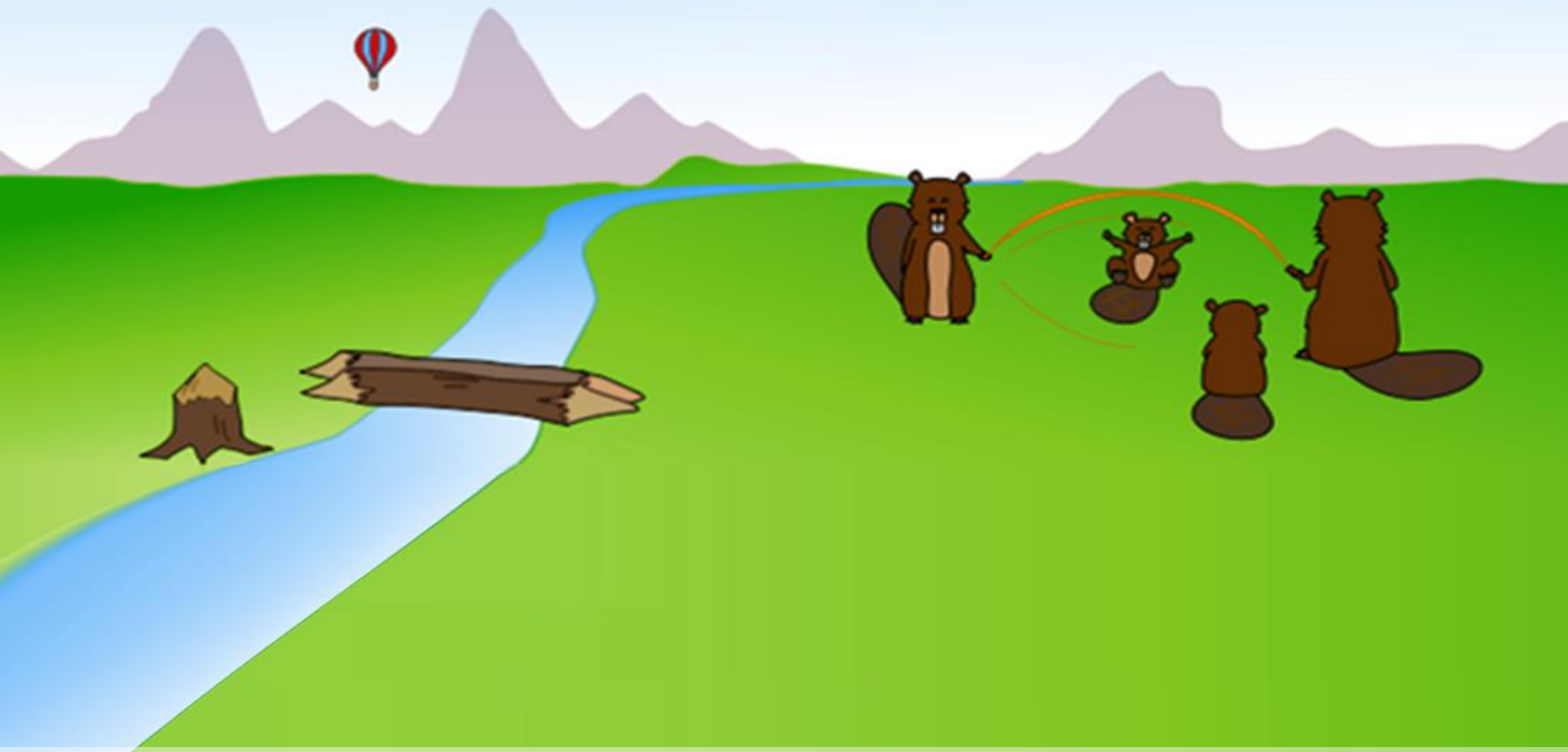


Bober 2017/18

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)
Manca Zaviršek (Osnovna šola Danile Kumar)
Špela Cerar (PEF, Univerza v Ljubljani)

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

Kazalo nalog

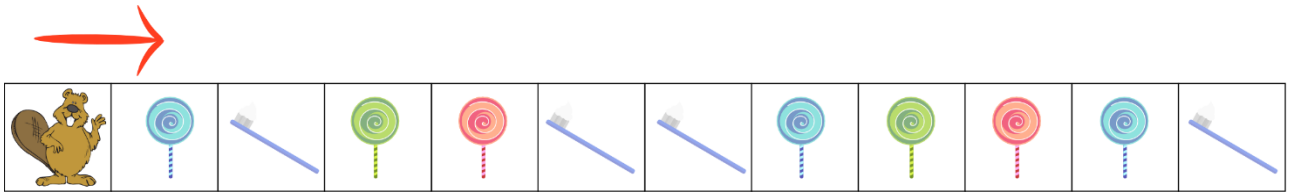
Hodnik lizik	5
Zmešane živali (lažja)	6
Ptičja hišica	7
Parkirišče	8
Pasje menjave	9
Prostozidarska šifra	10
Sporočilca	11
Francijeve nalepke	12
Lov na jagodo	13
Pot domov	14
Zmešane živali (težja)	15
Barvanje ograje	16
Koliko je ura?	17
Na igrišče	18
Pumptrack	20
Mesto krožišč	22
Preste	24
Tek po hlodih	25
Žaba Petra in žabec Toni	26
Pakiranje jabolk	27
Tek med prostori	28
Krajši zapis	29
Labirint	30
Levo-desno	32
Picerija	33
Presedanje	34
Rušenje zidov	35
Satovje	36
Skakanje po štorih	37
Žogice	38
Luči v pisarnah	40
Razpolavljanje ogrlice	41
Drsalci	42
Knjižni klub	43
Kocka	44
Mavrični stolp	46
Tuneli v bobrišču	47
Vdor v muzej	49
Oklepajoči okraski	50
Robot	51
Skrivna tipkovnica	53
Speči bober	54
Stiskanje slik	55
Nižanje stroškov	57
Zabava	58
Ena ena preveč	59
Delo na kmetiji	60
Nagrobnik	62
Obračanje denarja	63
Prepoznavanje števil	64
Računalniški center	66
Razbijanje kode	68
Robotski čistilci	70
Uspavalni napoj	71
Zlato	73
Igra s fižoli	75
Honomakato	76
Kaj je za kosilo?	78
Kovanci	79
Prenos datotek	80
Prepogibi	81
Robotski čistilec	83
Šola v Naughtinghamu	85
Še razpolavljanja ogrlice	87

Hodnik lizik

2.–3. razred

BOBRČEK JE NALETEL NA DOLG HODNIK Z LIZIKAMI IN ZOBNIMI ŠČETKAMI.

- PO HODNIKU GRE VEDNO NAPREJ IN SE NIKOLI NE VRAČA.
- OB VSAKI ZOBNI ŠČETKI SI UMIJE ZOBE.
- KO POLIŽE DVE LIZIKI, NE SME POJESTI NASLEDNJE, DOKLER SI NE UMIJE ZOB.
- LIZIKE ALI ZOBNE ŠČETKE NE MORE VZETI S SEBOJ NA NASLEDNJI KORAK.

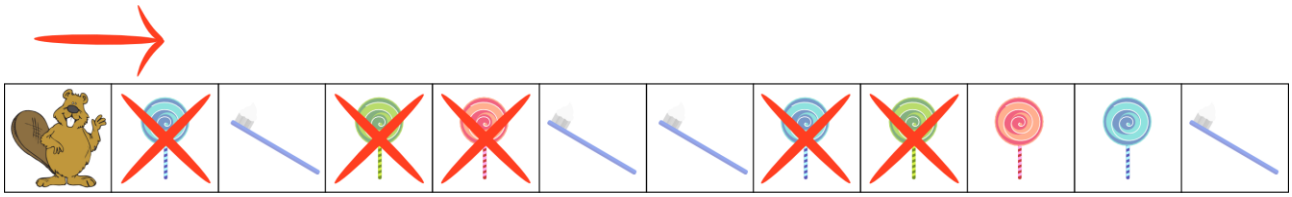


KOLIKO LIZIK BO POJEDEL?

Rešitev

PET.

BOBRČEK ZAČNE SVOJ SPREHOD PO HODNIKU IN POJE PRVO LIZIKO. V DRUGEM KORAKU SI UMIJE ZOBE. SLEDITA DVE ZAPOREDNI LIZIKI, PO KATERIH SI MORA PONOVNO UMITI ZOBE. NATO NALETI NA ŠTIRI ZAPOREDNE LIZIKE. POLIŽE LAHKO SAMO DVE, SAJ SI MORA PO DVEH LIZIKAH UMITI ZOBE.



Računalniško ozadje

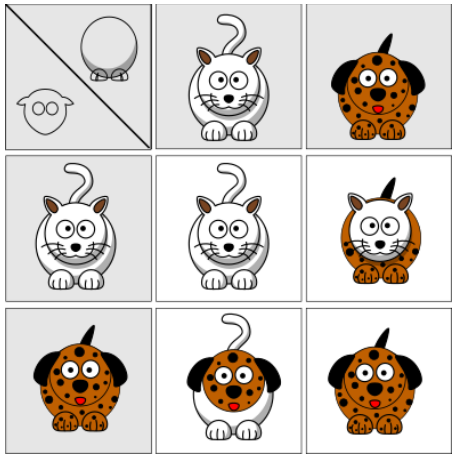
Bobrček mora polizati čim več lizik. Pri tem naleti na »omejitev«. To je pravilo, ki te sili, da moraš kaj narediti, ali pa ti kaj prepoveduje. Tudi pri igranju igre moraš slediti pravilom, ki so značilna za igro. Vsaka igra ima svoja pravila, ki jim moraš slediti, če želiš priti do zmage.

Pri računalništvu so omejitve zelo pomemben del problema, ki lahko reševanje problema otežijo.

Zmešane živali (lažja)

2.–3. razred

V DEŽELI ČIRA ČARA IMAJO MLADIČE TUDI RAZLIČNE ŽIVALI. MLADIČ IMA OČETOVO GLAVO IN MAMIN TRUP. SLIKA KAŽE MLADIČE MAČKOV IN PSOV Z MAČKAMI IN PSICAMI.



KAJ SE ZGODI S KOZAMI, KRAVAMI IN LEVI? ČE ZMEŠAMO **KRAVO** IN **KOZLA**, DOBIMO **KRAVZLA**. TA JE ŽE NARISAN V TABELI.

KAJ PA **LEVBIK** IN **KRAVLEV**? POVLECI PUŠČICI OD NJIJU DO NJUNIH MEST V TABELI!

Rešitev

Računalniško ozadje

Tabele so eden od pomembnih načinov shranjevanja podatkov v računalniku. Vse, kar postavimo v isto vrstico ali stolpec, si je v nekih lastnostih podobno.

Ptičja hišica

2.–3. razred

BABICA MARINKA JE MAJO VPRAŠALA, KAKŠNO PTIČJO HIŠICO ŽELI ZA ROSTNI DAN.

MAJA BI RADA HIŠICO Z DVEMA OKNOMA IN SRČKOM.

OBKROŽI PTIČJO HIŠICO, KI JO JE KUPILA BABICA.

Rešitev

HIŠICO 4, SAJ IMA DVE OKNI IN EN SRČEK.

HIŠICA 1 IMA LE ENO OKNO IN DVA SRČKA. HIŠICA 2 NIMA SRČKA. HIŠICA 3 IMA 3 OKNA, A NIMA SRČKA. HIŠICA 5 IMA SAMO ENO OKNO. HIŠICA 6 IMA 3 OKNA.

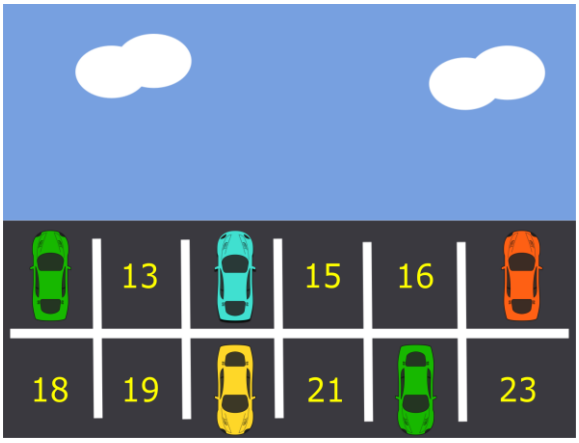
Računalniško ozadje

Ko se lotimo reševanja problema, moramo prepoznati, katere so njegove osnovne lastnosti.

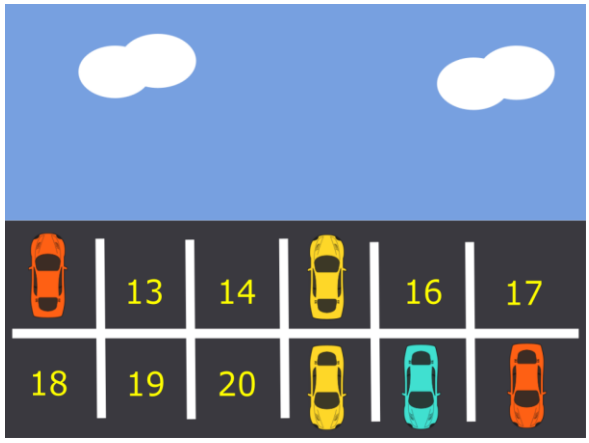
Naloga se osredotoča na prepoznavanje objektov z določenimi značilnostmi in izločanje objektov, ki ne ustrezajo želenim značilnostim.

Parkirišče

2.–5. razred



PARKIRIŠČE V PONEDELJEK



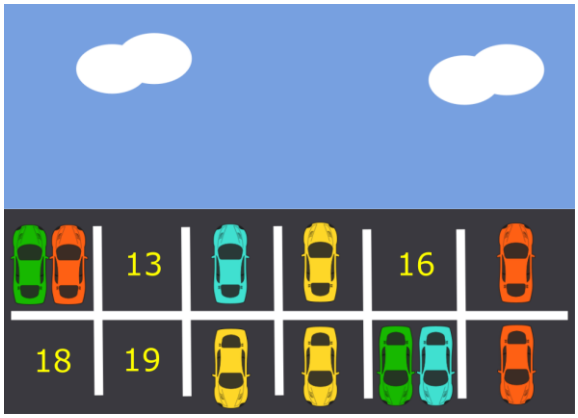
PARKIRIŠČE V TOREK

NAPIŠI ŠTEVILKE PARKIRIŠČ, KI SO BILA PROSTA OBA DNEVA!

Rešitev

13, 16, 18 IN 19.

SPODNJA SLIKA PRIKAZUJE ZASEDENOST PARKIRIŠČA V OBEH DNEH.



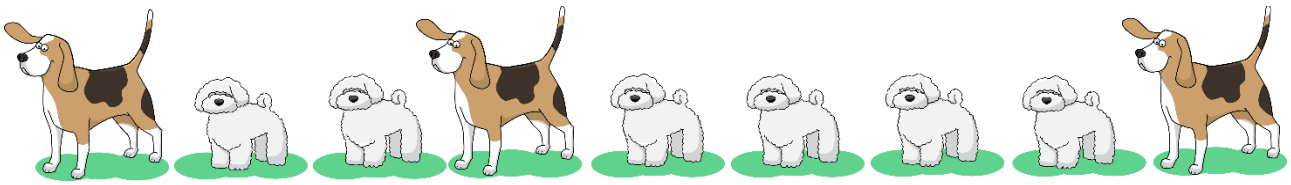
PARKIRIŠČE V PONEDELJEK IN TOREK

Računalniško ozadje

Matematiki bi te tule poučili, da si računal *presek dveh množic*. Če še ne veš, kaj je presek in kaj množica, nič hudega. Boš že še izvedel. Tudi računalnikarji jih pogosto uporabljamo za opisovanje zbirk kakih reči.

Pasje menjave

2.–5. razred



PSI SE LAHKO ZAMENJUJEJO TAKO, DA **SOSEDNJA** PSA ZAMENJATA SVOJI MESTI.

VELIKI PSI BI RADI BILI SKUPAJ.

NAJMANJ KOLIKO ZAMENJAV SOSEDOV BO POTREBNIH, DA BODO VSI VELIKI PSI SKUPAJ?

Rešitev

6

PES NA LEVI SE ZAMENJA NAJPREJ S PSOM NA SVOJI DESNI, NATO PA ŠE Z NASLEDNJIM PSOM, PA BO ZRAVEN PSA NA SREDI.

PES NA DESNI SE MORA NA PODOBEN NAČIN ZAMENJATI S ŠTIRIMI.

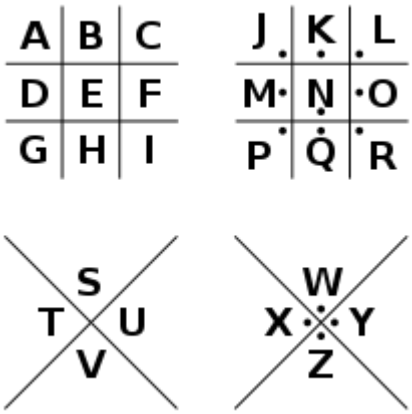
Računalniško ozadje

Računalnikarji si postavijo vprašanje »koliko korakov potrebujemo, da ... « vedno, kadar razmišljajo o tem, kako hiter bo določen program.

Prostozidarska šifra

3.–5. razred

Bober Benjamin in njegova soseda Katja si pod vrtno ograjo puščata sporočila, zapisana s skrivno šifro. Zapomnila sta si jo s pomočjo slik na desni.



Na primer, besedo SOSEDA bi Katja in Benjamin zapisala takole: 

Benjamin je od Katje prejel to sporočilo:


Kaj je želela Katja sporočiti Benjaminu?

A. Počakala ga bo na mostu.
B. Videla se bosta ob petih.
C. Spekla mu bo pecivo.
D. Dobila je novo rolko.

Rešitev

Pravilni odgovor je B.

V sporočilu piše: PRIDI OB PETIH.

Računalniško ozadje

Kriptografija, veda o skrivanju sporočil, je eno od najbolj zanimivih področij matematike in računalništva.

Ta pisava niti ni preveč skrivna. Če bi imeli daljše besedilo, bi že s preštevanjem pogostosti znakov razvozlali, kateri znak predstavlja katero črko – najbolj pogost bi bil E, naslednji A ... in po tem, ko bi določili nekaj najbolj pogostih znakov, bi ostale brez težav uganili.

Zakaj se naloga imenuje Prostozidarska šifra? To je eno od imen, pod katerimi je znana ta pisava. Imenujejo jo tudi Napoleonova, ali pisava 3-v-vrsto (zakaj, je očitno, ne?). Več o njej si lahko prebereš na (žal samo angleški) strani na Wikipediji: https://en.wikipedia.org/wiki/Pigpen_cipher.

