslednje največje število minut, ko moramo čakati na rešitev, je 8. Tudi če bi zbrali vse rešitve 1, 2, 3 in 6 v 8 minutah, bi vseeno porabili še

nadaljnjih 5 minut, da bi papir spravili do klopi 12. To pa je eno minuto več, kot si lahko privoščimo. Torej rešitve iz klopi 12 ne moremo vključiti pravočasno.

Pa pogledjmo še možnost, da vključimo rešitev iz klopi 15. Potem bi morali zbrati vse rešitve 1, 2, 3, 6 in 8 v 13 minutah, saj potrebujemo še dve dodatni minuti, da rešitve pridejo od klopi 8 (ali 3) do klopi 15. S slike lahko hitro ugotovimo, da čakanje blizu klopi 1, da pobereмо njeno rešitev, ni smiselno, saj rešitve iz klopi 6, ki je v njeni bližini, takrat še ne moremo dobiti, zato se bo moral papir še enkrat vrniti v ta del učilnice po rešitev 6. Podobno velja tudi za rešitev 3, saj bi čakanje na rešitev 8, ki je v bližini, bila izguba časa, prav tako pa tudi ponovno vračanje v ta del učilnice. Če torej nočemo predolgo čakati ali se ponovno vračati v že obiskane dele učilnice (kjer sta 1 in 6 oziroma 3 in 8), moramo začeti pri rešitvi 2. To lahko dobimo po dveh minutah. Če gremo nato po rešitvi 3 in 8 ter šele po tem pobereмо rešitvi 1 in 6, smo prekoračili čas, ki ga imamo na voljo (rešitev 8 lahko pobereмо šele po 8. minuti, potrebujemo pa še najmanj 5 minut, da pridemo od 8 do 6, ter še najmanj 7 minut, da pridemo od 6 do 15, kar je skupaj več kot razpoložljivih 15 minut). Torej moramo po rešitvi 2 takoj pobrati rešitvi 1 in 6 ter šele po tem rešitvi 3 in 8. Od klopi 2 potrebujemo 4 minute do klopi 1 (torej rešitev pobereмо v $2 + 4 = 6$ minutah). Dodatno minuto potrebujemo do klopi 6 (skupaj 7 minut). Dodatnih 5 minut potrebujemo do klopi 3 ali 8 (skupaj 12 minut) in še dve minuti, da obiščemo obe klopi (tj. 5 minut do 3 in 2 minuti do 8 ali pa 5 minut do 8 in 2 minuti do 3). Tako smo skupaj porabili 14 minut.

Od klopi 3 ali 8 do klopi 15 pa sta še 2 minuti, tako da skupaj porabimo 16 minut (celoten čas, ki je na voljo), list pa še ni prišel do klopi X. Torej ni možno, da zberemo 6 rešitev.

Lahko pa zberemo 5 rešitev. Ugotovili smo že, da lahko pridemo do klopi 3 v 14 minutah in pred tem zberemo rešitve po klopih 2, 1, 6 in 8. Klop 3 je od klopi X oddaljena le 1 minuto, tako da rešitev prispe pravočasno. Imamo več možnosti, kako zberemo 5 rešitev, ena med njimi je prikazana na sliki.



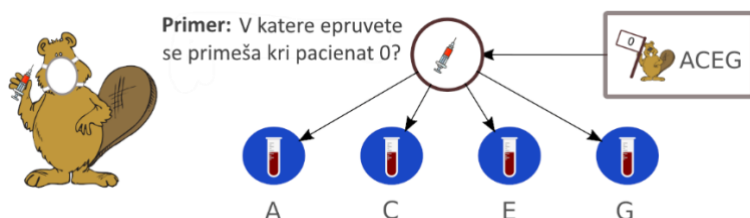
Računalniško ozadje

V nalogi je predstavljen problem optimizacije. V naši nalogi smo morali obiskati kar največ klopi, v katerih so bile pripravljene rešitve. Problem pa je imel še dodatno omejitev: čas, ki mora preteči, preden je rešitev na voljo. Problem lahko predstavimo z grafom, kjer so klopi vozlišča, povezave v grafu pa predstavljajo možen prehod papirja med klopi. Poiskati moramo tako pot v grafu, tj. zaporedje povezanih vozlišč, ki se zaključi v vozlišču X in vključuje kar največ s številkami označenih vozlišč. Pri tem mora zadostiti tudi vsem časovnim omejitvam.



Doktor Bober ve, da je natanko eden od njegovih 16 pacientov okužen z virusom. To lahko potrdi z laboratorijsko preiskavo krvi, a ima na voljo le 4 teste za ugotavljanje okužbe. Poleg tega zaradi pomanjkanja opreme nima na razpolago več kot 8 epruvet.

Dr. Bober vzame vzorec krvi vsem 16 pacientom ter vzorce premeša v 8 epruvet. Pred tem je epruvete označil s črkami A, B, C, D, E, F, G in H. Pozorno je tudi zabeležil, v katere epruvete je porazdelil krvni vzorec vsakega pacienta (zabeležka je na desni sliki). Tako je na primer kri pacienta 0 primešal v epruvete A, C, E in G.



Nato je vsebino treh epruvet testiral z uporabo laboratorijskega testa in ugotovil, da je vsebina epruvete A okužena, vsebini epruvet C in E pa nista okuženi.

Rezultati opravljenih testov:



epruveta A - okužena



epruveta C - ni okužena



epruveta E - ni okužena

Ostal mu je le še en test. Katero od preostalih štirih epruvet naj dr. Bober testira, da bo z gotovostjo lahko določil okuženega pacienta?

Razdelitev po epruvetah:

- 0 ACEG
- 1 ACEH
- 2 ACFG
- 3 ACFH
- 4 ADEG
- 5 ADEH
- 6 ADFG
- 7 ADFH
- 8 BCEG
- 9 BCEH
- 10 BCFG
- 11 BCFH
- 12 BDEG
- 13 BDEH
- 14 BDFG
- 15 BDFH

Rešitev

Testirati mora še epruveto G.

Pri razmisleku si pomagajmo s tabelo. V tabeli je z X označeno, če je kri danega pacienta v dani epruveti. Rezultati treh opravljenih testov so v tabeli označeni z barvami: svetlo zeleni so možni okuženi pacienti, ki so prispevali vzorce v epruveto A, svetlo modri so možni okuženi pacienti, ki niso prispevali vzorcev v epruveto C, ter zelenomodri so možni okuženi pacienti, ki niso prispevali vzorcev v epruveto E.

	A	B	C	D	E	F	G	H
0	X		X		X		X	
1	X		X		X			X
2	X		X			X	X	
3	X		X			X		X
4	X			X	X		X	
5	X			X	X			X
6	X			X		X	X	
7	X			X		X		X
8		X	X		X		X	
9		X	X		X			X
10		X	X			X	X	
11		X	X			X		X
12		X		X	X		X	
13		X		X	X			X
14		X		X		X	X	
15		X		X		X		X

Okužena A
 Neokužena C
 Neokužena E

}

Ta dva pacenta
ustrezata vsem
pogojem

Ker je kri v epruveti A okužena, mora biti okužen eden od pacientov 0 do 7, ki so prispevali kri v to epruveto. Ker epruveta C ni okužena, tudi pacienti 0 do 3, ki so dali kri v to epruveto, ne morejo biti okuženi. Torej je okuženi eden od pacientov 4 do 7. Podobno lahko izločimo tudi pacienta 4 in 5, ker tudi epruveta E ni okužena. Torej je okužen ali pacient 6 ali pacient 7. Kateri od njiju, pa moramo še preveriti, kar lahko naredimo le s testom epruvete G ali H.

Računalniško ozadje

Skupinsko testiranje je tehnika, ki so jo prvič uporabili med drugo svetovno vojno, in omogoča izvedbo občutno manj testov, če je v testirani populaciji le malo pozitivnih. Namesto da testiramo vsakega posameznika posebej, test izvedemo nad večjo skupino. V primeru, da je test skupine negativen, smo prihranili veliko testiranj. V primeru, da je test skupine pozitiven, pa moramo izvesti še nadaljnja testiranja.

Ta tehnika se ne uporablja le v biologiji, ampak tudi v inženirskih strokah in računalništvu, predvsem v povezavi z odkrivanjem napak.



Ana, Branko, Cvetka, Darko in Edi se igrajo telefonček. Ana šepetaje črkuje besedo Branku (npr. T-E-K-M-O-V-A-N-J-E). Ta zašepeta besedo Cvetki, a pri tem naredi napako: lahko zamenja eno črko (npr. T-E-K-M-O-S-A-N-J-E) ali pa eno črko izpusti (npr. T-E-K-O-V-A-N-J-E).

Cvetka nato zašepeta črkovano besedo naprej Darku, a tudi ona naredi eno napako. Tudi Darko, ki besedo zašepeta Ediju, naredi eno napako ... Vsak od njih vedno naredi natanko eno napako. Napaka je lahko tudi večkrat na isti črki oz. poziciji v besedi.

Če Ana zašepeta P-R-E-V-I-D-N-O-S-T, katera od navedenih besed bi lahko prišla do Edija?

- A. P-R-V-I-N-O
- B. P-E-E-V-I-D-N-O-S-T
- C. P-E-V-I-K-L-O-T
- D. P-R-E-V-I-D-N-O-S-T



Rešitev

Ana pravilno črkuje besedo Branku. Branko naredi eno napako, ko besedo črkuje Cvetki. Cvetka naredi še eno napako, ko besedo črkuje Darku. In na koncu Darko naredi še eno napako pri črkovanju besede Ediju. Torej so skupaj tri napake v besedi, ko ta prispe do Edija.

Odgovor A (P-R-V-I-N-O) ni pravilen, saj ima beseda le 6 črk, kar pomeni, da smo morali izpustiti štiri črke. To ni možno, ker so v besedi tri napake, torej tudi največ tri brisanja črke.

Odgovor C (P-E-V-I-K-S-O-T) ni pravilen, saj ima beseda le 8 črk, kar pomeni, da smo izpustili dve črki (to pomeni dve napaki). Poleg tega pa ima beseda še dve novi črki, K in L, kar pomeni še dve zamenjavi črk. To je skupaj več kot le tri napake.

Odgovor D (P-R-E-V-I-D-N-O-S-T) tudi ni pravilen, saj ima beseda 10 črk, kar pomeni, da ni nobene izpuščene črke. Torej morajo biti tri zamenjave črk. Ker pa je ta beseda enaka originalni besedi, bi morebitna zamenjava črk lahko bila narejena le v paru. Ker pa so bile narejene natanko tri napake, to ni možno.

Pravilen je le odgovor B (P-E-E-V-I-D-N-O-S-T). Tudi ta beseda ima 10 črk, zato ni bila nobena črka izpuščena. Črka R je bila zamenjana z E kot ena napaka. Druga napaka je zamenjava ene črke (recimo črke D v črko L). Pri tretji napaki pa je ta črka ponovno zamenjana za prvotno črko (torej L v D). Tako bi, na primer, lahko imeli naslednje zaporedje napak:

P-R-E-V-I-D-N-O-S-T → P-E-E-V-I-D-N-O-S-T → P-E-E-V-I-L-N-O-S-T → P-E-E-V-I-D-N-O-S-T

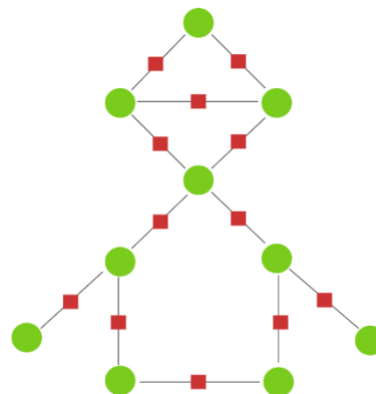
Računalniško ozadje

Podobne napake se pojavljajo tudi pri prenosu podatkov, zato so postopki prenosa sestavljeni tako, da lahko določene napake zaznajo ali pa celo popravijo.