Sporočilca

Violeta želi s pomočjo prijateljev poslati Levu sporočilo. Zapiše ga na kartice, na katerih je prostora le za 3 črke, in jih razdeli prijateljem, da jih odnesejo Levu. Ker ve, da so različno hitri, kartice oštevilči, da lahko Lev hitro ugotovi, kakšno je pravilno zaporedje. Sporočilo ČASZAPLES pošlje takole:

1	2	3
ČAS	ZAP	LES

Kaj pa želi sporočiti z naslednjimi listki? _____

3	5	2	1	4
API	TKE	DIN	PRI	ŠKO

Rešitev

Pridi na piškotke.

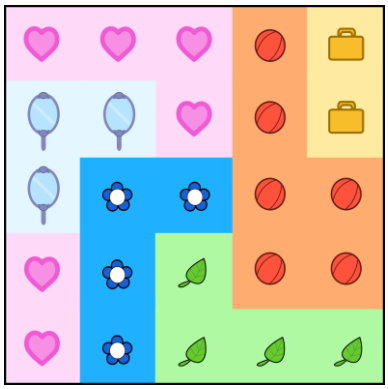
Računalniško ozadje

Tudi ko naprave komunicirajo po internetu, si navadno ne pošljejo celotnega sporočila – recimo datoteke ali pa vsebine spletne strani – v enem samem dolgem sporočilu, temveč ga razdelijo v manjše paketke. Običajna dolžina takega paketa je 1500 bajtov. Ker paketi potujejo po različnih poteh, ki niso nujno enako hitre, se lahko med seboj prehitevajo. Zato so oštevilčeni, tako da jih lahko njihov prejemnik zloži v pravilni vrstni red.

Francijeve nalepke

3.–5. razred

Bober Franci je nalepil šest pravokotnih nalepk različnih velikosti z različnimi vzorci. Lepil jih je eno čez drugo. V katerem zaporedju jih je nalepil na papir na desni?



A. 

B. 

C. 

D. 

Rešitev

Pravilni odgovor je D.

Nalepka s kovčki je edina, ki ni prekrita s katero od drugih nalepk, torej je bila nameščena zadnja. Nalepka s kovčki prekriva nalepko z žogami, nalepka z žogami pa nalepko z listi. Nalepka z listi prekriva nalepko z rožicami, nalepka z rožicami pa nalepko z ogledali. Nalepka z ogledali prekriva nalepko s srčki.

Računalniško ozadje

Zaporedje določenih nalog točno določa končni izdelek in zaporedje ukazov, ki jih damo računalniku, točno določa rešitev problema. Če bi enake nalepke nameščali v drugačnem zaporedju, bi dobili popolnoma drugačno sliko. Podobno bi se zgodilo, če bi spremenili zaporedje ukazov računalniku – dobili bi nekaj popolnoma drugega (lahko tudi nič).

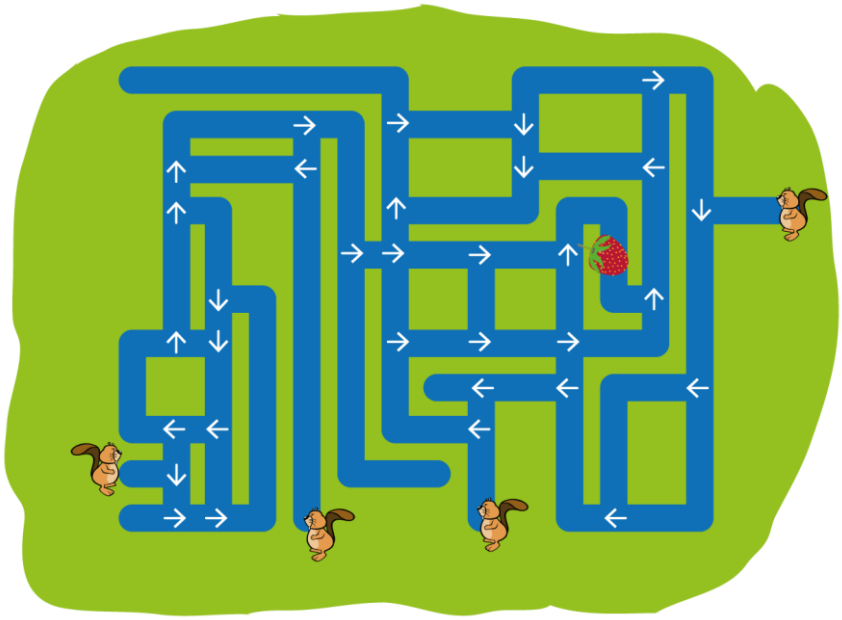
Ta naloga nas spominja tudi na plasti (angleško *layers*), ki se pogosto uporabljajo v programih za urejanje slik.

Lov na jagodo

4.–5. razred

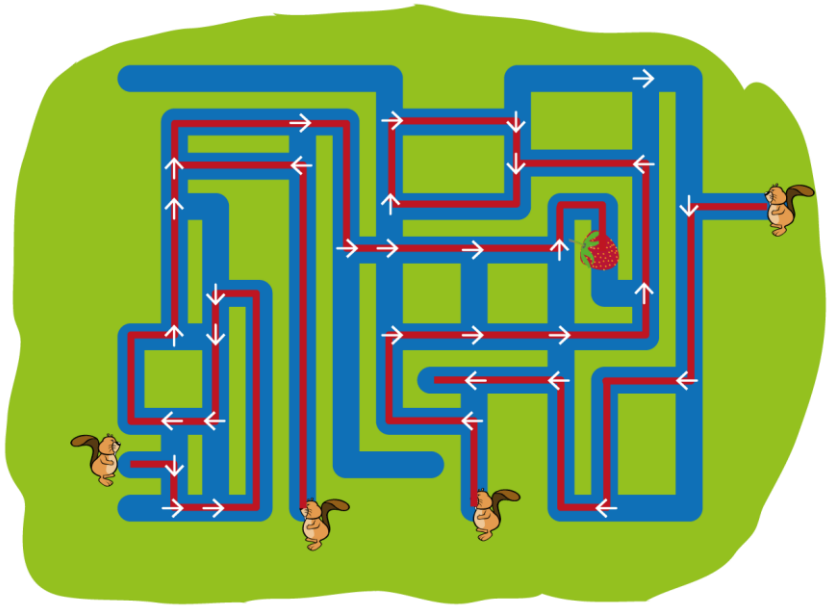
Štirje bobri skočijo v vodo na različnih mestih. Na vsakem križišču zavijejo v smer, ki jo kaže puščica.

Koliko bobrov bo priplavalo do jagode?



Rešitev

Na sliki so narisane poti vseh štirih bobrov. Le bobra na levi priplavata do jagode. Eden od ostalih zaide v slepo ulico, eden pa bo plaval v krogu.



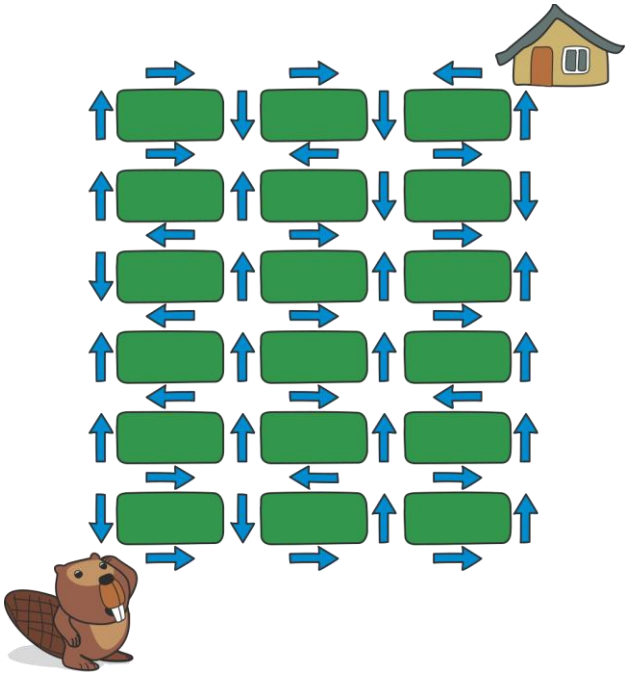
Računalniško ozadje

Tole je kot program – in bobri kot računalniki, ki prejmejo nov ukaz na vsakem križišču.

Pot domov

4.–5. razred

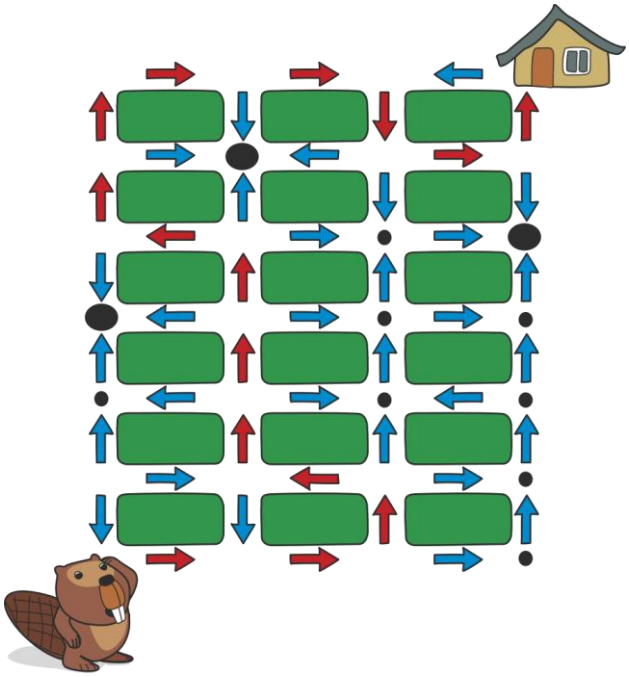
Bober Jani se lahko premika samo v smeri puščic. Označi pot, po kateri lahko pride domov.



Rešitev

Hitro lahko odkrijemo »črne luknje« (na spodnji sliki velike črne pike), v katere lahko pridemo, ne moremo pa iz njih. Naletimo lahko tudi na majhne črne pike, iz katerih lahko nadaljujemo samo do velikih črnih pik. Tem predelom se moramo izogniti, če želimo prispeti do cilja.

Od izhodišča do doma tako vodi samo ena pot. Enako rešitev dobimo, če se reševanja naloge lotimo v obratni smeri – od doma do bobra.



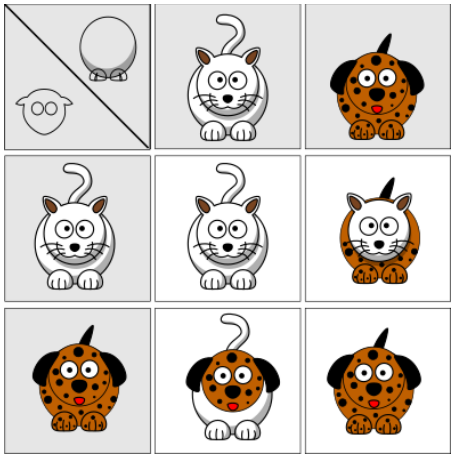
Računalniško ozadje

Iskanje poti je pogost problem v računalništvu.

Zmešane živali (težja)

4.–5. razred

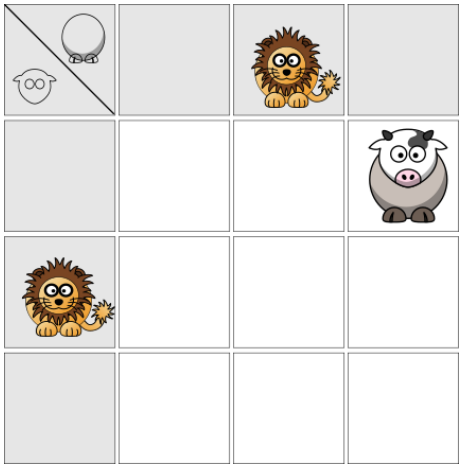
V deželi Čira čara imajo mladiče različne vrste živali. Mladiček ima očetovo glavo in mamino telo. Slika kaže mladiče mačkov in psov ter mačk in psic.



Kdo pa so starši teh živali?

Koze, krave in levi!

Dve živali sta že vrisani v tabelo na desni. Povleci puščice od ostalih štirih do njihovih mest.



Rešitev

Vse živali v isti vrsti morajo imeti enako glavo. Vse živali v istem stolpcu morajo imeti enako telo. Živali v celicah po diagonali imajo starše iste vrste in zato niso mešanci.

Računalniško ozadje

Tabele so eden od pomembnih načinov shranjevanja podatkov v računalniku. Vse, kar postavimo v isto vrstico ali stolpec, si je v nekih lastnostih podobno.

Barvanje ograje

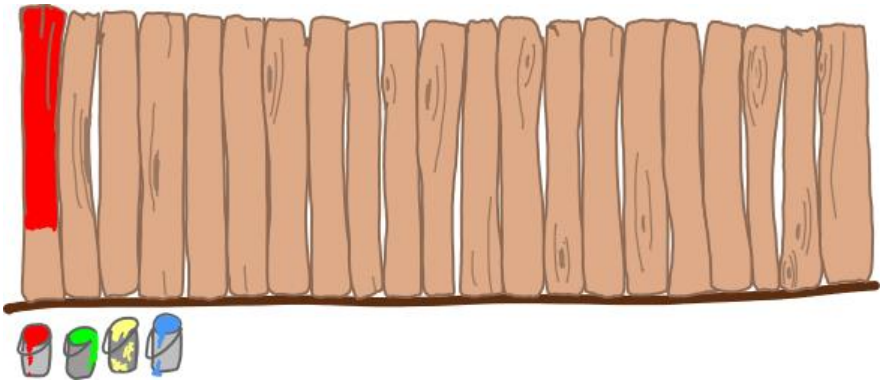
4.–7. razred

Tomaž barva ograjo z barvami iz štirih različno polnih posod. Barve jemlje po vrsti. Z vsako pobarva eno deščico, nato nadaljuje z naslednjo barvo. Če se posoda izprazni, jo odstrani iz vrste. Barvati bo nehal, ko bi moral dve zaporedni deščici pobarvati z isto barvo.

Posode imajo v začetku takšne količine barv:

- rdeča: za 5 deščic,
- zelena: za 3 deščice,
- rumena: za 7 deščic,
- modra: za 2 deščici.

Ograja je dolga 21 deščic. Koliko deščic bo pobarval?



Rešitev

Nalogo lahko očitno rešiš tako, da se sam lotiš barvanja. Najprej pobarva 🟥🟩🟨🟦🟥🟩🟨🟦. Zmanjka mu modre; rdeče ima še za 3 deščice, zelene za 1 in rumene za 5. Nadaljuje z 🟥🟩🟨, po čemer je brez zelene, rdeče ima še za dve deščici in rumene za štiri. Doda še 🟥🟨🟥🟨 in ostane brez rdeče. Ker mu ostane le še rumena – prav takšna pa je zadnja deščica – neha z barvanjem. Pobarval je 15 deščic.

Hitreje pa nalogo rešimo z razmislekom. Najprej mu bo zmanjkalo modre barve, nato zelene in na koncu rdeče. Po zadnji, peti rdeči deščici bo pobarval še rumeno deščico in končal barvanje. Vse barve bi bilo dovolj za 5 + 3 + 7 + 2 = 17 deščic. Ker pa mu bo ostalo za 2 deščici rumene barve, jih bo pobarval 15.

Računalniško ozadje

Za reševanje naloge moraš dobro razumeti navodila in jim slediti. Ko pregledujemo program in poskušamo ugotoviti, kako deluje (ali zakaj ne deluje), počnemo natančno isto.

Koliko je ura?

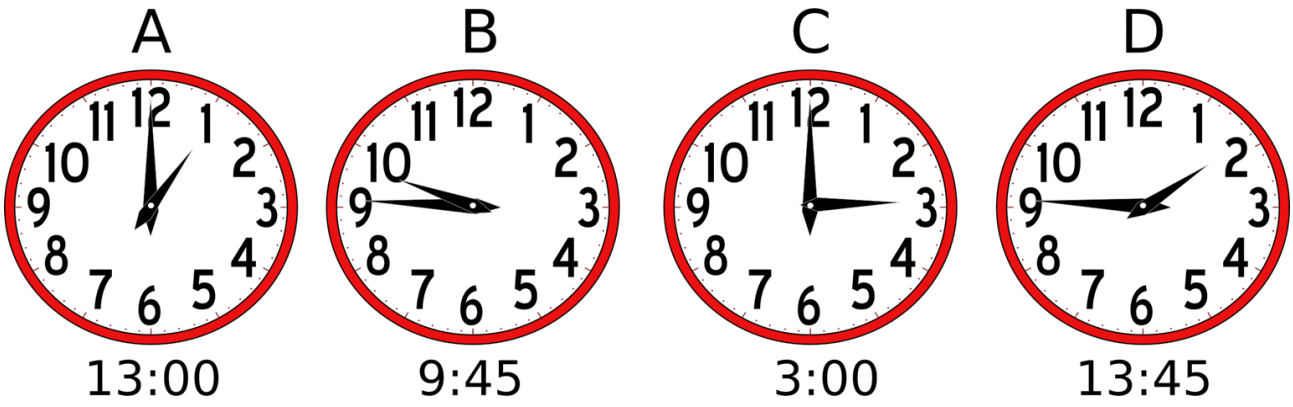
4.–7. razred

Bobri so zelo zaposleni. Niti na uro ne utegnejo pogledati! Koliko je ura, jim pove bitje zvona v mestnem stolpu.

- Ko je petnajst minut čez polno uro, zvon odbije enkrat.
- Ko do polne ure manjka pol ure, zvon odbije dvakrat.
- Ko je 45 minut čez polno uro, zvon odbije trikrat.
- Vsako polno uro zvon odbije štirikrat in še tolikokrat, kolikor je tedaj ura.

Ob petih, na primer, zvon odbije devetkrat.

Pred manj kot eno uro je zvon odbil 13-krat, ravnokar pa je odbil trikrat. Koliko je torej ura?



Rešitev

Pravilni odgovor je B.

Ko zvon bije 13-krat, je ura 9:00 (13 – 4 = 9). Če je od tedaj minila manj kot ena ura in zdaj zvoni trikrat, je ura 9:45.