Zaradi epidemije so morali zapreti več trgovin. Vendar pa mora imeti vsako naselje vsaj eno trgovino v bližini, da je zagotovljena oskrba prebivalcev s hrano. Za trgovino rečemo, da je v bližini naselja, če na poti od naselja do trgovine ni potrebno iti skozi nobeno drugo naselje.

Na desni sliki je zemljevid, na katerem zeleni krogi predstavljajo naselja, rdeči kvadrati pa trgovine. Povezave med njimi pa predstavljajo ceste.

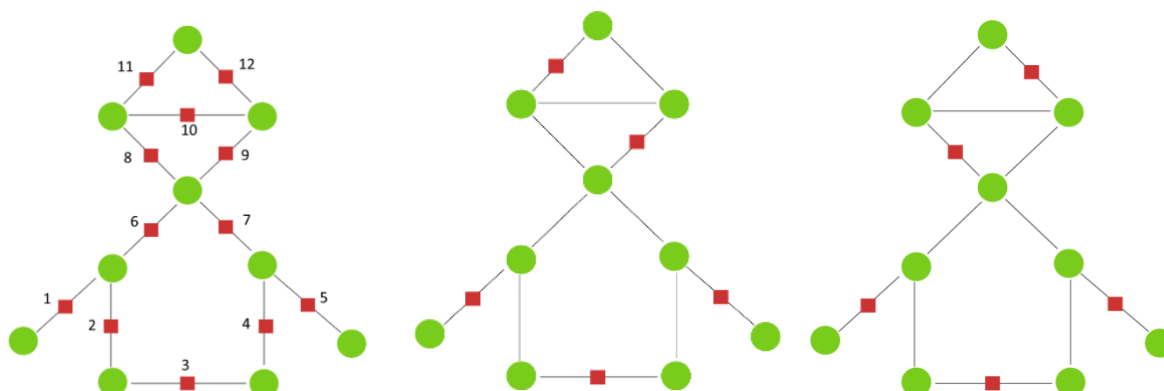
Najmanj koliko trgovin mora ostati odprtih?



Rešitev

Odprtih mora ostati najmanj 5 trgovin.

Dvanajst trgovin na zemljevidu smo oštevilčili, kot kaže leva slika.



Sigurno morata ostati odprti trgovini 1 in 5 saj oskrbujeta najbolj levo in najbolj desno naselje. Od trgovin 2, 3 in 4 zadostuje, da je odprta le trgovina 3. Sedaj moramo še razmisliti, katere trgovine s številkami od 8 do 12 lahko zapremo, da bodo ustrezno oskrbovana vsa štiri zgornja naselja. Ena sama odprta trgovina za štiri naselja ne zadostuje, zato morata biti odprti najmanj dve. To sta lahko trgovini 8 in 12 ali pa trgovini 9 in 11. V obeh primerih so oskrbljena vsa štiri naselja. Več odprtih trgovin tako ne potrebujemo. Skupaj moramo torej imeti najmanj 5 odprtih trgovin. Obe možni rešitvi prikazujeta srednja in desna slika.

Računalniško ozadje

Problem v tej nalogi se imenuje problem minimalnega pokritja grafa. Pri tem problemu moramo poiskati najmanjše število povezav tako, da je vsako vozlišče v grafu »pokrito« z eno od izbranih povezav. V naši nalogi so naselja vozlišča, ki jih moramo pokriti, ceste s trgovinami pa so povezave, ki pokrivajo ta vozlišča.



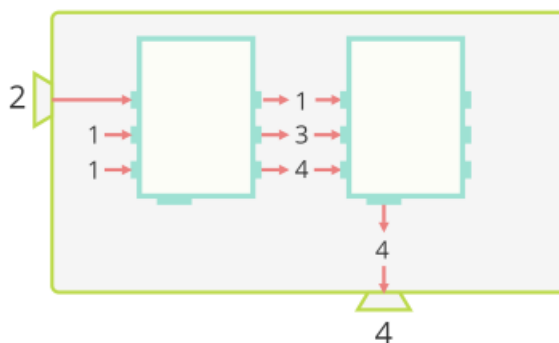
Profesor Bebras je izdelal poseben računski stroj, ki mu na vhod poda neko število in stroj vrne število kot rezultat. Stroj je sestavljen iz več komponent, ki pa vse delujejo na enak način. Vsaka komponenta sprejme na vходу tri števila in jih obdela na naslednji način:

- Če je prvo število 1, vrne tretje število kot izhod računskega stroja.
- Sicer pa naredi naslednje:
 - Prvo število zmanjša za 1, rezultat pa je novo prvo število.
 - Drugo število poveča za 2, rezultat pa je novo drugo število.
 - Tretjemu številu prišteje novo drugo število, rezultat pa je novo tretje število.
 - Vsa tri nova števila (v tem vrstnem redu) prenese kot vhod v naslednjo komponento.

Računski stroj število na vходу prenese kot prvo število v prvo komponento. Ostali dve vhodni števili te komponente sta enaki 1.

Slika kaže, kako računski stroj preračuna vhodno število 2, v tem primeru uporabi dve komponenti. Rezultat je število 4.

Če podamo računskemu stroju število 4, kaj dobimo kot rezultat?



Rešitev

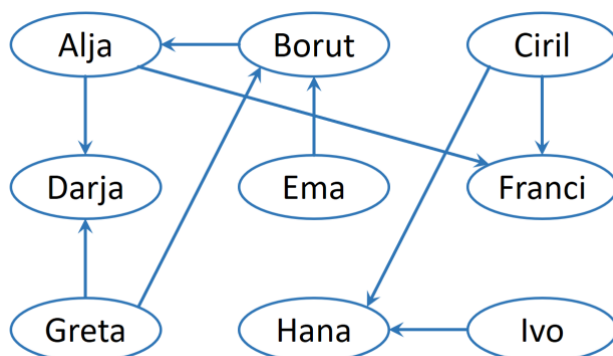
Ker se prvo število v vsaki komponenti zmanjša za ena in ker dobimo rezultat računskega stroja v tisti komponenti, ki ima prvo število enako 1, uporabi računski stroj 4 komponente, da izračuna rezultat. Če sledimo delovanju komponent, dobi prva komponenta na vходу števila (4, 1, 1) in izračuna nova tri števila (3, 3, 4), ki jih poda kot vходу drugi komponenti. Ta spet izračuna nova tri števila (2, 5, 9), ki so vходу v tretjo komponento. Tretja komponenta na podlagi svojega vhoda izračuna nova tri števila (1, 7, 16), ki jih poda kot vходу četrte komponenti. Ker četrta komponenta dobi na vходу prvo število, ki je enako 1, se računanje zaključi in računski stroj kot rezultat vrne tretje število, ki ga je na vходу dobila četrta komponenta, to je število 16.

Računalniško ozadje

Pri reševanju problemov si pogosto pomagamo tako, da bolj zapleten problem razdelimo na več podproblemov enakega tipa. Tak način reševanja problemov imenujemo »deli in vladaj« (angl. *divide and conquer*), uporablja pa se v veliko pomembnih algoritmih. Rekurzija v računalništvu je način reševanja problema, kjer je rešitev odvisna od rešitev manjših primerkov istega problema. Rekurzija je ena od ključnih idej, ki se uporabljajo v računalništvu.

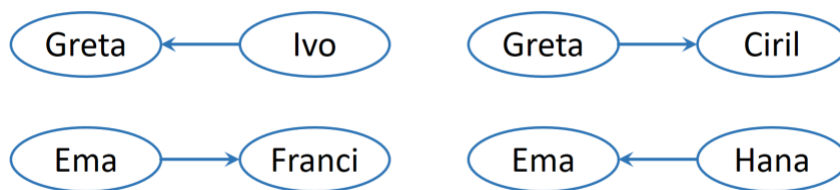


V okviru šolskega šahovskega krožka so na šoli organizirali sobotni šahovski turnir, na katerem je sodelovalo devet mladih šahistov. Dopoldne so že odigrali nekaj partij, katerih rezultate prikazuje spodnja slika. Na sliki so puščice narisane od zmagovalca proti poražencu (na primer, Borut je premagal Aljo, Alja pa je premagala Darjo).



Ker so se dopoldanske partije nepričakovano zavlekle, niso imeli dovolj časa, da bi odigrali vse načrtovane partije do konca. Zato so se dogovorili, da bodo najboljšega šahista med njimi poiskali kar s pomočjo dodatnega pravila: če šahist A premaga šahista B in šahist B premaga šahista C, se šteje, kot da je šahist A premagal tudi šahista C (čeprav nista odigrala partije). Zmagovalec turnirja pa bo tisti šahist, ki bo premagal vse ostale udeležence z upoštevanjem tega dodatnega pravila.

Kljub temu novemu pravilu so ugotovili, da zmagovalca ne morejo določiti. Zato so odigrali še dodatne štiri partije, katerih rezultati so naslednji:



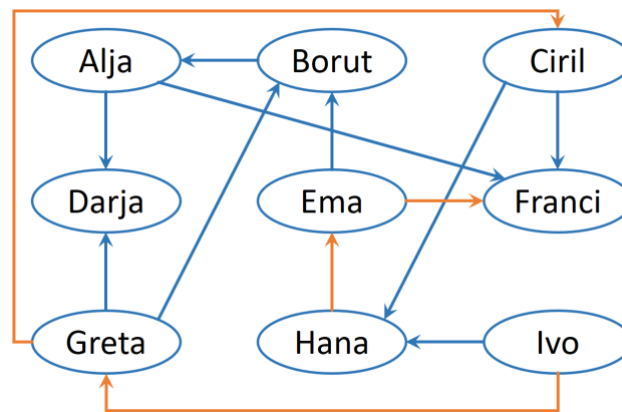
Kdo je zmagal na turnirju?

Rešitev

Na turnirju je zmagal Ivo.

Če lahko z upoštevanjem podanega pravila določimo zmagovalca, to pomeni, da v narisanim grafu rezultatov partij lahko s sledenjem puščic v obratni smeri od katerekoli osebe pridemo vedno do iste osebe (to je do zmagovalca). Torej zmagovalec ne bo imel nobene puščice, ki bi kazala vanj (zmagovalec ima le puščice, ki vodijo do drugih udeležencev turnirja), in to mora veljati za natanko eno osebo. Če take osebe v grafu ni, to pomeni, da z uporabo tega pravila ne moremo določiti zmagovalca. Če je takih oseb več, potrebujemo še dodatne partije, da lahko določimo zmagovalca.

Na prvotni sliki imamo štiri take osebe, kjer puščice kažejo le stran od osebe; to so Greta, Ivo, Ciril in Ema. Ker so odigrali še dodatne štiri partije, v graf dodamo še rezultate teh štirih partij. Dobimo graf na spodnji sliki.



Sedaj vidimo, da Greta ne more biti zmagovalka, ker jo je premagal Ivo. Ciril je izgubil proti Greti, Ema pa je izgubila proti Hani. Torej je Ivo edina oseba, ki je lahko zmagovalec.

Računalniško ozadje

Usmerjen graf lahko uporabimo za predstavitev (nesimetričnih) relacij med pari objektov. V usmerjenem grafu so objekti predstavljeni z vozlišči, puščica od objekta A do objekta B pa predstavlja relacijo med tema dvema objektoma.

V naši nalogi so objekti udeleženci turnirja, relacija pa je »zmagovalec«. Pri tem smo nekaj relacij dobili neposredno iz rezultatov odigranih partij, druge relacije pa smo dodali na podlagi pravila, ki je določalo, da je relacija zmagovalca *tranzitivna*. Ta lastnost navadno ni značilna za športe, kar pravzaprav naredi tekme bolj zanimive: če Alja premaga Boruta in Borut premaga Cirila, ni nikjer rečeno, da bo Alja v tekmi premagala tudi Cirila.