Računalniško ozadje

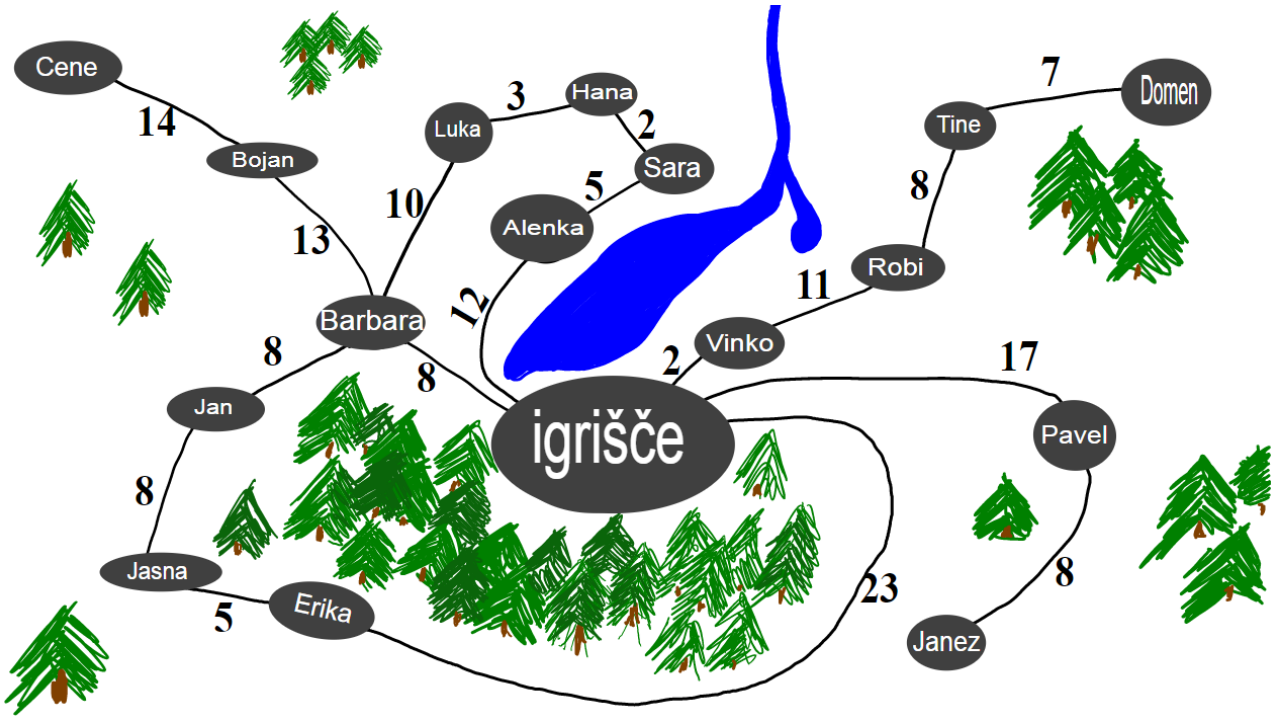
Bitje zvona je neke vrste kodiranje, v tem primeru trenutnega časa. Zvon se oglasi v točno določenih časovnih intervalih (na petnajst minut) in s številom udarcev posreduje sporočilo. Z dogovorom (standardom) smo določili, kaj pomeni posamezno število udarcev: najprej zaslišimo od enega do štiri udarce, vsak udarec pomeni četrt ure čez polno uro. Ko zaslišimo štiri udarce, vemo, da lahko pričakujemo še nadaljnje udarce, ki pomenijo polno uro, torej jih je lahko od enega do dvanajst.

Takšno sporočanje ure je bilo potrebno predvsem, ko ljudje še niso imeli ročnih ur in mobilnih telefonov, da bi lahko vedno vedeli, koliko je ura. Naloga prihaja iz Španije, a tudi zvonovi slovenskih cerkva še danes redno, podnevi in ponoči, na prav tak način sporočajo, koliko je ura.

Na igrišče

4.–7. razred

Peter se je odločil, da bo svoj rojstni dan praznoval na igrišču. Zemljevid kaže dolžine poti med naselji, v katerih živijo Petrovi prijatelji. Hana, recimo, je od igrišča oddaljena 2 + 5 + 12 = 19 km (če gre po najkrajši možni poti).



Na Petrovo rojstnodnevno zabavo bodo prišli le tisti, ki so od igrišča oddaljeni manj kot 20 km. Koliko obiskovalcev bo imel?

Rešitev

Devet.

Nalogo lahko rešimo tako, da za vsakega od možnih obiskovalcev preverimo, kako daleč je od igrišča.

Lahko pa se je lotimo bolj sistematično. V začetku si zapišemo, kateri od prijateljev so neposredno povezani z igriščem. Ti bi *lahko* prišli na zabavo. Nato v vsakem koraku dodamo najbližjega prijatelja, ki še ni na seznamu teh, ki pridejo, na seznam teh, ki bi lahko prišli, pa dodamo vse njegove sosede, če niso že predaleč in še niso na seznamu.

seznam teh, ki pridejo	njihovi sosedje, ki še niso predaleč
	Vinko (2), Barbara (8), Alenka (12), Pavel (17) Erike pa za zdaj ne dodamo, saj je od igrišča oddaljena 23.
Vinko (2)	Barbara (8), Alenka (12), **Robi (13),** Pavel (17)
Barbara (8)	Alenka (12), Robi (13), **Jan (16),** Pavel (17), **Luka (18)**; Bojan pa ne, saj je oddaljen 8 + 13 = 21.
Alenka (12)	Robi (13), Jan (16), **Sara (17),** Pavel (17), Luka (18)
Robi (13)	Jan (16), Sara (17), Pavel (17), Luka (18); Tineta ne dodamo, saj je oddaljen 13 + 8 = 21.
Jan (16)	Sara (17), Pavel (17), Luka (18); Jasne ne dodamo, saj je oddaljena 16 + 8 = 24.
Sara (17)	Pavel (17), Luka (18), **Hana (19)**
Pavel (17)	Luka (18), Hana (19); Janeza ne dodamo, saj je oddaljen 17 + 8 = 25.
Luka (18)	Hana (19)
Hana (19)	

Računalniško ozadje

Na prvi način – z ročnim preštevanjem – bi nalogo reševali ljudje. Na drugi način jo rešuje računalnik. Težava pri prvem načinu je, da bi odpovedal, če bi imel Peter milijon prijateljev. Drugi pa dobro deluje tudi v tem primeru.

Pa ima res kdo milijon prijateljev? Najbrž ne. Vendar praktično enak problem rešujejo navigacijski programi, ko iščejo najboljšo pot med dvema krajema. V tem primeru je križišč med podanim parom krajev lahko ogromno, algoritem pa mora vseeno zelo hitro poiskati pot med njima.

Pumptrack

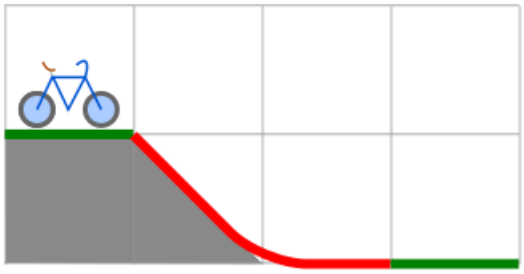
4.–7. razred

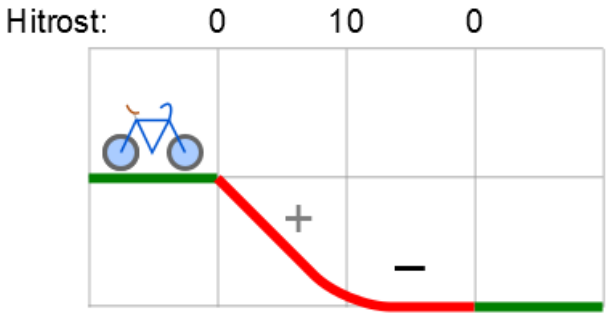
Pavel je prvič na pumptracku. Ker je previden, na klancih ne bo ne poganjal ne zaviral. Takole:

- na vsaki enoti spusta (en kvadratek) se mu hitrost poveča za 10 km/h;
- na vsaki enoti dviga se mu hitrost zmanjša za 10 km/h;
- na vsaki enoti ravnine lahko poganja in tako poveča hitrost za 10 km/h, zavira in jo tako zmanjša za 10 km/h, ali pa ne počne ničesar in obdrži trenutno hitrost.

Na začetku ima hitrost 0 km/h in se le odrine po prvem spustu. Ko na koncu proge pride čez zeleno ciljno črto, mora imeti spet hitrost 0.

Tule je skica preproste poti, ki jo lahko prevozi na ta način. Na spustu pospeši na 10 km/h, na ravnini pa zavira in se tako do zelenega dela ravno ustavi.

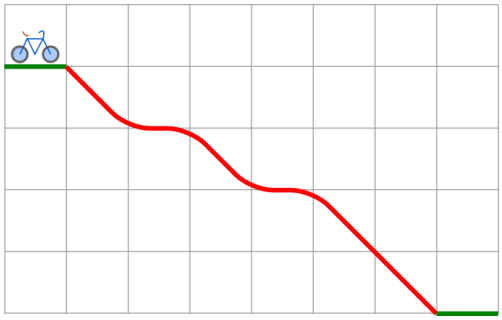




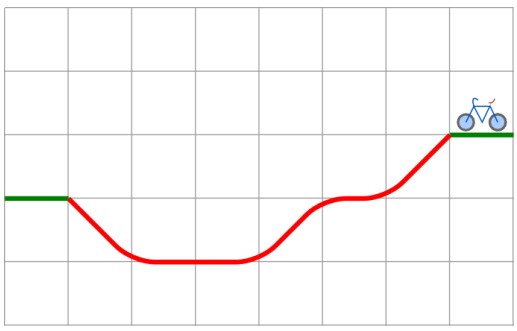
Na ta način lahko prevozi le eno od spodnjih prog. Katero?

A.

B.



C.



D.

Rešitev

V primeru A je ciljni klanec predolg in hitrost pade pod 0. V primerih B in D pripelje kolesar v cilj s previsoko hitrostjo. Primer C je pravilen.

Računalniško ozadje

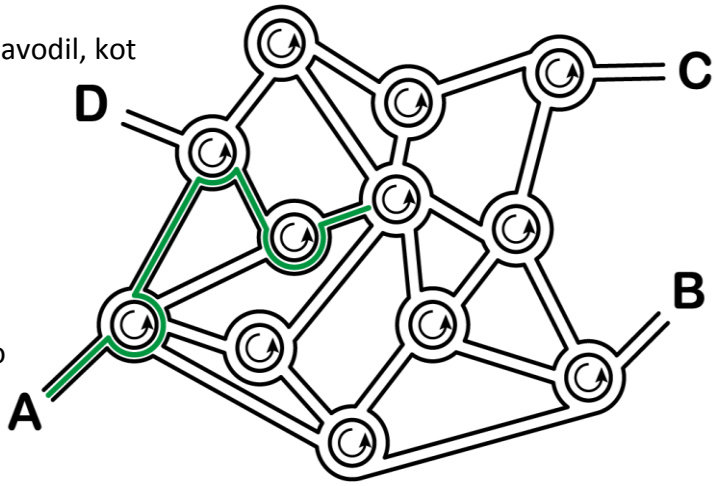
Dandanes ničesar – ne dirkalne steze, ne avtomobila, ne letala, ne zvoka v koncertni dvorani ne načrtujejo drugače, kot da najprej izračunajo številne simulacije s pomočjo računalnikov.

Mesto križišč

4.–9. razred

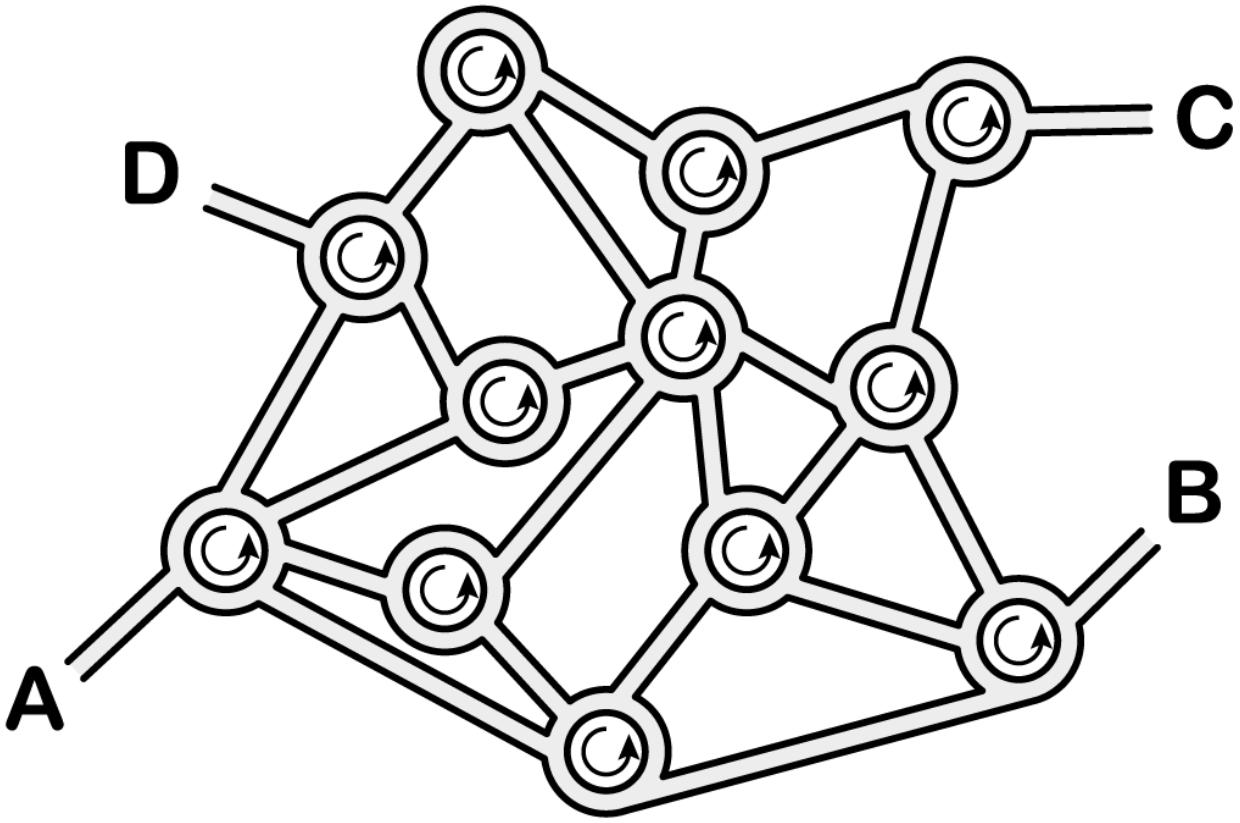
V mestu križišč avtomobilska navigacija ne daje navodil, kot

- na naslednjem križišču zavijte na četrti izvoz,
- na naslednjem križišču zavijte na prvi izvoz,
- na naslednjem križišču zavijte na drugi izvoz.



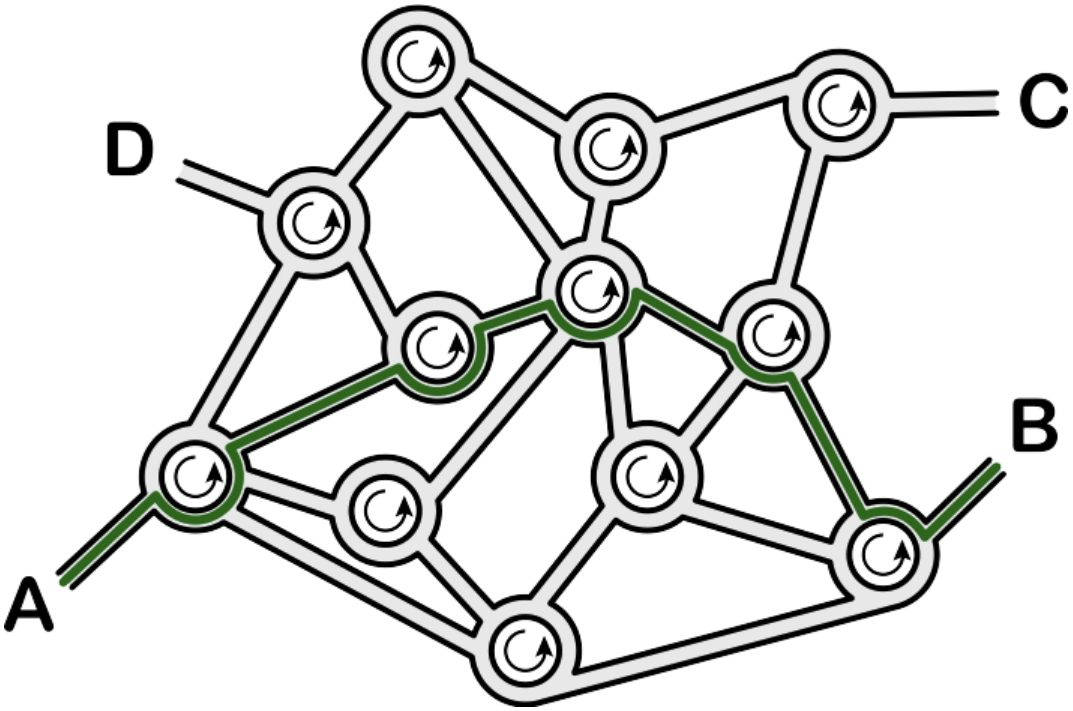
Namesto tega pove kar zaporedje izvozov. Če smo v A in so navodila 4 1 2, bomo peljali, kot kaže slika.

Pri katerem izvozu bomo končali, če začnemo pri A in namesto gornjemu zaporedju sledimo zaporedju izvozov 3 1 3 2 3?



Rešitev

Pri izvozu B.



Računalniško ozadje

Zaporedje števil je kot preprost program, ki ga izvaja voznik avtomobila.

Preste

4.–9. razred

V pekarni v dveh minutah spečejo tri preste. Vsak kupec lahko dobi naenkrat največ tri preste. Če jih želi kupiti več, se mora postaviti na konec vrste in počakati. V vrsti stojijo

- Ana, ki želi kupiti 7 prest,
- Berta, ki želi kupiti 3 preste,
- Cilka, ki želi kupiti 5 prest.

Čez dve minuti bo torej Ana dobila prve tri preste in se postavila na konec. Naslednja bo preste dobila Berta ... Čez koliko časa bo imela Cilka vseh pet prest?

Rešitev

Čez deset minut.