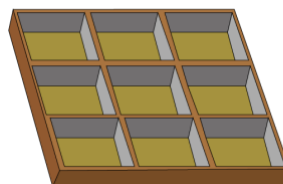
V nalogi opisani problem je pogost v računalništvu: nekaterih objektov medsebojno ne primerjamo neposredno, a še vedno želimo zgraditi nek vrstni red, pri katerem lahko primerjamo vsak par objektov (seveda je možnih več različnih razvrstitev).



Imamo zaboj z devetimi prekatami in devet frnikol.

Izberemo katerokoli število od 0 do 9 frnikol in jih postavimo v zaboj tako, da veljajo naslednja tri pravila:

- Vsaka frnikola je v drugem prekatu.
- Skupno število frnikol v vsaki vrstici je sodo.
- Skupno število frnikol v vsakem stolpcu je sodo.



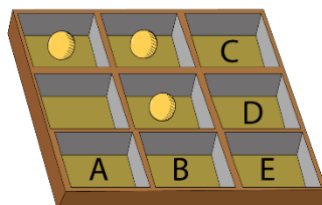
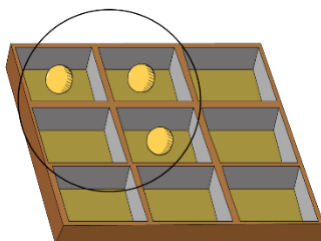
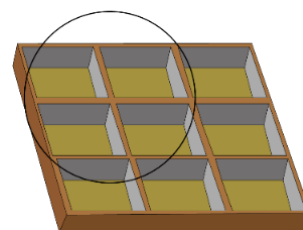
Na koliko različnih načinov lahko postavimo frnikole v zaboj?

Rešitev

Frnikole lahko postavimo v zaboj na 16 različnih načinov.

Najprej se osredotočimo le na del zaboja velikosti 2 x 2.

V tem delu imamo 4 prekate in v vsakem prekatu je lahko ena frnikola ali pa te v prekatu ni. Torej imamo skupaj lahko $2^4 = 16$ različnih načinov za postavitve frnikol v ta del zaboja.



Ko se odločimo, kako bomo zapolnili ta del zaboja, nimamo nobene izbire več glede zapolnitve preostalih prekatov v zaboju, če se držimo postavljenih pravil. Pravili, da mora biti skupno število frnikol v vsakem stolpcu in v vsaki vrstici sodo,

natančno določata, kako moramo zapolniti preostale prekate (torej, ali v nek prekat postavimo frnikolo ali ne). Na primer, če imamo del zaboja zapolnjen, kot kaže leva slika, moramo zapolniti še mesta od A do E (desna slika). Ker ima prvi stolpec le eno frnikolo, moramo na mesto A postaviti frnikolo, da bo ta stolpec imel sodo število frnikol (to je 2). V drugem stolpcu že imamo dve frnikoli, zato mora biti mesto B prazno. Podobno razmislimo še za preostala mesta: C mora biti prazno, D in E pa morata vsebovati frnikolo.

Torej lahko v zaboj postavimo frnikole na 16 različnih načinov.

Računalniško ozadje

Varen in pravilen prenos podatkov je pomembna naloga v računalništvu. Eden od načinov, da zagotovimo, da se podatki med prenosom ne izgubijo ali spremenijo, je, da uporabimo parnostni bit.

Parnostni bit je dodatni bit, ki ga dodamo na konec originalnega sporočila in je odvisen od vsebine sporočila. Ko sporočilo prispe do prejemnika, ta ponovno izračuna parnostni bit. Če se izračunana vrednost ne ujema s poslano vrednostjo, to pomeni, da je pri prenosu prišlo do napake.

V naši nalogi sta zadnja vrstica in zadnji stolpec služila kot parnostni biti. Če bi zaboj frnikol poslali kot sporočilo, bi prejemnik lahko preveril vsoto vseh frnikol v vsakem stolpcu in v vsaki vrstici. Če ta ne bi bila soda, bi sporočil pošiljatelju, da je šlo med prenosom nekaj narobe.



V tovarni avtomobilov pripravljajo novo linijo za proizvodnjo avtomobilov. Vsak izdelan avto se lahko razlikuje v štirih lastnostih: barvi, notranjosti, pnevmatikah in motorju. Možnosti so prikazane v spodnji tabeli.

barva	notranjost	pnevmatike	motor
0 modra	0 črno usnje	0 športne	0 dizel
1 rdeča	1 belo usnje	1 mestne	1 bencin
2 bela	2 črno blago	2 cestne	2 hibrid
3 črna	3 belo blago	3 terenske	3 plin
4 zelena	4 rdeči nadstandardni materiali	4 nadstandardne	4 elektrika

Vsaki kombinaciji lastnosti izdelanega avta bodo v tovarni določili številko, in sicer od 0 do 624. Številko 0 bodo dodelili avtu z lastnostmi 0 0 0 0, to je avtu modre barve s črnim usnjem v notranjosti, športnimi pnevmatikami in dizelskim motorjem. Avto s številko 1 bo modre barve s črnim usnjem, športnimi pnevmatikami in bencinskim motorjem (lastnosti 0 0 0 1). Avto s številko 2 bo imel lastnosti 0 0 0 2, medtem ko bo avto s številko 5 imel lastnosti 0 0 1 0.

Tako bo na primer avto z lastnostmi 2 0 1 4 bel avto s črnim usnjem, mestnimi pnevmatikami in električnim motorjem. Tak avto bo dobil številko 259.

Katero številko bo dobil zelen avto z rdečimi nadstandardnimi materiali v notranjosti, z nadstandardnimi pnevmatikami in motorjem na plin?

Rešitev

Število 623 določa zelen avto z rdečimi nadstandardnimi materiali v notranjosti, z nadstandardnimi pnevmatikami in motorjem na plin.

Lastnosti tega avtomobila lahko zapišemo kot 4 (zelena), 4 (rdeči nadstandardni materiali), 4 (nadstandardne pnevmatike) in 3 (plin), torej 4 4 4 3. Če ta števila pretvorimo iz petiškega v desetiški sistem, dobimo število 623, kar je številka tega avtomobila. In zakaj ravno petiški sistem?

Na podlagi podanih primerov lahko ugotovimo, da vsaka naslednja vrsta motorja poveča številko avtomobila za 1. Vsaka naslednja vrsta pnevmatik poveča številko avtomobila za 5, vsaka naslednja vrsta notranjosti pa za 25 (= 5x5). Podobno vsaka naslednja barva avtomobila poveča številko avtomobila za 125 (= 25x5). Torej je številka izbranega modela avtomobila enaka: (barva) 4×125 + (notranjost) 4×25 + (pnevmatike) 4×5 + (motor) $3 = 623$.

Pravzaprav smo vsak model avtomobila zakodirali v petiškem sistemu, ker imamo po 5 različnih vrst vsake lastnosti avtomobila. Izbrane lastnosti so zakodirane s številom 4443_5 , kar je v desetiškem sistemu enako 623.

Računalniško ozadje

Spreminjanje osnove številskega sistema je lahko zelo uporabno, saj tako lahko sporočamo informacije, ki v desetiškem sistemu ne bi bile tako razumljive.

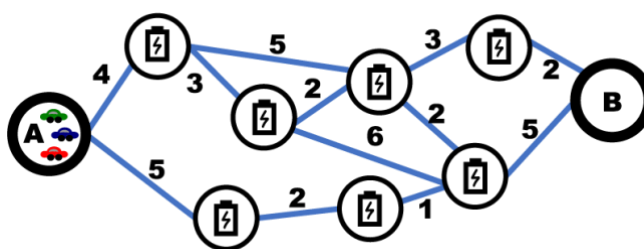
Desetiški sistem, ki ga uporabljamo v našem vsakdanjem življenju, ni tisti, ki ga uporabljajo računalniki za shranjevanje števil. Računalniki uporabljajo dvojiški (ali *binarni*) sistem, ki uporablja le ničle in enice. Kadar v računalnik vpišemo številko, jo računalnik pretvori v dvojiški zapis in to uporabi za računanje. Pravzaprav računalniki v taki binarni obliki shranjujejo vse podatke, ne le števila. Vsi podatki so zapisani kot zaporedje ničel in enic, tudi zvok, slike, filmi, 3D modeli in podobno.



Modri avto lahko z enim polnjenjem prevozi 4 km in za polnjenje porabi 3 minute. Zeleni avto lahko s polno baterijo prevozi 5 km in potrebuje 4 minute za polnjenje. Rdeči avto pa lahko prevozi 6 km, za polnjenje pa potrebuje 5 minut.

Ni nujno, da se avto ustavi na vsaki polnilni postaji na poti, a če se ustavi na polnilni postaji, mora baterijo napolniti do konca, za kar vedno porabi celoten čas polnjenja (ne glede na to, kako prazna je njegova baterija). Vsi avtomobili vozijo z isto hitrostjo in za 1 km potrebujejo 1 minuto. Na začetku imajo vsi trije avtomobili popolnoma napolnjene baterije.

Slika kaže zemljevid cest in polnilnih postaj med krajema A in B. Številke, ki so pripisane cestam, določajo razdaljo med polnilnimi postajami oz. kraji v kilometrih.



Če upoštevamo ta zemljevid, kateri avtomobil lahko najhitreje pride iz kraja A v kraj B?

Rešitev

Zeleni avtomobil lahko pride najhitreje iz kraja A v kraj B.

Zeleni avto lahko gre po poti z razdaljami {5, 2, 1, 5}, kar je skupaj 13 km, za kar porabi 13 minut. Ustaviti se mora na prvi in na tretji polnilni postaji, za kar porabi 8 minut (2×4 minute). Torej za pot do kraja B porabi skupaj $13 + 8 = 21$ minut.

Najhitrejša pot za modri avto traja 26 minut, če sledi poti {4, 3, 2, 3, 2}, za kar porabi 14 minut, ustaviti pa se mora na vseh štirih polnilnih postajah, za kar porabi še dodatnih 12 minut (4×3 minute). Skupaj torej za pot porabi $14 + 12 = 26$ minut.

Najhitrejša pot za rdeči avto pa je 23 minut po poti {5, 2, 1, 5}, za kar porabi 13 minut. Poleg tega se mora ustaviti še na prvi in tretji polnilni postaji, kar je še dodatnih 10 minut (2×5 minut) za polnjenje. Skupaj torej porabi $13 + 10 = 23$ minut.

Računalniško ozadje

Iskanje najkrajše poti med dvema točkama je pogost problem v računalništvu (rešitev je zelo uporabna npr. pri cestnem omrežju ali telekomunikacijskem omrežju). Za reševanje tega problema v poljubnem omrežju so bili razviti različni algoritmi. Eden takih je Dijkstrov algoritem, ki ga je objavil nizozemski računalničar Edsger Dijkstra. Ta algoritem poišče najkrajšo pot od začetne točke do vseh ostalih točk v grafu.

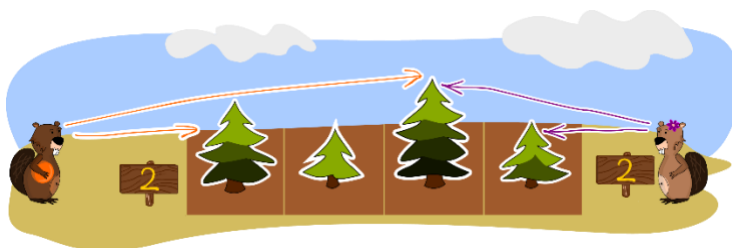
Drevesni sudoku

Bobrovo polje je razdeljeno na 16 parcel, ki so urejene v mrežo 4 x 4. Na vsaki parceli je posajeno natanko eno drevo.