Čas		Čakajoči
Začetek		Ana (7), Berta (3), Cilka (5)
2 minuti	Ana dobi tri preste in gre na konec.	Berta (3), Cilka (5), Ana (še 4)
4 minuti	Berta dobi tri preste in odide.	Cilka (5), Ana (še 4)
6 minut	Cilka dobi tri preste in gre na konec.	Ana (še 4), Cilka (še 2)
8 minut	Ana dobi tri preste in gre na konec.	Cilka (še 2), Ana (še 1)
10 minut	Cilka dobi dve presti in odide. Tudi Ana dobi preostalo presto in odide.	

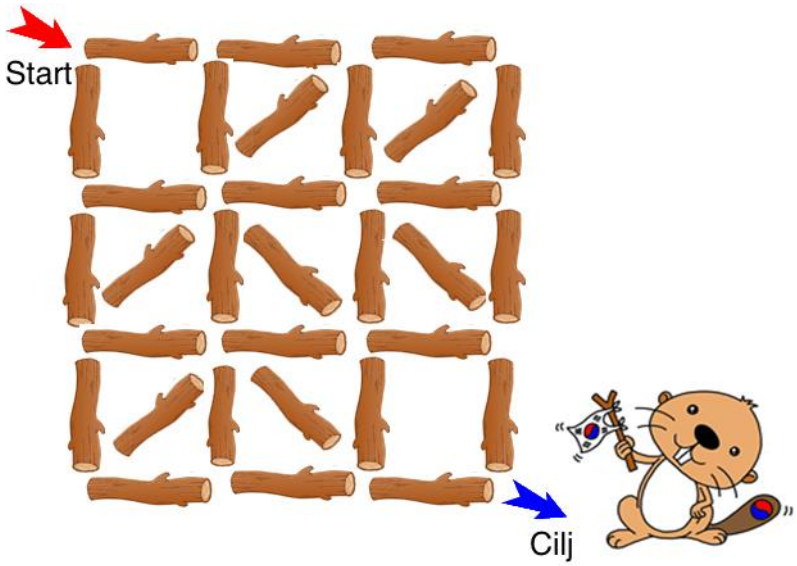
Računalniško ozadje

Kadar imaš hkrati odprtih deset programov, računalnik ne izvaja vseh hkrati, temveč v resnici zelo zelo hitro preklaplja med njimi. Vsakemu nameni droben delček sekunde in nato nadaljuje z naslednjim – podobno kot Ana, Berta in Cilka dobijo nekaj prest, nato pa morajo čakati, da pridejo spet na vrsto.

Tek po hlodih

4.–9. razred

Na rojstnodnevni zabavi je Peter pripravil igro s tekanjem po hlodih. Njegovi prijatelji morajo preteči pot od gornjega levega do spodnjega desnega kota tako, da tečejo po natančno petih hlodih. Na koliko različnih načinov je mogoče rešiti to nalogo?



Rešitev

Deset.

Najprej opazimo, da moramo teči po enem od hlodov, postavljenih diagonalno v pravi smeri (zgoraj levo – spodaj desno). Takšni so trije.

Ko to vemo, lahko pozorno preverimo vse možne poti in pazimo, da nobene ne izpustimo. Lahko pa jih preštejemo z naslednjim razmislekom.

- Do diagonalnega hloda na desni lahko pridemo na tri načine, od njega do cilja pa na enega samega. Prek diagonalnega hloda na desni torej vodijo tri različne poti.
- Do diagonalnega hloda na sredi pridemo na dva načina, od njega do cilja pa vodita spet dva. Dvakrat dva je štiri; prek srednjega hloda torej vodijo štiri poti.
- Do spodnjega diagonalnega hloda lahko pridemo na tri načine, od njega do cilja pa na en način.

Skupaj imamo torej 3 + 4 + 3 = 10 načinov.

Računalniško ozadje

Sistematičen pristop k reševanju problemov je natančno veščina, ki se je naučimo ob računalništvu.

Žaba Petra in žabec Toni

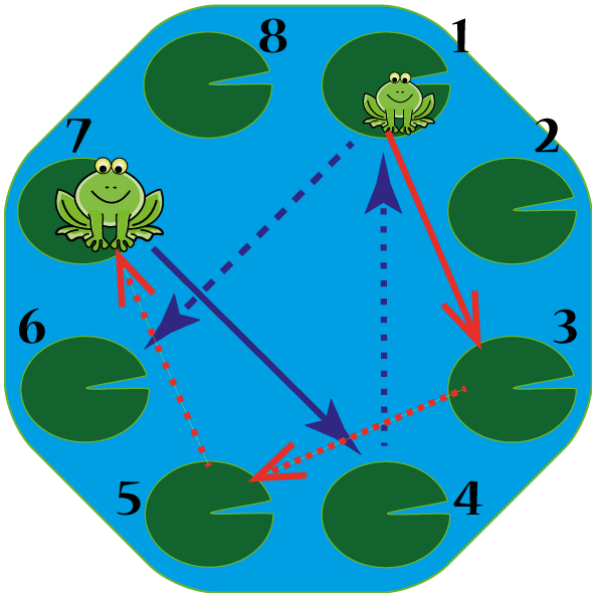
4.–9. razred

Velika žaba Petra hoče obiskati malega žabca Tonija. Proti njemu gre v smeri, nasprotni urinemu kazalcu. Ker je velika, preskakuje po dva lista, tako da skoči na vsakega tretjega: z lista 7 na 4, s 4 na 1 ...

Mali žabec Toni skače v smeri urinega kazalca proti Petri. Ker je majhen, lahko preskoči le en list in torej skače dva lista daleč. Z 1 na 3, s 3 na 5 ...

Oba skačeta istočasno. Ko se Toni čez tri skoke znajde na listu 7, je Petra ravno na listu 6.

Na katerem listu se bosta končno srečala?



Rešitev

Na listu 5.

Petra	7	4	1	6	3	8	5
Toni	1	3	5	7	1	3	5

Nalogo najlažje rešiš tako, da z dvema prstoma slediš žabama. Ali pa vzameš kar dve figurici, če nalogo rešuješ na listu.

Računalniško ozadje

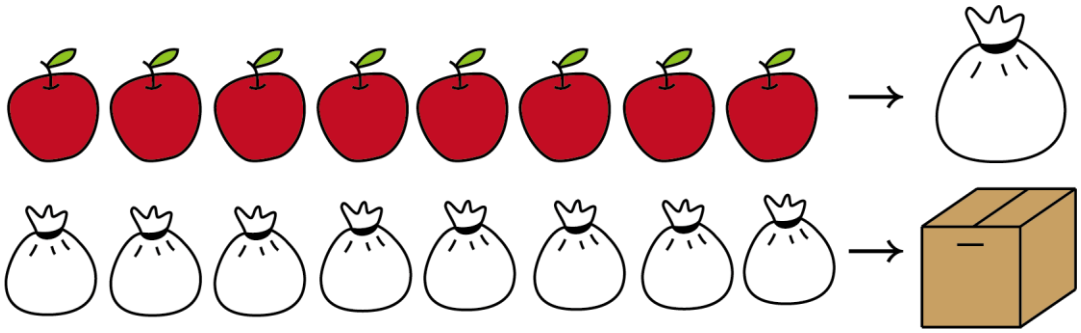
Programer mora pogosto slediti poteku programa – predvsem, kadar ta ne deluje, kakor bi moral. To ni tako zelo drugače od tega, kar si moral početi pri tej nalogi ti, ko si sledil Petri in Toniju. Matematiki pa bi v tem prepoznali modularno aritmetiko in diofantske enačbe ali kaj podobno učenega.

Pakiranje jabolk

6.–7. razred

Bobri pakirajo jabolka za prodajo.

- V vsako vrečo dajo natančno osem jabolk. Če vreče ne morejo napolniti do konca, preostalih jabolk ne dajo v vrečo.
- V vsako škatlo dajo osem vreč. Če škatle ne morejo napolniti do konca, preostalih vreč ne dajo v škatlo.



Zložiti morajo 275 jabolk. Koliko škatel bodo napolnili, koliko vreč bo ostalo izven škatel in koliko jabolk bo ostalo izven vreč?

Rešitev

Štiri škatle, dve vreči in tri jabolka.

V škatlo gre 8 vreč z 8 jabolki, torej 64 jabolk. 257 moramo torej deliti s 64: polne bodo štiri škatle, v katerih bo 256 jabolk.

Kaj bo z ostalimi 275 – 256 = 19 jabolki? Teh 19 jabolk zadošča za dve vreči, tri jabolka pa ostanejo.

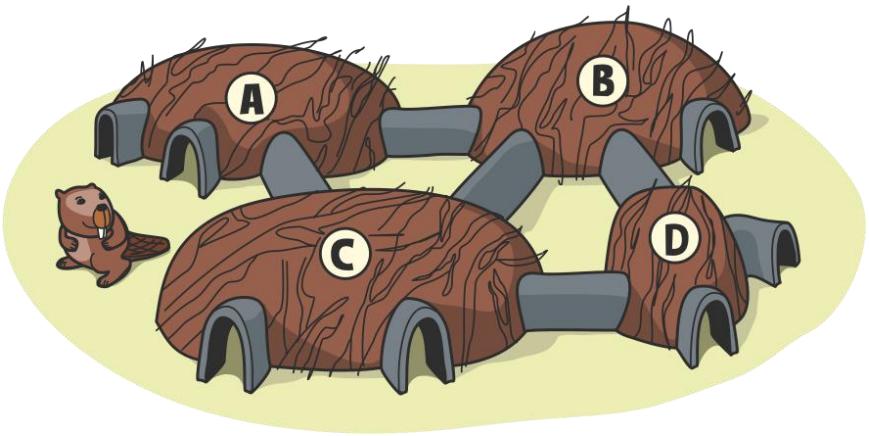
Računalniško ozadje

V računalništvu pogosto uporabljamo dvojiški zapis. Tako kot ima desetiški zapis enice, desetice, stotice, tisočice in tako naprej, ima dvojiški zapis enice, dvojice, štirice, osmice, šestnajstice ... V tej nalogi pa smo število 275 napisali po osmiško: osmiški zapis ima namreč enice, osmice, štiriinšestdesetice in tako naprej.

Osmiški zapis se v računalništvu včasih uporablja kot okrajšava za dvojiškega: ena osmiška števka ustreza trem dvojiškim števkam oziroma trem bitom. Kar prepričajmo se: število 275 bi po dvojiško zapisali kot 100010011 (le preveri, da je to res 256 + 16 + 2 + 1). Če jemljemo po tri bite skupaj, dobimo 100 010 011. Vsako trojko pretvorimo v desetiški zapis: 100 je 4, 010 je 2 in 011 je 3. To pa je ravno odgovor na vprašanje iz naloge: štiri škatle, dve vreči in tri jabolka.

Tek med prostori

6.–9. razred



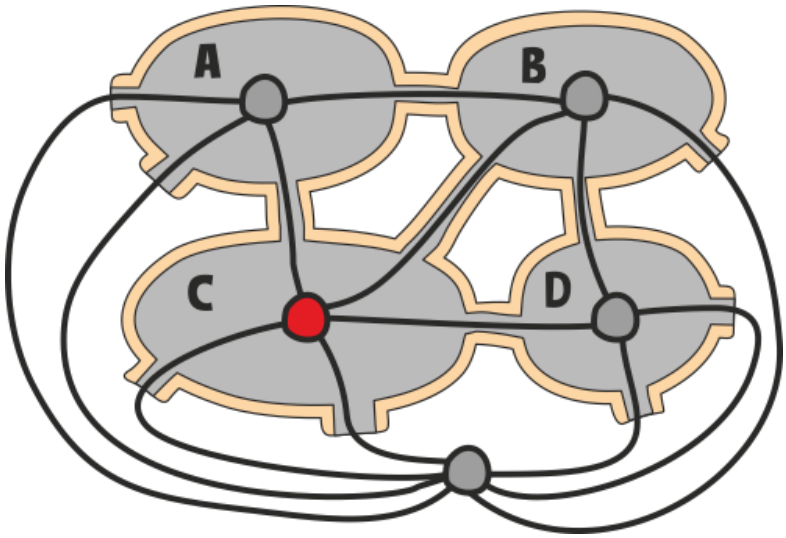
Družina bobrov je zgradila bobrišče s 4 sobami, ki so med seboj povezane s 5 tuneli in imajo 7 zunanjih vhodov.

Bobrčki so med igro opazili, da lahko tečejo po bobrišču tako, da prečkajo vse tunele in vhode v hiši natanko enkrat.

Če bobrček začne tak tek zunaj, kje ga bo končal?

Zunaj? V sobi A? V sobi B? Morda C? D? Ali pa lahko vemo, da le v eni od sob A ali B? Morda A ali C? A ali D, B ali C, B ali D, C ali D? Ali pa ne moremo o tem zagotovo reči ničesar?

Rešitev



V sobi C. Sobe A, B in D imajo 4 vrata. Ko bobrček prvič pride v eno od njih, odide skozi ena od preostalih treh vrat. Ko vstopi drugič, bo prišel skozi tretja vrata, odšel skozi zadnja, četrta in v to sobo se ne bo več vrnil.

Soba C pa ima liho število vrat, zato bo šel enkrat skoznjo, drugič pa bo v njej ostal. Če bobrček začne svoj tek zunaj, ga bo končal v sobi C.

Računalniško ozadje

Poti, ki gre po vsaki povezavi natančno enkrat, pravimo Eulerjev sprehod. Ime je dobil po švicarskem matematiku Leonhardu Eulerju, ki se sicer morda ni veliko sprehajal, pač pa je veliko prispeval k matematični teoriji grafov, ki se ukvarja s tovrstnimi problemi in je pomembna tudi za računalništvo.

Krajši zapis

6.–9. razred

Zaporedje

lahko krajše opišemo kot

5 6 1 2 4.

Kako je videti zaporedje, ki je opisano z

4 2 1 1 3 2?

A. 

B. 

C. 

D. 

Rešitev

B.

Prva sličica pove, s čim se zaporedje začne. Začne se z bobrom. Bobrov je 5, sledi 6 dreves, 1 bober, 2 drevesi, 4 bobri.

Zaporedje iz vprašanja se začne z drevesom. Imamo 4 drevesa, 2 bobra ... Pravzaprav sploh ni potrebno gledati naprej. B je edino zaporedje, ki se začne s štirimi drevesi.

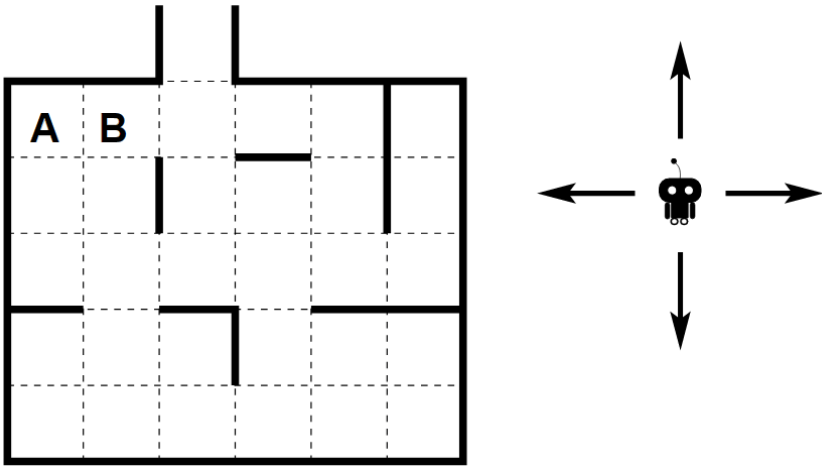
Računalniško ozadje

Takšen način skrajševanja zapisov se včasih uporablja v datotekah PDF, ki vsebujejo črno-bele slike. V črno-belih slikah se izmenjujejo zaporedja črnih in belih pik, tako kot se tu izmenjujejo bobri in drevesa.

Labirint

6. – 9. razred, državno

Marko je naredil robota in ga postavil v manjši labirint z neprehodnimi stenami – označujejo jih debelejše črte na sliki. Črtkane črte na sliki pa označujejo mrežo, ki določa možne pozicije robota.



Marko poda robotu zaporedje ukazov. Na ukaz se robot odzove s premikanjem v eno od smeri, ki jih nakazujejo puščice. Robot se premika, dokler ne pride do stene (ali pa ven iz labirinta). Ko se robot začne premikati, ga ne moremo več ustaviti in mu podati drugega ukaza, dokler se sam ne ustavi ob steni.

Če je na primer robot v kvadratu A, se lahko premakne štiri kvadrate v desno ali pa dva kvadrata navzdol, ne more pa se začeti premikati in se ustaviti na sredi poti, na primer v kvadratu B.

Marko želi ugotoviti, ali njegov robot lahko pride iz labirinta, če mu poda primerno zaporedje ukazov. To ga zanima za dve različni začetni poziciji v labirintu, to sta kvadrata A in B.

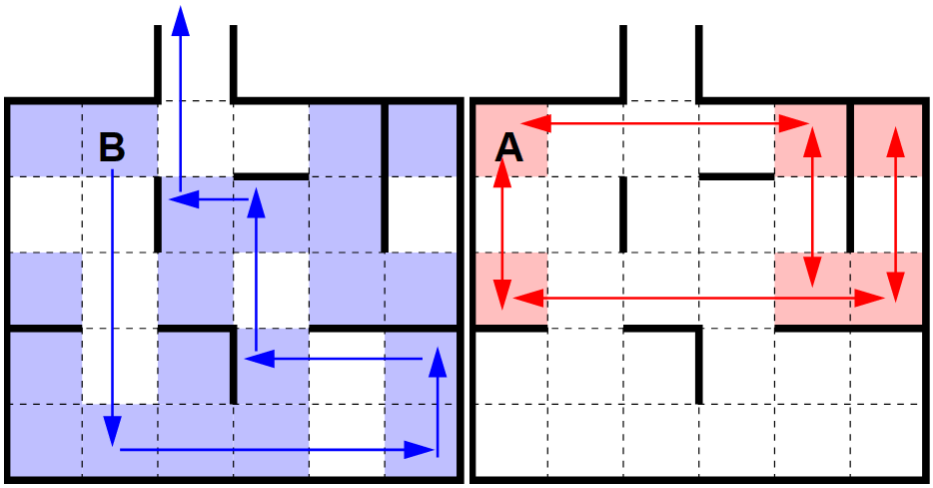
Katera od spodnjih trditev drži?

A. Robot lahko pride iz labirinta z obeh začetnih pozicij A in B.
B. Robot lahko pride iz labirinta z začetne pozicije A, ne pa tudi z B.
C. Robot lahko pride iz labirinta z začetne pozicije B, ne pa tudi z A.
D. Robot ne more priti iz labirinta z nobene od začetnih pozicij A ali B.

Rešitev

Pravilen odgovor je C: robot lahko pride iz labirinta, če začne na poziciji B, a ne z začetne pozicije A.

Rešitev je prikazana na slikah.



Modre puščice na levi sliki prikazujejo premike z začetne pozicije B, ki vodijo do izhoda iz labirinta (torej zaporedje ukazov, ki uspešno pripelje robota iz labirinta). Modro obarvani kvadrati prikazujejo vse pozicije, kjer se robot lahko ustavi.

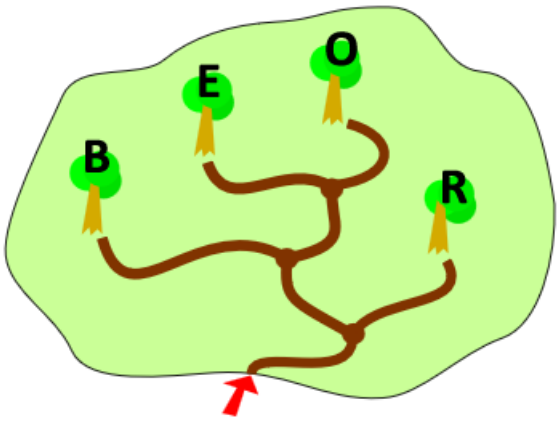
Podobno so na desni sliki prikazane rdeče puščice, ki pomenijo vse možne premike robota z začetne pozicije A. Tudi na tej sliki so rdeče obarvani kvadrati tisti, kjer se robot lahko ustavi.

Računalniško ozadje

Programiranje robotov je dober uvod v programiranje, saj lahko jasno vidimo, kaj se zgodi na vsakem koraku. Principi pa so enaki kot pri drugih vrstah programiranja: vsak ukaz v računalniškem programu na nek način spremeni »stanje« računalnika, naš cilj pa je doseči želeno stanje, na primer dobiti rezultat izračuna ali prikazati sliko na zaslonu.

Levo-desno

6. – 9. razred, državno



Za kodiranje svojih sporočil uporabljajo bobri kodo, ki temelji na zemljevidu njihovega osrednjega parka (glej sliko). Vsako drevo v parku je označeno z eno črko. Kodo za vsako črko dobimo tako, da sledimo poti od vhoda v park do ustreznega drevesa, na vsakem razpotju pa zavijemo levo (L) ali desno (D).

Tako ima na primer črka B kodo LL, saj moramo dvakrat zaviti levo, če želimo priti od vhoda v park do drevesa B. Kodo za besedo pa sestavimo iz kod za posamezne črke. Koda za besedo OBE je tako LDDLLLDL.

Katero besedo smo zakodirali s kodo LLLDDD?

Rešitev

Odgovor je BOR.

V tabeli smo zapisali kode levo-desno za vse črke na drevesih v parku:

B	**E**	**O**	**R**
LL	LDL	LDD	D

Če želimo odkodirati besedo LLLDDD, poiščemo najprej prvo zakodirano črko in njeno kodo. Ker nobena črka nima kode L, mora biti prva črka zakodirana z najmanj dvema znakoma in edina možnost je LL, torej je to črka B. Podobno odkodiramo še preostanek besede, to je LDDD. Druga črka je zakodirana v LDD, torej je to črka O. Ostane nam še koda D, v katero se zakodira črka R. Iskana beseda je torej BOR.

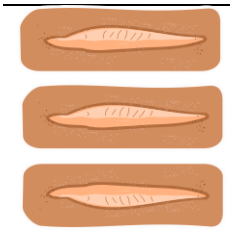
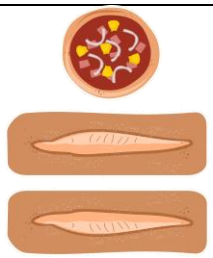
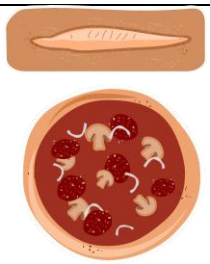
Računalniško ozadje

Če bi črko L zamenjali z 0 in črko R z 1, bi dobili dvojiški zapis. V tem primeru se zemljevid parka spremeni v nekaj, kar računalnikarji imenujemo *binarno drevo*. Tako lahko dolgo in zapleteno pot zelo enostavno shranimo v računalniku in pri tem porabimo tudi le malo prostora.

Pri tej kodi je zanimivo dejstvo, da nam za uspešno dekodiranje sporočila ni potrebno vedeti, kje se začne oz. konča koda posamezne črke (torej ne potrebujemo v kodi nobenih presledkov ali posebnih oznak, posledično pa je zakodirana beseda lahko še krajša). To je zato, ker smo kode izbrali tako, da se nobena koda ne začne s katero koli drugo kodo. Tako kodo imenujemo *prefiksna koda* in se v računalništvu pogosto uporablja.

Picerija

V piceriji imajo majhno peč za pico, zato lahko picopek hkrati speče le nekaj jedi. Spodaj lahko vidimo vse možne kombinacije in čas peke.

		
3 ciabatte	1 majhna pica in 2 ciabatti	1 ciabatta in 1 velika pica

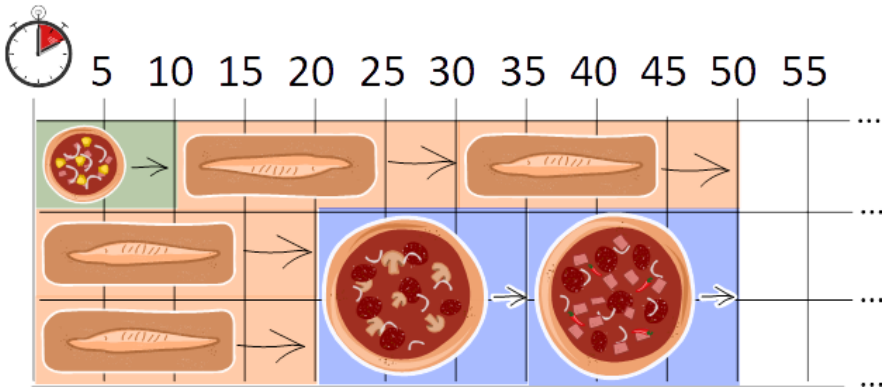
Čas peke:

Majhna pica	10 minut
Velika pica	15 minut
Ciabatta	20 minut

V piceriji imajo veliko gostov, zato mora picopek pametno načrtovati peko, da gostje lahko dobijo hrano, kolikor hitro je to mogoče. Ciabatte in pice lahko položi v peč v poljubnem vrstnem redu. Vsaka posamezna jed mora ostati v peči, dokler ni do konca pečena.

Gostje naročijo eno majhno pico, dve veliki pici in štiri ciabatte. Najmanj koliko časa bo minilo, preden bo celotno naročilo pečeno?

Rešitev



Obstaja več enako dobrih rešitev. Ključno je, da ugotovimo, da traja enako dolgo, če zaporedoma spečemo dve ciabatti in malo pico ali če zaporedoma spečemo dve veliki pici in dve ciabatti, ki se pečeta istočasno. Za oboje je potrebnih 50 minut, čas in prostor v peči sta tako polno zasedena, zato hitrejše rešitve ni. Ena od možnih rešitev je prikazana na sliki.

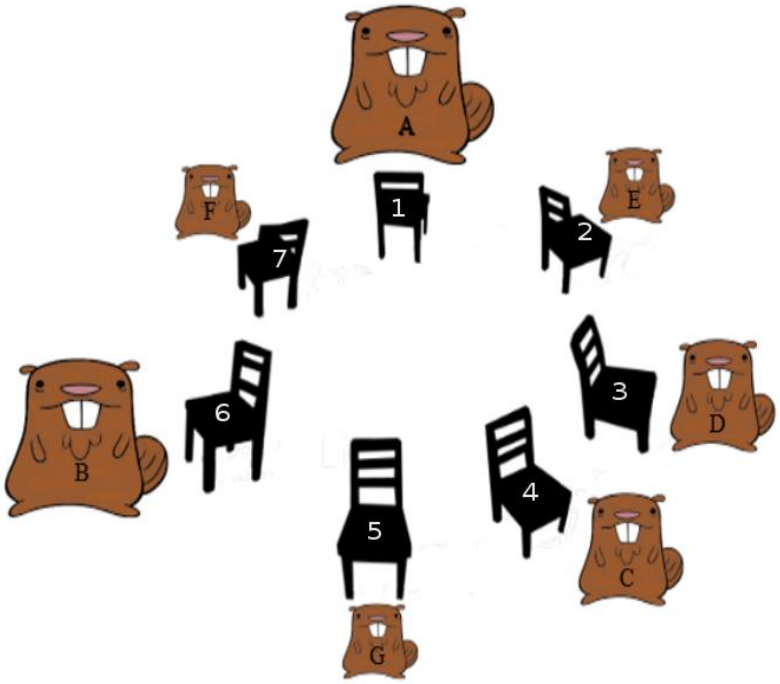
Računalniško ozadje

V računalniku ves čas poteka razvrščanje procesov. Picopekovo nalogo v računalniku prevzame razvrščevalnik.

Presedanje

6. – 9. razred, državno

Bobri se postavijo vsak k svojemu stolu, kot kaže slika.



Nato se začnejo premikati:

- velika bobra se v vsakem koraku igre pomakneta za tri stole v nasprotni smeri urinega kazalca,
- srednja bobra se v vsakem koraku premakneta za dva stola v nasprotni smeri urinega kazalca,
- mali bobri se premikajo na sosednji stol v smeri urinega kazalca.

Na istem stolu lahko sedi tudi več bobrov hkrati.

Naredili bodo tri korake igre. Katera stola bosta tedaj prosta?

Rešitev

2 in 7.

Velika bobra se premakneta za 9 korakov. A se premakne z 1 na **6** in B se premakne s 6 na **4**.

Srednja bobra se premakneta za 6 korakov: C s 4 na **5** in D s 3 na **4**.

Mali bobri se premaknejo za 3 korake: E z 2 na **5**, F s 7 na **3**, G s 5 na **1**.

Zasedeni so torej stoli 1, 3, 4, 5 in 6. Prosta sta 2 in 7.

Računalniško ozadje

Nalogo bi lahko reševali tako, da bi opazovali, kaj se dogaja po vsakem koraku. A to bi bilo težje. Če opazimo, da se vsak bober giba neodvisno od ostalih, lahko namesto posameznih korakov opazujemo posamezne bobre.

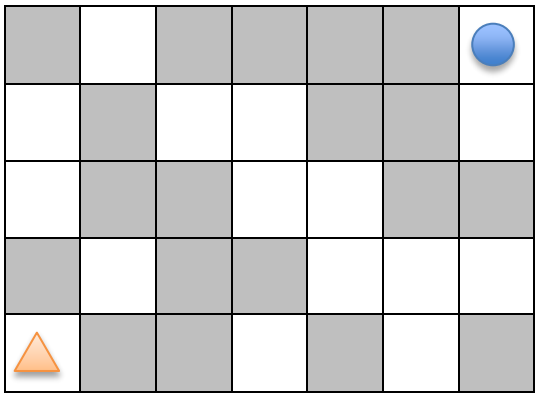
Računalniških problemov se je potrebno lotiti na pravi način, pa je rešitev preprostejša.

Rušenje zidov

6. – 9. razred, državno

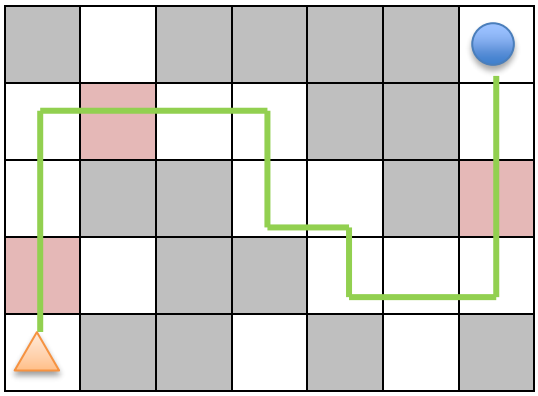
Labirint je sestavljen iz praznih polj (beli kvadrati) in zidov (sivi kvadrati).

Premikamo se lahko le po praznih poljih, ki so navpično ali vodoravno povezana med seboj.



Najmanj koliko zidov je potrebno porušiti, da se lahko po labirintu iz spodnjega levega kota sprehodimo v zgornji desni kot?

Rešitev



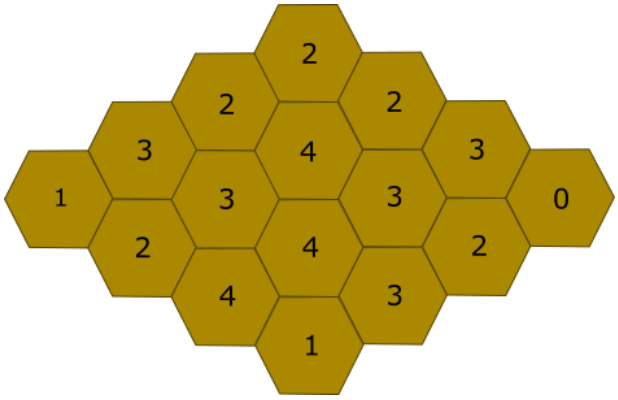
Zidovi, ki jih je potrebno porušiti, so označeni z rdečo barvo.

Računalniško ozadje

Za sistematično rešitev naloge je potrebno algoritmično razmišljanje. Za vsako polje je potrebno pogledati, koliko zidov je potrebno porušiti, da bi prišli do posameznega prostega polja.

Satovje

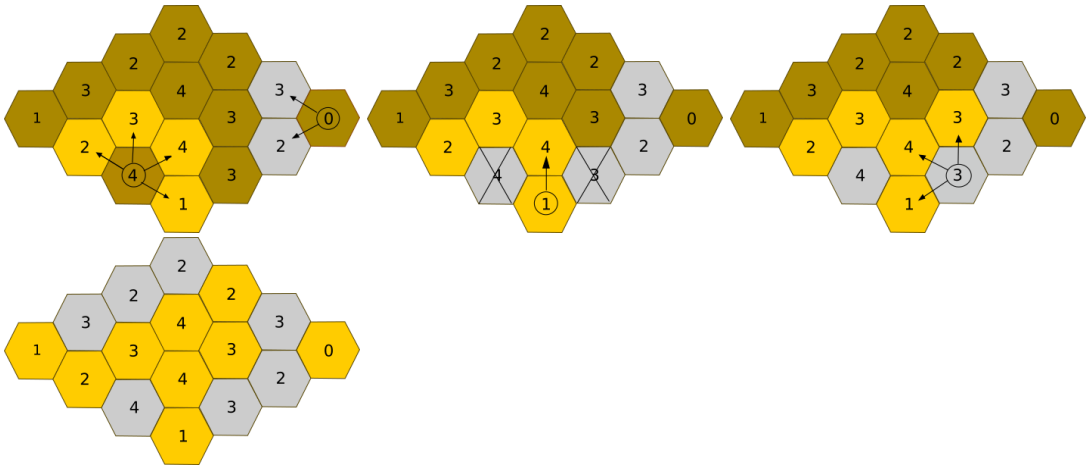
Manca ima rada med. Dedek ji je prinesel satovje. Nekateri deli so polni, nekateri prazni. Na vsakem delu je zapisano, koliko izmed sosednjih delov je polnih.



Koliko delov je polnih?

Rešitev

Začni pri delu, ki ima vse sosednje dele polne, in delu, ki ima vse sosednje dele prazne. Potem lahko postopno ugotoviš, kateri deli so še polni in kateri prazni. Polnih je 9 delov.

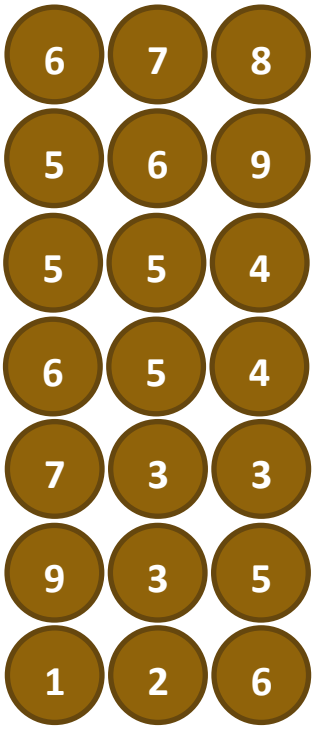


Računalniško ozadje

Da si lahko rešil nalogo, si si moral izmisliti primeren postopek reševanja. Računalnikarji temu rečejo *algoritem*.

Skakanje po štorih

6. – 9. razred, državno



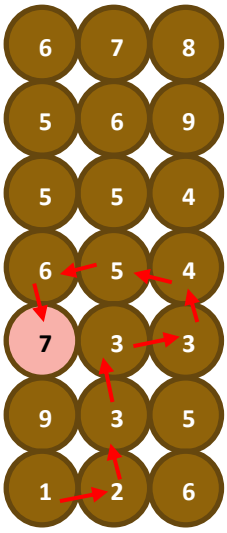
Katja se na Bobrovem vrtu igra na drevesnih štorih. Z vsakega štora opazuje štiri sosednje štore (spredaj, zadaj, levo in desno) in sledi tem trem pravilom:

1. Če je eden od sosednjih štorov za ena višji od tistega, na katerem stoji, skoči nanj.
2. Če ne, skoči na štor, ki je enako visok, ampak samo, če na njem še ni bila.
3. Če se s trenutnega štora ne more premakniti po prejšnjih dveh pravilih, se ustavi.

Višine in mesta, na katerih so posamezni štori, so na sliki.

Katja začne na štoru z višino 1. Kako visoko bo priskakala?

Rešitev



Računalniško ozadje

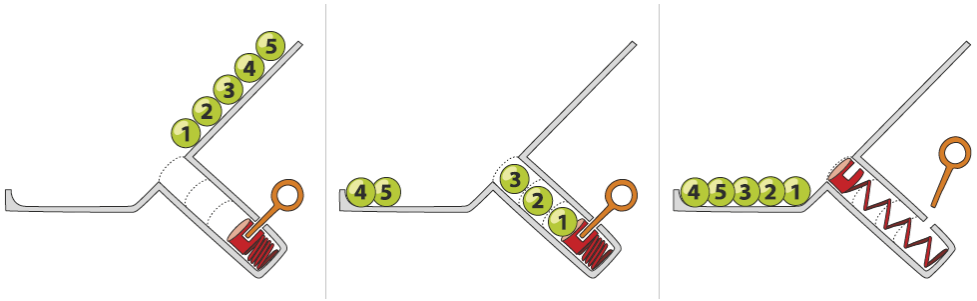
Vsak vidi, da obstaja pot, po kateri lahko pridemo do štora z višino 9 zgoraj desno, Katja pa gleda le en korak naprej in ne celotne slike, zato tega ne ve. V računalništvu marsikdaj napišemo program, ki najde rešitev na podoben način, tako da upošteva le en korak naenkrat. Take algoritme imenujemo požrešni algoritmi, saj požrešno izbirajo tisto, kar je v posameznem koraku videti kot najboljša rešitev. A kot vidimo, to marsikdaj ne pripelje do optimalnega rezultata.