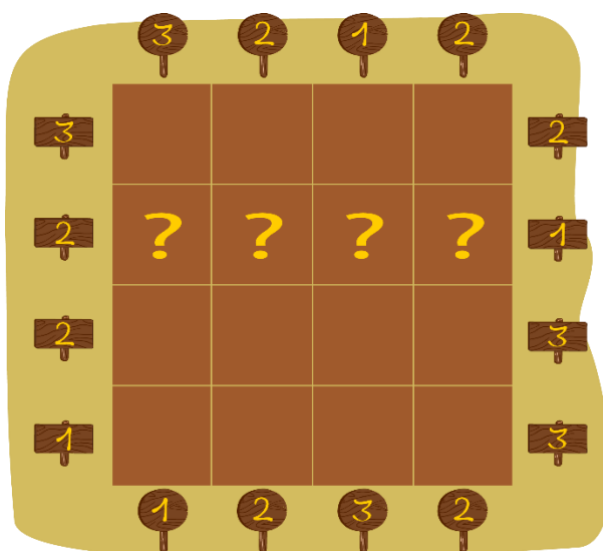
Šestnajst dreves različnih velikosti, 1 () , 2 () , 3 () in 4 () , je posajenih na Bobrovem polju tako, da ustrezajo naslednjemu pravilu:

- Vsaka vrsta vsebuje natanko eno drevo vsake velikosti.
- Vsak stolpec vsebuje natanko eno drevo vsake velikosti.

Bobri radi gledajo drevesa. Ko jih opazujejo v ravni črti (kot kaže spodnja slika), ne morejo videti vseh dreves, saj so nižja drevesa skrita za višjimi. Bobri so na začetek in na konec vsake vrstice in vsakega stolpca tega 4 x 4 velikega polja postavili table, na katere so zapisali, koliko dreves se vidi s te pozicije, kjer stoji tabla.



Na spodnji sliki so zapisana števila na tablah, a manjkajo drevesa. Katera drevesa so posajena (namesto vprašajev) v drugi vrstici polja, gledano od leve proti desni?



Rešitev

Pravilen odgovor je 3 (🌲), 1 (🌲), 2 (🌲) in 4 (🌲).

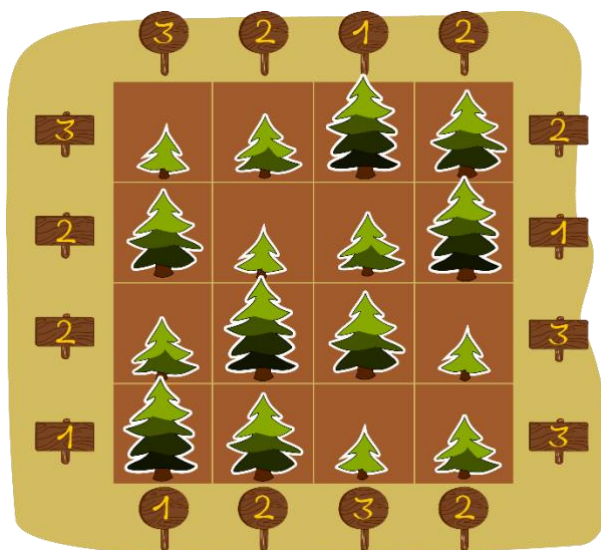
Za podano polje z vpisanimi številkami na tablah je ena sama rešitev. Poskusimo jo poiskati. Vse enice na tablah nam pomagajo, da v mrežo pravilno postavimo vsa štiri najvišja drevesa: če bober z neke pozicije lahko vidi le eno drevo, mora biti prvo drevo, ki ga vidi, največje (velikosti 4). Na ta način lahko vpišemo tri drevesa velikosti 4, četrto pa mora biti, zaradi postavljenega pravila, v edini preostali vrstici in stolpcu, v katerih drevesa velikosti 4 še ni.



Tudi ostala drevesa postavimo glede na to pravilo (ki je enako pravilu igre sudoku). Ker v drugi vrstici z leve vidimo le dve drevesi (torej poleg najvišjega vidimo le prvo drevo), mora biti drevo v prvem stolpcu velikosti 3, da zakrije drugi dve drevesi. Postavimo še v tretjo vrstico drevo velikosti 3. V prvi stolpec ga ne smemo postaviti, ker je tako veliko drevo v prvem stolpcu že v drugi vrstici. In če v tretji vrstici z desne vidimo tri drevesa, mora biti drevo velikosti 3 v tretjem stolpcu (če bi bil v četrtem stolpcu, bi videli z desne le dve drevesi). Podobno sklepamo tudi za četrto vrstico: če z desne vidimo 3 drevesa, drevo velikosti 3 ne sme biti v četrtem stolpcu (saj bi v tem primeru videli le dve drevesi: 3 in 4), torej je lahko le v drugem stolpcu (ker v tretjem stolpcu je tako veliko drevo že v tretji vrstici). Edina še prosta parcela za četrto drevo velikosti 3 je torej v zadnjem stolpcu prve vrstice.



Sedaj moramo na parcele razporediti le še drevesa velikosti 1 in 2. V prvi vrstici z leve vidimo tri drevesa, torej sta prvi dve drevesi v tej vrstici velikosti 1 in 2, po vrsti (če bi bilo drevo velikosti 1 desno od drevesa velikosti 2, bi se skrilo za to drevo in bi lahko videli le dve drevesi: 2 in 4). V prvi stolpec postavimo še drevo velikosti 2 na edino prazno parcelo. Podobno zapolnimo tudi tretjo vrstico z drevesom velikosti 1, nato pa četrti stolpec z drevesom velikosti 2. Manjkajoče drevo velikosti 1 postavimo še na edino prazno parcelo v drugem stolpcu in v četrti vrstici. In za konec postavimo še drevo velikosti 2 na edino še prazno parcelo na polju.



Računalniško ozadje

V nalogi smo uporabili temeljna znanja iz računalništva. Poiskati smo morali rešitev problema, ki zadošča podanim omejitvam. Poleg tega pa je naloga od nas zahtevala sposobnost, da rekonstruiramo objekt na podlagi delnih informacij tako, da uporabimo znanje glede lastnosti tega objekta. Slednje lahko s pridom uporabimo pri stisnjeni predstavitvi objekta. Poleg tega pa lahko tako tudi iščemo napake v predstavitvi podatkov.