Zakaj potem to počnemo? Zakaj programi ne pogledajo »celotne slike«? Ker je število možnih poti, ki bi jih morali preiskati, lahko tako veliko, da noben računalnik ne bo nikoli uspel pregledati vseh. Zato se moramo zadovoljiti s postopkom, ki včasih da le "dovolj dobro" rešitev.

Žogice

6. – 9. razred, državno

Oštevilčene žogice spustimo po klančini. Žogice lahko padejo v luknje na klančini in tako se vrstni red žogic spremeni. Ko žogica pride do luknje, pade vanjo, če je še dovolj prostora. Če prostora v luknji ni, se žogica odkotali dalje. Vsaka luknja ima na dnu tudi zatič; če ga odstranimo, sprožimo vzmet, ki vrže žogice iz luknje. Poglejmo primer:

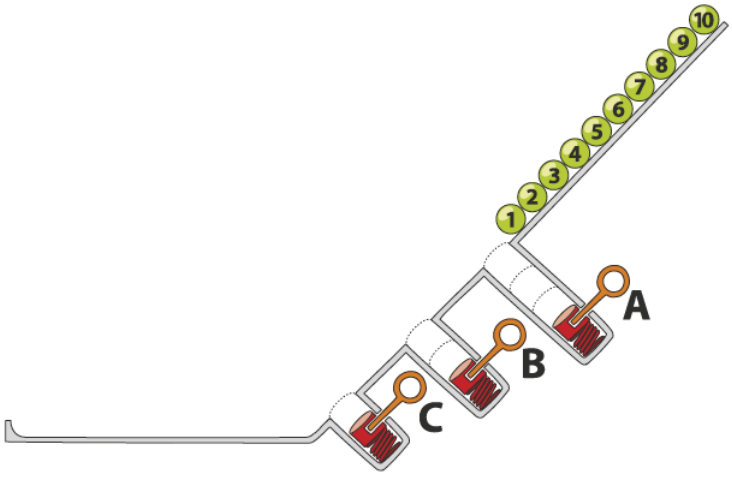


stanje, preden pet žogic spustimo po rampi

ko se žogice ustavijo

končno stanje, potem ko smo odstranili zatič

Po klančini spustimo deset žogic. Na klančini so tri luknje, A, B in C, v katerih je prostora za 3, 2 in 1 žogico, po vrsti (glej spodnjo sliko). Zatiče odstranjujemo po vrsti, najprej A, nato B in na koncu C, a vedno najprej počakamo, da se žogice nehajo kotaliti.



Kakšen bo končni vrstni red žogic?

A. 7 8 9 10 1 2 3 4 5 6

B. 7 8 9 10 1 2 3 5 4 6

C. 10 9 8 7 6 5 4 3 2 1

D. 7 8 9 10 3 2 1 5 4 6

Rešitev

Pravilen odgovor je D: 

Luknja A ima prostora za tri žogice, zato žogice od 1 do 3 po vrsti padejo vanjo, žogice od 4 do 10 po vrsti pa se odkotalijo mimo nje. Luknja B ima prostora za dve žogici, zato žogici 4 in 5 po vrsti padeta vanjo, žogice od 6 do 10 po vrsti pa se odkotalijo mimo nje. Luknja C ima prostor le za eno žogico, vanjo pade žogica 6, žogice od 7 do 10 po vrsti pa se odkotalijo mimo nje. Torej so prve štiri žogice na dnu tiste s številkami od 7 do 10 po vrsti. Nato odstranimo zatič pri A, kar izvrže žogice 3, 2 in 1 po vrsti. Te tri žogice se zakotalijo do dna, kjer imamo sedaj žogice 7, 8, 9, 10, 3, 2, 1. Nato odstranimo še zatič pri B ter izvržemo žogici 5 in 4, po vrsti. Tudi ti dve žogici se odkotalita do dna, kjer imamo sedaj naslednje zaporedje žogic: 7, 8, 9, 10, 3, 2, 1, 5, 4. Na koncu odstranimo še zatič C in žogica 6 se odkotali kot zadnja do dna. Zaporedje žogic na dnu je sedaj 7, 8, 9, 10, 3, 2, 1, 5, 4, 6, kar je tudi pravilen odgovor.

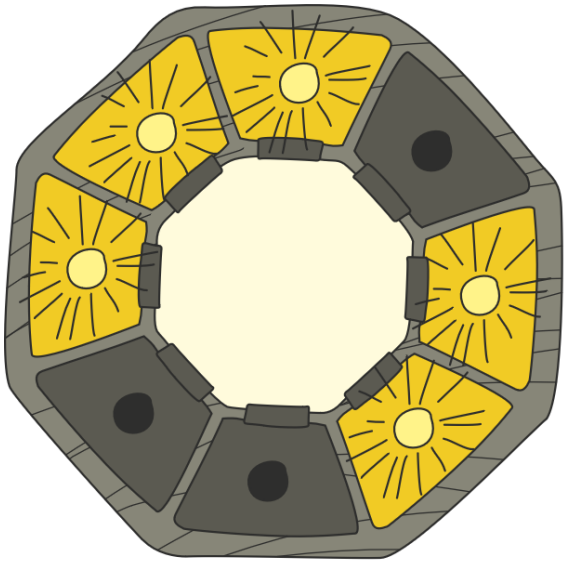
Računalniško ozadje

Luknja na klančini se obnaša kot *sklad*, ki je zelo uporabna *podatkovna struktura* (to je, način organiziranja podatkov). Sklad deluje po principu zadnji noter, prvi ven (angleško *last-in first-out* ali krajše *LIFO*). Prva žogica, ki je padla v luknjo, je bila iz nje izvržena zadnja. Čeprav je sama ideja sklada zelo preprosta, lahko sklad uporabimo v mnogih različnih situacijah.

Primer uporabe sklada je pri preverjanju ustrezno postavljenih oklepajev v izrazih: npr. izraz ((1+2)*3) ima pravilno postavljene oklepaje, izraz ((4+5)*(6-7) pa ne. Algoritem preverjanja ujemanja oklepajev je naslednji: vse predklepaje postavimo na sklad (operaciji dodajanja na sklad rečemo angleško *push*) in ko najdemo ujemajoči zaklepaj, s sklada odstranimo en predklepaj (operaciji odstranjevanja rečemo angleško *pop*). Če je na koncu sklad prazen in če smo lahko uspešno izvedli vse operacije pop, se oklepaji v izrazu ujemajo. Tako smo namesto kompliciranega algoritma za preverjanje ujemanja oklepajev uporabili le ustrezen način za organiziranje podatkov – zadostovala je le uporaba principa zadnji noter, prvi ven.

Luči v pisarnah

6.–9. razred, srednja šola



Osem programerjev ima vsak svojo pisarno. Ko so v pisarni in delajo, imajo prižgano luč. Ko odidejo, jo ugasnejo. Kot lahko vidimo na sliki, nekateri danes manjkajo.

Njihova šefica redno beleži, kdo je v pisarni in dela, tako da opazuje luči. Za to uporablja posebne simbole. Ne vemo niti, s katero pisarno je začela zapis, niti, v katero smer je popisovala pisarne.

Kateri od zapisov ustreza stanju na sliki?

A. # & & & # # & #
B. & & # & & # # &
C. & # & & & # & #
D. # & # # & & & &

Rešitev

Pravilni odgovor je B, to je & & # & & # # &.

Ker ne vemo, kateri znak pomeni prižgano luč in kateri ugasnjeno, predpostavimo, da znak # pomeni prižgano in znak & ugasnjeno luč. Torej mora pravilen odgovor vsebovati 5 znakov # in 3 znake &. Temu pogoju ne ustreza nobeden od podanih odgovorov. Torej naša predpostavka ne drži.

To pomeni, da prižgano luč označuje znak &, ugasnjeno luč pa znak #. Tako mora pravilen odgovor vsebovati 5 znakov & in 3 znake #. Odgovor A ima en # preveč (in en & premalo), zato ni pravilen. Pravilen odgovor ima tudi skupino treh znakov & in skupino dveh znakov &, med tema dvema skupinama pa morajo biti znaki #. Vidimo, da temu pogoju ustreza le odgovor B (odgovor C nima skupine dveh znakov &, odgovor D pa ima skupino štirih znakov &).

Računalniško ozadje

Računalniški strokovnjaki imajo radi binarna (dvojiška) števila. Binarno število je predstavljeno z uporabo dveh števk, 0 in 1. Vsaka števka torej predstavlja dve stanji, kot na primer prižgano in ugasnjeno luč v naši nalogi. Čeprav za predstavitev teh dveh stanj navadno uporabljamo števki 0 in 1, lahko v splošnem uporabimo poljubna dva simbola (kot sta & in # v naši nalogi). Način predstavitve navadno prepustimo osebi ali napravi, ki jo zapisuje.

Razpolavljanje ogrlice

6.–9. razred, srednja šola

Andreja ima ogrlico iz zaporedja perlic: diamant, kocka, kroglica, diamant, kocka, kroglica ... in tako dalje.

1. Andreja odstrani perlico na sredini ogrlice in tako dobi dva krajša, enako dolga dela ogrlice. Levi del ogrlice odloži.
2. Z desnim delom ponovi postopek: odstrani perlico na sredini in spet obdrži le desni del.
3. Preostali desni del razdeli na enak način in spet obdrži le desno polovico.
4. To ponovi še enkrat.

Ostane ji le ena perlica, v obliki diamanta. Katere perlice je odstranila z ogrlice in v kakšnem vrstnem redu?

A. Prva: , druga: , tretja: , četrta:

B. Prva: , druga: , tretja: , četrta:

C. Prva: , druga: , tretja: , četrta:

D. Prva: , druga: , tretja: , četrta:

Rešitev

Da bi rešil to nalogo, moraš najprej ugotoviti, koliko perlic je bilo na ogrlici na začetku. To narediš tako, da nazaj izračunaš, koliko perlic je bilo na ogrlici v posameznem koraku: na koncu imamo samo 1 perlico, torej smo imeli v prejšnjem koraku 3 ($2 \cdot 1 + 1$), v tretjem koraku smo jih imeli 7 ($2 \cdot 3 + 1$), v drugem 15 ($2 \cdot 7 + 1$) in na začetku 31 ($2 \cdot 15 + 1$).

V prvem koraku odstrani 16. perlico po vrsti, to je . V drugem koraku odstrani 24. perlico, to je . V tretjem koraku odstrani 28. perlico, to je . V četrtem koraku odstrani 30. perlico, to je .

Računalniško ozadje

Razpolavljanje in razpolavljanje in razpolavljanje ... je osnova mnogih pomembnih postopkov v računalništvu.

Drsalci

Sedem drsalcev drsa v vrsti po dolgem zamrznjenem kanalu. Začnejo v vrstnem redu, ki ga prikazuje slika.

Da se ne bi preveč utrudili, se vsaki dve minuti prvi drsalec pomakne na konec vrste.

Kateri drsalec bo spredaj v 59. minuti?

P Q R S T U V

Rešitev

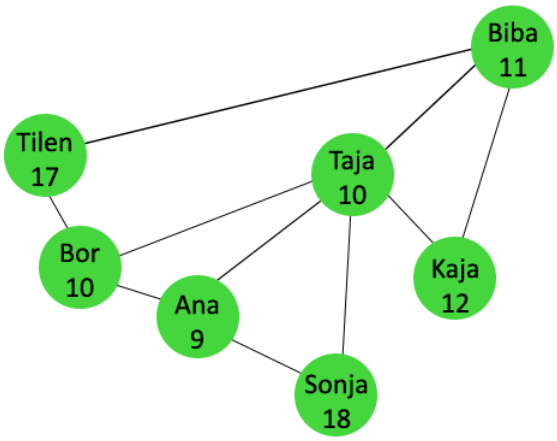
Sedem drsalcev se stalno izmenjuje v vodstvu, kar pomeni, da je vsakih 14 minut vrstni red enak kot na začetku. Drsalec V drsa kot prvi ob času 0, 14, 28, 42, 56. V 58. minuti gre drsalec V na konec, tako da je prvi drsalec v 59. minuti U.

Računalniško ozadje

Pri reševanju naloge smo računali ostanke po deljenju s 14. Takšno računanje, ki se mu učeno reče *modularna aritmetika*, je osnova mnogih postopkov v računalništvu. Predvsem je zanimiva njena uporaba v kriptografiji, vedi o šifriranju.

Knjižni klub

8.–9. razred, srednja šola



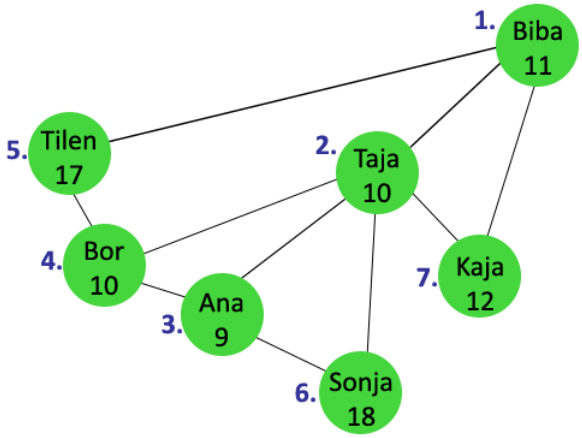
Slika kaže prijateljstva med sedmimi učenci, ki radi berejo. Zapisana so njihova imena in starosti, povezava pa nakazuje prijateljstvo.

Radi si tudi posojajo knjige. Kdor dobi knjigo, ki je še ni prebral, jo prebere, nato pa jo da svojemu najmlajšemu prijatelju, ki te knjige še ni prebral. Če so knjigo prebrali že vsi prijatelji, jo vrne tistemu prijatelju, ki mu jo je dal.

Biba je prebrala novo knjigo in jo želi deliti s prijatelji. Kdo bo zadnji prebral to knjigo?

Rešitev

Pravilen odgovor je: Kaja.



Biba da knjigo Taji, ki je njena najmlajša prijateljica. Ko Taja knjigo prebere, jo preda Ani. Ana jo da Boru, ta pa Tilnu. Tilen knjigo vrne Boru, Bor pa jo vrne Ani. Nato od Ane dobi knjigo Sonja in jo tudi vrne Ani. Ana jo vrne Taji, Taja pa da knjigo Kaji, ki je edina še ni prebrala.

Vrstni red, v katerem prijatelji berejo knjigo, je označen na sliki.

Računalniško ozadje

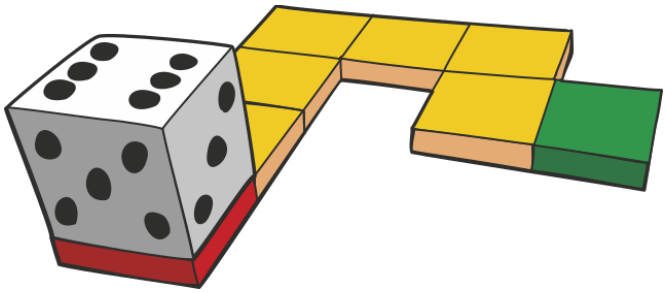
Veliko podatkov, s katerimi se srečujemo v računalništvu, je medsebojno povezanih. Naše najljubše družbeno omrežje lahko predstavimo kot množico elementov (osebe) in povezav (prijateljstva). Tudi cestno omrežje je množica elementov (kraji) in povezav (ceste). Elementi prehranjevalne verige so živa bitja in povezave povedo, kdo je plenilec in kdo plen.

Za analizo takih mrež obstaja veliko algoritmov, ki ovrednotijo različne lastnosti. Eden bolj znanih algoritmov je iskanje v globino. Z njim lahko preiščemo vse elemente grafa v določenem vrstnem redu. Če graf preiskujemo v globino iz enega elementa, lahko ugotovimo, katere druge elemente lahko dosežemo, če sledimo povezavam. Podobno lahko naredimo za vse ostale elemente.

Problem v nalogi zahteva, da sledimo knjigi skozi omrežje prijateljev. Pot, ki jo naredi knjiga skozi omrežje prijateljev, je zelo podobna vrstnemu redu preiskovanja grafa v globino.

Kocka

Janez premika kocko po poti, označeni s kvadrati. Za premik kocke z enega kvadrata na drugega Janez zvrne kocko čez rob, ki leži med obema kvadratoma. To ponovi sedemkrat, dokler kocka ne leži na zelenem kvadratu na desni.



Upoštevaj, da je vsota pik na nasprotnih si ploskvah vedno 7 (1 je nasproti 6, 2 je nasproti 5, 3 je nasproti 4). Na začetku kocka leži na ploskvi z eno piko (nasproti ploskve s 6 pikami), kot kaže slika. Po premiku kocke na naslednje polje leži na ploskvi z 2 pikama (nasproti ploskve s 5 pikami).

Koliko pik je na spodnji ploskvi kocke, ko kocka leži na zelenem kvadratu?

Rešitev

4

Da rešimo nalogo, si moramo predstavljati, kako se obrača kocka. Ni preprosto. Kocka leži na ploskvah v naslednjem vrstnem redu: 1, 2, 6, 5, 3, 2, 6, 4.

Nalogo smo pokazali matematiku Sašu, pa nam je razložil lep način, kako slediti obračanju kocke, ne da bi se zmotili. Morda se ti sprva ne bo zdel preprost, a poskusi se poglobiti in ga razumeti.

Kako je kocka obrnjena, opišemo s tremi številkami; recimo, da bomo najprej napisali številki, ki sta spredaj in desno, nato pa tisto, ki je zgoraj. Kocko na sliki bi torej opisali z (5, 3, 6). V splošnem pa imamo tri števila, na primer (a, b, c).

Kaj se zgodi, ko kocko prekucnemo nazaj? 5 gre na vrh, 3 ostane pri miru, spredaj pa se pojavi številka, ki je nasprotna 6, torej 7 – 6, se pravi 1. V splošnem se (a, b, c) spremeni v (7 – c, b, a).

Kaj se zgodi, ko jo prekucnemo desno? Če bi predtem stala, kot stoji zdaj, ostane 5 na mestu, 6 gre desno, zgoraj pa se pojavi, kar je nasprotno trojki, torej (5, 6, 4). V splošnem se (a, b, c) spremeni v (a, c, 7 – b).

Kaj se zgodi, ko jo prekucnemo nazaj? Če bi to storili s to kocko, bo 6 spredaj, 3 ostane desno, zgoraj pa se pojavi, kar je nasprotno 5 (torej 2). Dobimo torej (6, 3, 2). V splošnem se (a, b, c) spremeni v (c, b, 7 – a).

Zberimo vse na enem mestu:

Premik	Matematični zapis	In zapis po domače
nazaj:	(a, b, c) → (7 – c, b, a)	prvo številko na konec, na prvo mesto pa daj 7 – zadnjo
desno:	(a, b, c) → (a, c, 7 – b)	zadnjo številko daj na drugo mesto, na zadnje daj 7 – drugo
naprej:	(a, b, c) → (c, b, 7 – a)	zadnjo številko daj na prvo mesto, na zadnje daj 7 – prvo

Zdaj pa moramo izračunati, kaj naredi zaporedje nazaj, nazaj, nazaj, desno, desno, naprej, desno.

začetek (5, 3, 6)

nazaj (7 – 6, 3, 5) = (1, 3, 5)

nazaj (7 – 5, 3, 1) = (2, 3, 1)

nazaj (7 – 1, 3, 2) = (6, 3, 2)

desno (6, 2, 7 – 3) = (6, 2, 4)

desno (6, 4, 7 – 2) = (6, 4, 5)

naprej (5, 4, 7 – 6) = (5, 4, 1)

desno (5, 1, 7 – 4) = (5, 1, 3)

Zgoraj je 3, torej je spodaj 4.

Računalniško ozadje

To je primer problema, pri katerem moramo slediti izvajanju navodil, da pridemo do končne rešitve.

Z »matematičnim zapisom« pa je naloga še nekaj več: pokaže nam, kako lahko s sistematičnim, formalnim zapisom spremenimo nalogo v obliko, ki je abstraktnejša, vendar jo lažje rešimo, ne da bi se pri tem zmotili. Takšna rešitev je na prvi pogled težja, a le na prvega: ko jo razumemo, bi lahko na ta način hitro vrteli kocko po veliko daljših poteh, ne da bi si morali ob tem predstavljati, kaj se dogaja z njo. (Le še obrat na levo bi potrebovali. Ga znaš sam zapisati?)

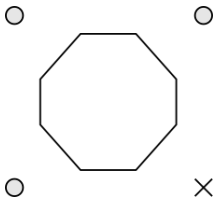
Kako pa bi napisali program, ki reši nalogo? Potrebovali bi natančno takšno »formalizacijo« problema, kot je tale tu. Saj res – znaš kaj programirati? Četudi »le« v Scratchu? Bi znal napisati program, ki reši to nalogo?

Mavrični stolp

8.–9. razred, srednja šola

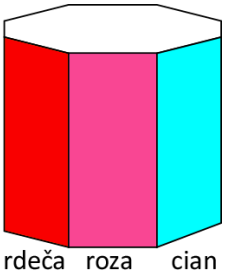
Sredi mesta stoji mavrični stolp. Ima osem stranic, vsaka je druge barve: cian, modra, zelena, oranžna, roza, rdeča, vijolična in rumena, ne nujno v tem vrstnem redu. Ana, Berta in Ciril stojijo na različnih točkah, označenih s krožci, ti pa stojiš na točki, ki je označena s križcem.

- Ana vidi cian, vijolično in rumeno steno.
- Berta vidi oranžno, modro in vijolično steno.
- Ciril pa vidi zeleno, rdečo in oranžno steno.

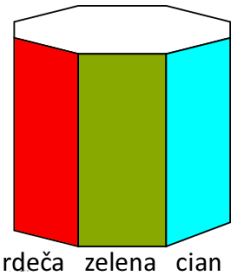


Navedene barve niso nujno v tem vrstnem redu. Katere barve stene vidiš ti?

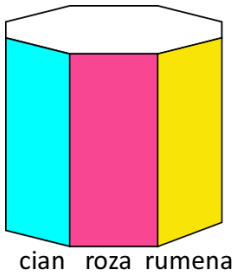
A.



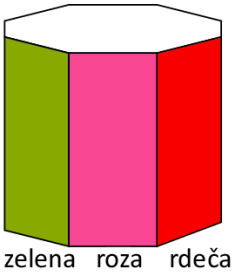
B.



C.



D.



Rešitev

Pravilna rešitev je A.

Ker nihče drug ne vidi roza stene, mora biti ta neposredno pred nami. To izloči odgovor B.

Pri odgovoru D vidimo hkrati zeleno in rdečo steno, obe barvi pa vidi tudi Ciril. Glede na obliko stolpa to ni možno, zato tudi odgovor D ni pravilen.

Podobno velja za odgovor C: če mi vidimo hkrati cian in rumeno, potem teh dveh barv ne more videti tudi Ana, ne glede na to, na katerem od treh označenih mest stoji.

Torej nam ostane le odgovor A, ki je edini pravilen. Berta stoji nasproti nas, med Ano in Cirilom (če ne bi stala na tem mestu, bi videla vsaj eno steno enake barve, kot jo vidimo mi). Ana stoji desno od nas, saj oba vidiva cianovo steno (ta je na naši desni). Ciril pa stoji levo od nas, saj oba vidiva rdečo steno. Barve sten stolpa pa so v naslednjem zaporedju (če gledamo v smeri urinega kazalca): roza, rdeča, zelena, oranžna, modra, vijolična, rumena, cian.

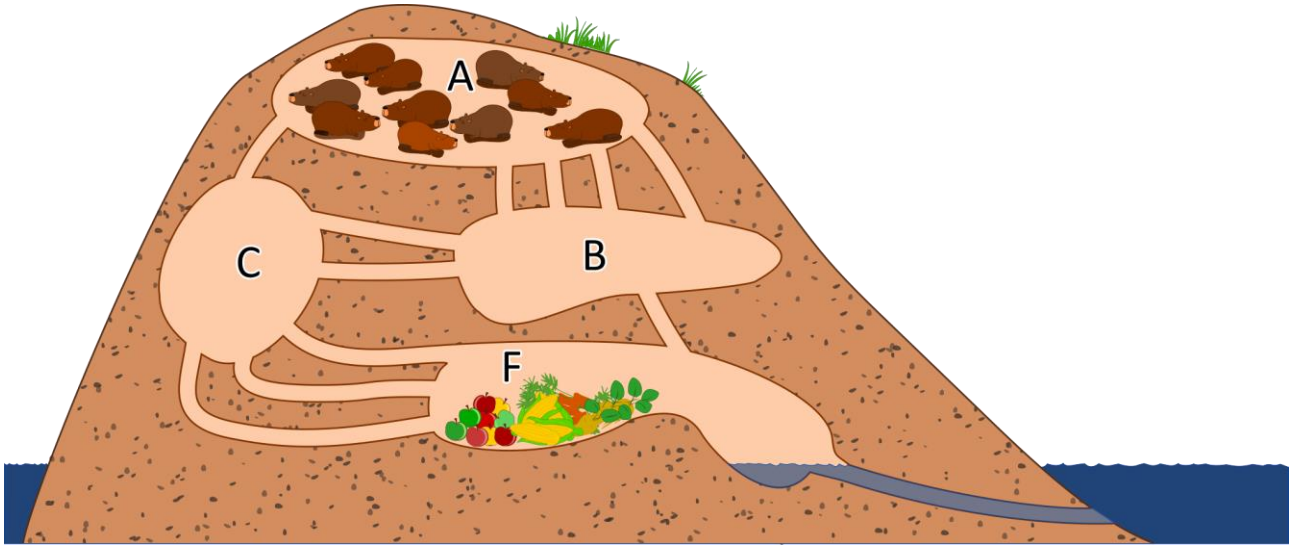
Računalniško ozadje

Rešitev smo poiskali s pomočjo omejitev (povezav med ljudmi in barvami), s katerimi smo izključevali nemogoče kombinacije. Obstaja zanimiv računalniški jezik, Prolog, ki mu za razliko od ostalih jezikov ne podamo postopka, temveč povezave, računalnik pa poišče možne rešitve.

Tuneli v bobrišču

8.–9. razred, srednja šola

V bobrišču so tuneli, ki povezujejo štiri sobe (A, B, C in F). Prve tri sobe (A, B in C) so sobe, v katerih živijo, četrta soba (F) je shramba.



Deset bobrov je v sobi A, postajajo lačni in želijo priti na malico v sobo F. Ker so vsi bobri zelo lačni, želijo vsi priti v shrambo, kolikor hitro se le da.

Bobri tunel prehodijo v 1 minuti. Hkrati je lahko v enem tunelu samo en bober.

Povezave med sobami so sestavljene iz določenega števila tunelov:

- med A in B so 4 tuneli,
- med A in C je 1 tunel,
- med B in C sta 2 tunela,
- med B in F je 1 tunel,
- med C in F so 3 tuneli.

V vsaki sobi je lahko poljubno število bobrov naenkrat.

Po koliko minutah bodo v najboljšem primeru vsi bobri v shrambi?

Rešitev

V najboljšem primeru bodo vsi bobri v shrambi po 4 minutah.

V bobrišču sta dve poti, po katerih lahko bober iz sobe A pride v sobo F v 2 minutah. To sta pot iz A preko B v F in pot iz A preko C v F. Preko vsake od teh dveh poti gre lahko hkrati le en bober.

V bobrišču je tudi pot, po kateri bober pride iz sobe A v sobo F v 3 minutah. To je pot iz sobe A preko B in C do F. Po tej poti gresta lahko hkrati 2 bobra.

Ker iz sobe A v sobo B vodijo 4 tuneli in iz sobe B proti sobi F le 3 tuneli, bo en bober obtičal v sobi B, če bodo uporabili vse 4 poti iz A v B hkrati. Povezave med B in C ter B in F so ozko grlo.

Spodnja tabela po minutah prikazuje premike in stanja v sobah po teh premikih:

Premiki in stanja	**Število bobrov v posamezni sobi (po premikih)**			
	A	**B**	**C**	**F**
Na začetku	**10**	**0**	**0**	**0**
3 bobri gredo iz A v B 1 bober gre iz A v C **Po 1 minuti**	**6**	**3**	**1**	**0**
1 bober gre iz B v F 1 bober gre iz C v F 2 bobra gresta iz B v C 3 bobri gredo iz A v B 1 bober gre iz A v C **Po 2 minutah**	**2**	**3**	**3**	**2**
1 bober gre iz B v F 3 bobri grejo iz C v F 2 bobra gresta iz B v C 1 bober gre iz A v B 1 bober gre iz A v C **Po 3 minutah**	**0**	**1**	**3**	**6**
1 bober gre iz B v F 3 bobri grejo iz C v F **Po 4 minutah**	**0**	**0**	**0**	**10**

Računalniško ozadje

V nalogi si reševal enega najbolj znanih računalniških problemov: problem iskanja največjega pretoka. Izrek, ki je enako slaven kot problem, pravi, da je največji možni pretok enak najmanjšemu prerezu. Najmanjši prerez je tak »prerez« grafa, ki najbolj omeji pretok. Če ga poiščemo, vemo tudi, kakšen je največji možni pretok.

Vdor v muzej

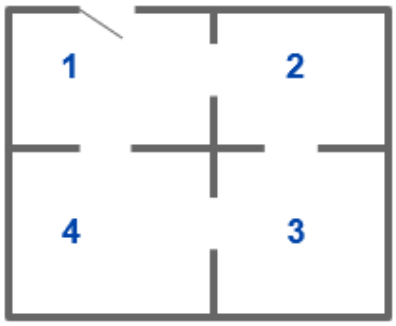
8.–9. razred, srednja šola

V muzeju sodobne bobrske drevesne umetnosti so namestili nov inteligentni varnostni sistem za odkrivanje vsiljivcev. Vsiljivec v muzeju je oseba, ki vstopi v muzej, a ne skozi glavni vhod.

Inteligentni varnostni sistem deluje tako, da vsakič, ko oseba vstopi ali zapusti prostor, zazna natančno število oseb v vsakem prostoru in podatke zabeleži v tabelo. Sistem vedno pravilno locira vsako osebo v en prostor v muzeju. V vsak prostor lahko vstopi, ali pa ga zapusti, tudi več oseb hkrati.

Tabela kaže zapis inteligentnega varnostnega sistema, slika pa kaže razpored prostorov v muzeju.

Čas	Soba 1	Soba 2	Soba 3	Soba 4
10:00	2	0	0	0
10:07	3	0	0	0
10:08	2	1	0	0
10:12	4	1	1	0
10:13	2	2	3	0
10:17	5	2	2	1
10:20	4	1	2	2



Ob kateri uri je sistem v muzeju zaznal vsiljivca?

Rešitev

Pravilen odgovor je ob 10:13.

Ob tem času sta v sobo 3 vstopili dve osebi, a v sosednji sobi (v sobi 2) je bila pred tem le ena oseba in soba 4 je bila prazna. Torej je moral nekdo v sobo 3 vstopiti od zunaj, ne preko glavnega vhoda.

Računalniško ozadje

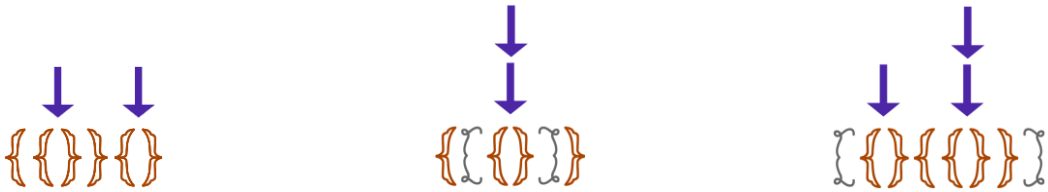
Varnostne sisteme, ki beležijo število oseb na določenih kritičnih območjih, lahko najdemo na primer na letališčih. Računalniški programi obdelujejo živo sliko iz kamere, na slikah izluščijo osebe ter jih preštejejo. Ti programi uporabljajo umetno inteligenco (na primer za prepoznavo ljudi), pa tudi zelo enostavna logična pravila, kot smo jih imeli v tej nalogi pri ugotavljanju vdorov v muzej.

Oklepajoči okraski

6. – 9. razred in srednja šola, državno

V zlatarni izdelujejo zapestnice, sestavljene iz parov okraskov v obliki oklepajev. Zapestnico izdelajo tako, da za osnovo vzamejo enega od naslednjih dveh parov okraskov:  ali .

Nato v katerikoli del zapestnice dodajo zaporedoma več parov okraskov, kot kažejo naslednji primeri:



Katero od zapestnic so z opisanim postopkom naredili v zlatarni?

A. 

B. 

C. 

D. 

Rešitev

Pravilen odgovor je D. Začeli so s parom okraskov, nato so v sredino vstavili še en par, naslednji par pa so vstavili še v sredino predhodnega para.

Vse ostale zapestnice ne ustrezajo opisani metodi. Pri odgovoru A je napačna pozicija tretjega okraska, saj je desni del prvega para okraskov pred desnim delom drugega para okraskov. Pri odgovoru B je napačna pozicija prvega okraska: zapestnica se začne z desnim delom para okraskov, namesto z levim delom, kar ni pravilno. Pri odgovoru C pa je napačna pozicija četrtega okraska; imamo tri leve dele enega para okraska in nato tri desne dele drugega para okraska, nikjer pa nimamo nobenih parov.

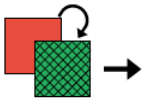
Računalniško ozadje

Pravila, ki jih uporabljajo v zlatarni pri izdelavi zapestnic so enaka pravilom, ki jih uporabljamo za oklepaje pri zapisu izrazov. Računalnikarji rečejo, da so izrazi s pravilno postavljenimi oklepaji dobro sestavljeni, tisti z napakami pa so slabo sestavljeni. Za dobro sestavljeni izraz rečemo, da je sintaktično pravilen, kar pomeni, da ustreza zahtevani sintaksi. Sintaktične napake v kompleksnih izrazih je zelo težko odkriti, čeprav je sintaktične napake praviloma lažje poiskati kot semantične napake, ki se nanašajo na logične napake programerja.

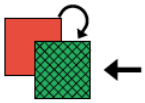
Robot

6. – 9. razred, srednja šola, državno

Milan je sestavil robota, ki bere barvne kvadrate, spremeni njihovo barvo ter se prestavi za en kvadrat na levo ali desno. Robot deluje po navodilih, kot sta na primer naslednji:



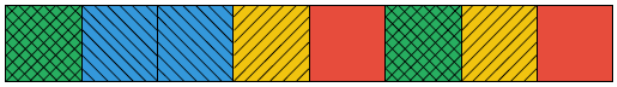
Če si na rdečem kvadratu, spremeni barvo kvadrata na zeleno in se premakni za en kvadrat v desno.



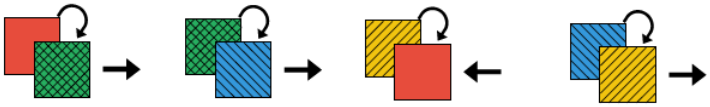
Če si na rdečem kvadratu, spremeni barvo kvadrata na zeleno in se premakni za en kvadrat v levo.

Na začetku stoji robot na najbolj levem kvadratu. Robot preveri barvo kvadrata, poišče pravilo, ki ustreza tej barvi, spremeni barvo kvadrata glede na pravilo ter se premakne, kot zahteva pravilo. V naslednjem koraku robot ponovi postopek za kvadrat, na katerem se nahaja. To ponavlja, dokler se ne ustavi. Robot se ustavi, če ne najde ustreznega pravila za kvadrat, na katerem se nahaja, ali če se premakne izven kvadratov.

Robotu smo podali naslednje zaporedje kvadratov:

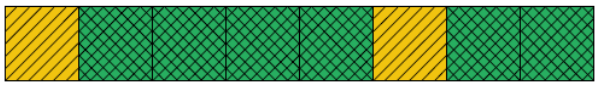


in naslednja pravila:



Kako bodo izgledali kvadrati, ko se bo robot ustavil?

A.



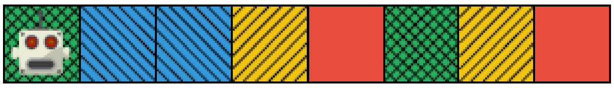
B.

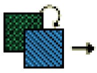
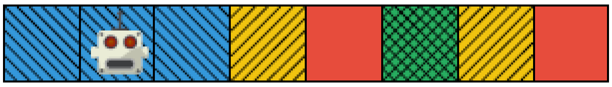
C.