Popravljanje napak je splošna tehnika v računalništvu, s katero zagotovimo večjo zanesljivost pri prenosu podatkov preko različnih kanalov, pri katerem lahko pride do izgube podatkov ali pa se ti pokvarijo (kakšna ničla se nenamerno spremeni v enico ali obratno). Popravljanje napak zajema rekonstrukcijo netočnih podatkov, tako da dobimo podatke spet v njihovi originalni in pravilni obliki.

Pregled nalog

			2.	3.	4.	5.	6.	7.	8.	9.	SŠ
Izberi medvedka	Kanada										
Bobrov salon	Kanada										
Skrivna sporočila	Švica										
Vesoljski ZOO	Irska										
Znamke	Avstralija										
Ogled razstave	Švica										
Skodelice	Irska										
Najtežja škatla	Japonska										
Otok zakladov	Avstrija										
Visoki stolpi	Kanada										
Pobrisana tabela	Švica										
Barvanje hiše	Japonska										
Drevesna struktura	Litva										
Zapomni si geslo	Litva										
Zvezdice in lune	Belgija										
Knjižnica	Kanada										
Gledališka predstava	Slovaška										
Vezenje	Hrvaška										
Drevesni sudoku	Švica										
Vratar Jaka	Ciper										
Nova hiša	Južna Koreja										
Polaganje ploščic	Japonska										
Poskočni kenguru	Litva										
Sestavljanke	Makedonija										
Ljubzenska pisma	Slovenija										
Epidemija	Tajvan										
Toplotna karta	Nemčija										
Vedra z vodo	Nova Zelandija										
Zmedeni poštar	Ciper										
Nalepke	Finska										
Potujoča pekarna	Slovenija										
Prepisovanje	Nova Zelandija										
Rezultati preiskav	Irska										
Telefonček	Indija										
Trgovine	Avstrija										
Računski stroj	Nemčija										
Šah mat	Vietnam										
Frnikole v zaboju	Vietnam										
Tovarna avtomobilov	Salvador										
Električni avtomobili	Irska										
Drevesni sudoku	Švica										

Avtorji nalog

Ágnes Erdősné Németh, Madžarska

Aidan Mooney, Irska

Alenka Kavčič, Slovenija

Alisher Ikramov, Uzbekistan

Andrej Blaho, Slovaška

Anja Koron, Slovenija

Anna Morpurgo, Italija

Anoki Eischer, Avstrija

Anton Chukhnov, Rusija

Arnheidur Gudmundsdottir, Islandija

Blaž Abram, Slovenija

Brianna Maloney, Nova Zelandija

Carla Martet-Doleu, Irska

Carlo Belletini, Italija

Chia-Yi Ku, Tajvan

Chris Roffey, Velika Britanija

Christian Dazko, Madžarska

Daniela Bezáková, Slovaška

David Simeonovski, Severna Makedonija

Dimitris Hadjipantelis, Ciper

Ela Veža, Hrvaška

Emil Stankov, Severna Makedonija

Eric Song, Nova Zelandija

Eslam Mohamed Wageed, Egipt

Eugenio Bravo, Španija

Ezra Templonuevo, Filipini

Florentina Voboril, Avstrija

Gerald Futschek, Avstrija

Hamed Mohebbi, Iran

Heikki Hyyrö, Finska

Hiroyuki Nagataki, Japonska

Humberto Sermeño, Salvador

Husnul Hakim, Indonezija

Hyeonah Lee, Južna Koreja

Issac Siu, Nova Zelandija

J.P. Pretti, Kanada

Janez Demšar, Slovenija

Jean-Philippe Pellet, Švica

Jinbao Zhang, Kitajska

Jonatan Pipping, Švedska

Jose Carlos Ramirez Peralta, Salvador

Judith Lin, Tajvan

Juraj Hromkovič, Švica

Kevin Casey, Irska

Khairul Anwar Mohamad Zaki, Malezija

Kris Coolsaet, Belgija

Kwangsik Moon, Južna Koreja

Kyra Willekens, Nizozemska

Lidia Feklistova, Estonija

Luis Mario Ramirez Peralta, Salvador

Maayan Levy, Nova Zelandija

Maciej Sysło, Poljska

Madhavan Mukund, Indija

Margot Phillipps, Nova Zelandija

Marielle Léonard, Francija

Marios O. Choudary, Pakistan

Megan MacDiarmid, Nova Zelandija
Melinda Phelps , Avstralija
Meng-ting Tsai, Tajvan
Michael Page, Avstralija
Michael Weigend, Nemčija
Michal Winczer, Slovaška
Milan Rajković, Srbija
Mile Jovanov, Severna Makedonija
Monika Tomcsányiová, Slovaška
Motomasa Omura, Japonska
Nicholas Grace, Nova Zelandija
Nora A. Escherle, Švica
Nursultan Akhmetov, Kazahstan
Omran Kouba, Sirija
Pedro Marcelino, Portugalska
Peter Tomcsányi, Slovaška
Prathyush Ponnekanti, Indija
Rechilda Villame, Filipini
Regula Lacher, Švica
Rosemary Monahan, Irska
Sam Zhuang, Nova Zelandija
Sarah Chan, Kanada
Sohpie Koh, Singapur
Sung-Ryul Kim, Južna Koreja
Susanne Datzko, Švica
Špela Cerar, Slovenija
Taina Lehtimäki, Irska
Takeharu Ishizuka, Japonska
Thomas Grubb, Avstralija
Thomas Ioannou, Ciper
Tina Žakelj, Slovenija
Tolmantas Dagys, Litva

Tom Grubb, Avstralija
Tom Naughton, Irska
Tony René Andersen, Norveška
Ulrich Kiesmüller, Nemčija
Ungyeol Jung, Južna Koreja
Vaidotas Kinčius, Litva
Valentina Dagiene, Litva
Victor Koleszar, Urugvaj
Vipul Shah, Indija
Vu Luan, Vietnam
Vu Van Luan, Vietnam
Wilfried Baumann, Avstrija
Willem van der Vegt, Nizozemska
Wolfgang Pohl, Nemčija
Xavier Muñoz, Španija
Yunge Yu, Nova Zelandija