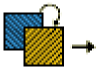
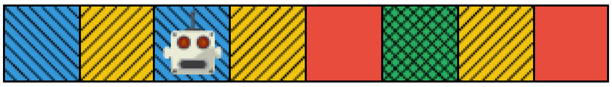
D.

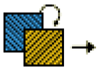
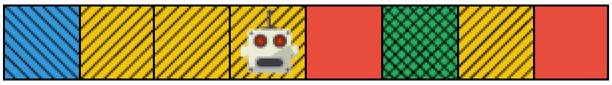
Rešitev

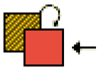
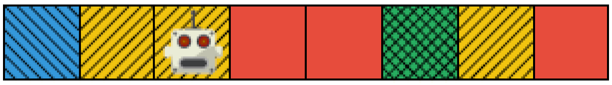
Pravilen odgovor je A.

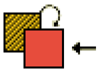
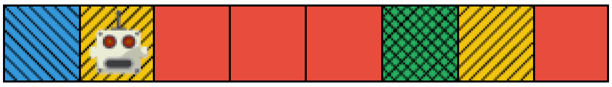
Začetno stanje: 

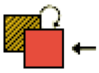
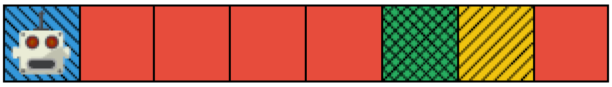
Če sledimo pravilu  , dobimo 

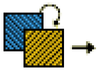
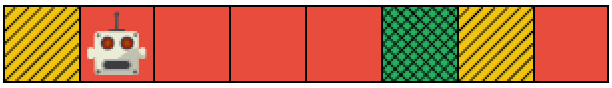
Če sledimo pravilu  , dobimo 

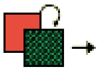
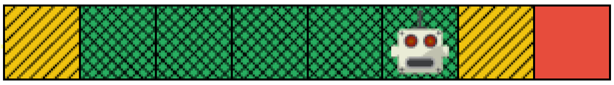
Če sledimo pravilu  , dobimo 

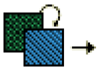
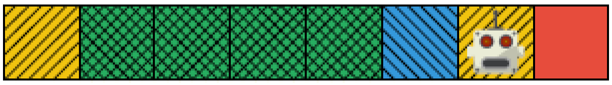
Če sledimo pravilu  , dobimo 

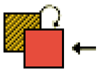
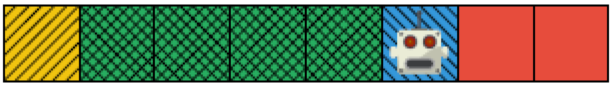
Če sledimo pravilu  , dobimo 

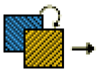
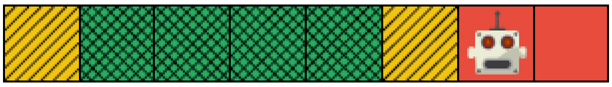
Če sledimo pravilu  , dobimo 

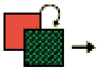
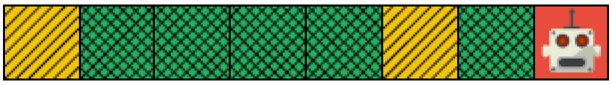
Če sledimo pravilu  , dobimo 

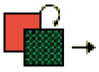
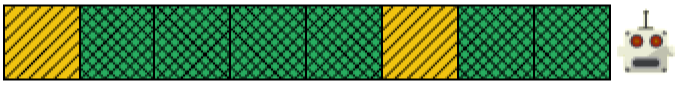
Če štirikrat sledimo pravilu  , dobimo 

Če sledimo pravilu  , dobimo 

Če sledimo pravilu  , dobimo 

Če sledimo pravilu  , dobimo 

Če sledimo pravilu  , dobimo 

Če sledimo pravilu  , dobimo 

Sedaj je robot prišel izven kvadratov, zato se ustavi.

Računalniško ozadje

Naš robot v nalogi se obnaša zelo podobno kot Turingov stroj. Čeprav je zelo enostaven, je Turingov stroj zelo uporaben model računanja za računalnikarje, saj je ekvivalenten večini programskih jezikov. To pomeni, da lahko katerikoli računalniški program spremenimo v Turingov stroj in nasprotno, vsak Turingov stroj lahko pretvorimo v računalniški program.

Skrivna tipkovnica

6. – 9. razred, srednja šola, državno

Pri običajni računalniški tipkovnici se ob pritisku na tipko na zaslonu prikaže ustrezna črka.

Pri skrivni tipkovnici pa se ob pritisku na tipko izpiše druga črka. S tem je preprosto napisati skrivna sporočila. Na primer, če pritisneš na tipko za črko Q, se izpiše L.

Če napišeš »JUTRI SE DOBIMO NA PICI«, se izpiše »KIZEU AR FLVUNL MS ŽUŠU«. Kakšna je povezava med črkami?

A.

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

B.

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

C.

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

D.

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

Rešitev

D. Povezave med črkami lahko ugotovimo tako, da gledamo povezavo med posameznimi črkami in najdemo povezave med njimi. Vse črke I se na primer preslikajo v črko U in tako naprej za vse črke v besedilu.

A ni pravilen odgovor, saj se P preslika v Ž in ne v Č, kot je zapisano v tabeli. B ni pravilen odgovor, ker se P preslika v Ž in ne v Š, kot kaže tabela. C ni pravilen odgovor, ker se A preslika v S in ne v F, kot kaže tabela.

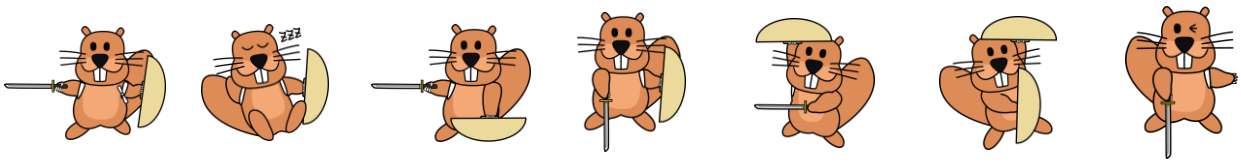
Računalniško ozadje

Ta tipkovnica je podobna Enigmi, ki jo je med 2. svetovno vojno nemška vojska uporabljala za pošiljanje skrivnih sporočil. Enigma je bila nadgrajena še z rotorjem, s pomočjo katerega so se povezave med črkami po vsaki zašifrirani črki še dodatno premešale.

Speči bober

6. – 9. razred, srednja šola, državno

Lucija se s sedmimi prijatelji igra igro meč in ščit. Spodnje slike prikazujejo najljubše položaje njenih prijateljev:

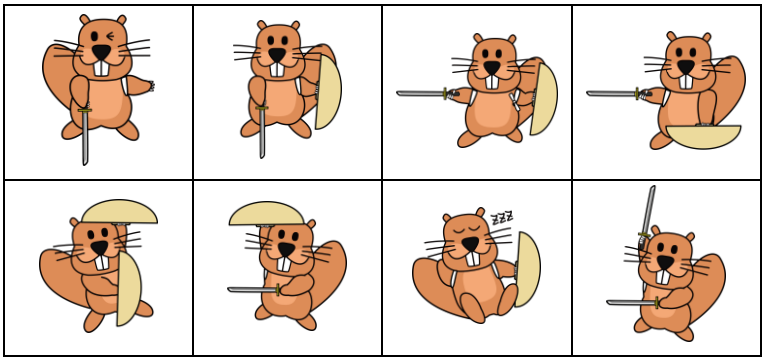


Bobri se želijo slikati tako, da bo vsak meč usmerjen v drugega bobra in da bo vsak ščit preprečil, da bi meč bobra poškodoval. Lucija je že zasedla svoje mesto na sliki:

A	B	C	D
E	F	G	

Kam na sliki se bo postavil speči bober?

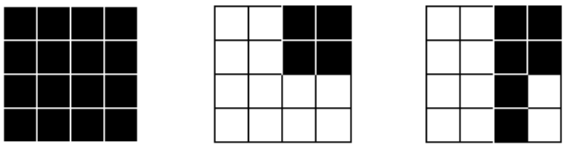
Rešitev



Računalniško ozadje

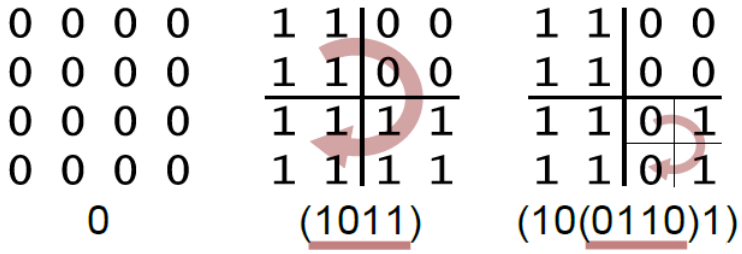
Tako kot pri sestavljanju slike v tej nalogi, tudi v računalništvu pogosto iščemo zakonitosti, ki jih moramo upoštevati pri reševanju nalog.

Poglejmo naslednje črno-bele bitne slike velikosti 4 x 4:



Te slike lahko shranimo z uporabo binarnih števk: 1 zapišemo za bele piksle, 0 pa za črne. Tako bi za slike velikosti 4 x 4 potrebovali 16 števk.

Vendar pa lahko za shranjevanje takih slik uporabimo tudi drugačno metodo stiskanja, ki omogoča shranjevanje z uporabo manj prostora (predvsem za enostavne vzorce):



Binarne števke so razporejene v mrežo kot piksli na sliki. Metoda stiskanja sestavi rezultat kot niz znakov, pri tem pa uporabi naslednji pravili:

1. Če so vse števke v mreži enake 0, je rezultat niz "0" (glej levo sliko). Če so vse števke enake 1, je rezultat "1".
2. Sicer pa mrežo razdeli na štiri podmreže. Na vsaki podmreži uporabi to metodo stiskanja, po vrsti od zgornje leve podmreže v smeri urinega kazalca. Rezultate stiskanj posameznih podmrež združi z uporabo oklepajev "(" in ")".

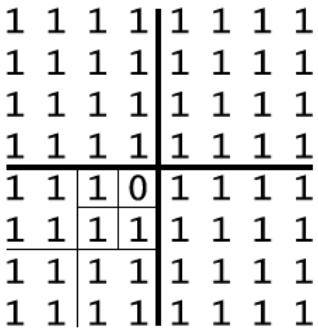
Srednja in desna slika prikazujeta dva taka primera stiskanja. Na desni sliki v spodnjem desnem kotu lahko vidimo, da podmreža lahko vsebuje tudi le eno samo števko; v tem primeru uporabimo le pravilo 1.

Za sliko velikosti 8 x 8 je mreža binarnih števk narisana na desni sliki. Kakšen niz dobimo kot rezultat, če na tej sliki uporabimo opisano metodo stiskanja?

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

Rešitev

Mrežo na sliki razdelimo na podmreže, kot prikazuje spodnja slika. Prve tri podmreže velikosti 4 x 4 stisnemo v niz "111", zadnjo podmrežo pa moramo spet razdeliti na štiri 2 x 2 podmreže. Tri od njih lahko zakodiramo z eno števko, drugo podmrežo pa moramo ponovno razdeliti na štiri podmreže velikosti 1 x 1. Niz torej sestavimo takole. Najprej imamo (111X), kjer je X zakodirana četrta podmreža. To zakodiramo kot (1Y11), kjer je Y zakodirana druga podmreža. To pa zakodiramo kot (1011). Če sestavimo celoten niz, dobimo (111(1(1011)11)).

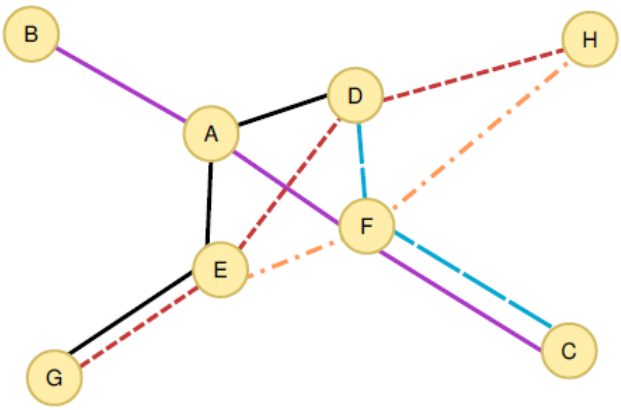


Računalniško ozadje

Čeprav opisana metoda stiskanja deluje nekoliko nenavadno, tak način predstavitve slik uporabljajo tudi bolj resni algoritmi za stiskanje slik. Takemu stiskanju rečemo stiskanje s pomočjo štiri-dreves (angleško *quad-trees*). Štiri-drevo je podatkovna struktura, pri kateri ima vsako notranje vozlišče natanko štiri otroke. Če sliko razdelimo na štiri območja, vsak otrok predstavlja eno območje. Območja lahko nadalje delimo v manjše dele. Pri delitvi slike navadno ne uporabljamo enako velikih delov (kot v našem primeru), saj nam delitev na neenake dele lahko prinese veliko boljše stiskanje.

Nižanje stroškov

6. – 9. razred, srednja šola, državno



Železniško podjetje vzdržuje osem železniških postaj in pet železniških povezav, ki jih kaže slika (vsaka povezava je narisana v drugi barvi). Železniško omrežje je zastavljeno tako, da lahko potujemo iz katerega koli kraja v kateri koli drug kraj z največ enim prestopanjem. Tako na primer za pot od B do H lahko vzamemo vijolično povezavo med B in F, tam prestopimo na oranžno povezavo in se peljemo do H.

Železniško podjetje želi znižati stroške poslovanja, zato bodo ukinili eno ali več povezav. Vendar morajo paziti, da vse postaje še vedno ostanejo povezane z železniškim omrežjem in da potovanje s katere koli postaje na katero koli drugo postajo še vedno zahteva največ eno prestopanje.

Največ koliko povezav lahko ukinejo?

Rešitev

Odgovor je največ 2 povezavi (ostanejo 3 povezave).

Če odstranimo rdečo povezavo (GEDH) in modro povezavo (DFC), je še vedno zadoščeno vsem pogojem. Povezave, ki ostanejo (BAFC, GEAD in EFH), namreč očitno povezujejo vseh osem postaj, poleg tega pa tudi velja, da ima vsak par povezav še vedno (najmanj) eno skupno postajo. Torej lahko s katere koli postaje pridemo na katero koli drugo postajo z enim samim prestopanjem.

Ali bi lahko odstranili tri povezave? Oziroma, če vprašanje zastavimo drugače: ali bi še zadostili vsem pogojem, če bi uporabili le dve povezavi? Odgovor je ne. Opazimo lahko, da je vijolična povezava (BAFC) edina, ki pelje do postaje B, zato jo moramo obdržati. Potem mora druga povezava povezovati vse ostale štiri postaje; edina taka možna povezava je rdeča (GEDH). Vendar pa ti dve povezavi nimata skupne postaje, zato z vlakom ne moremo potovati, na primer, od postaje B do postaje H. Torej dve povezavi ne zadostujeta za izpolnjevanje postavljenih pogojev.

Računalniško ozadje

V nalogi smo za predstavitev železniškega omrežja uporabili graf. V računalništvu pogosto uporabljamo grafe za predstavitev kompleksnih problemov.

Zabava

6. – 9. razred, srednja šola, državno

Študenti Bobrove akademije organizirajo zabavo, ki bo potekala od 10.00 do 20.00. Ves čas trajanja zabave potrebujejo vsaj enega prostovoljca, ki bo pri vhodu preverjal vstopnice obiskovalcev. Študenti, ki so se javili, da bodo pomagali, so v spodnjo tabelo napisali čas, ko so lahko dežurni.

11.00–12.00	15.30–16.30	19.00–20.00
10.00–10.30	10.15–11.15	19.15–19.30
17.15– 17.45	14.00–15.00	16.15–17.30
18.15–19.00	17.30–19.00	12.00–13.30
13.45–14.30	14.45–16.00	

Ostal je en termin, ko ni na voljo nikogar. Kdaj je to?

Rešitev

Najprej razvrstimo časovne rezine po času začetka:

10.00	10.15	11.00	12.00	13.45	14.00	14.45	15.30	16.15	17.15	17.30	18.15	19.00	19.15
|	|	|	|	|	|	|	|	|	|	|	|	|	|
10.30	11.15	12.00	13.30	14.30	15.00	16.00	16.30	17.30	17.45	19.00	19.00	20.00	19.30

Nato pogledamo časovne rezine in združimo tiste sosednje rezine, ki se prekrivajo. Tako dobimo dve ločeni časovni rezini 10.00-13.30 in 13.45-20.00. Vidimo, da se ni nihče javil, da bi bil pri vhodu v času od 13.30 do 13.45.

Računalniško ozadje

Nalogo lahko pogosto rešimo hitreje, če podatke prej uredimo po smiselnem vrstnem redu. Tako lažje najdemo iskani element, podvojene elemente in podobno.

Ena ena preveč

8. – 9. razred, srednja šola, državno

Pred kratkim sem odkril neumno napako, ki sem jo naredil v zelo dolgem besedilu. Namesto vseh znakov 1 bi moral pisati 11 in namesto vseh 11 bi moral pisati 1. Na srečo sem precej brihten in imam tudi urejevalnik besedila, v katerem lahko neko zaporedje znakov zamenjam z nekim drugim zaporedjem znakov. Poglejte, kaj dobimo, če besedop zopenjopo vse pojavitve am (razen zadnje) z op! Ali še bolje, kaj naredi zamenjaka kseh pojakitek v (razen zadnje) s k.

Besedilo mi je uspelo popraviti po enem od spodnjih postopkov. Katerem?

A. Vse znake 11 sem zamenjal z 1, nato pa sem zamenjal še vse znake 1 z 11.
B. Vse znake 1 sem zamenjal z 11, nato pa sem zamenjal še vse znake 11 z 1.
C. Vse znake 1 sem zamenjal z $, nato vse $ z 11 in nato še vse 11 z 1.
D. Vse znake 11 sem zamenjal z $, nato vse 1 z 11 in nato še vse $ z 1.

Rešitev

Pravilen odgovor je D.

Pri odgovoru A po zamenjavi vseh 11 z 1 ne bi mogli več ločiti, katera 1 predstavlja prvotno 1, ki jo moramo nato zamenjati z 11, in katera 1 je bila ravnokar spremenjena z 11. Na koncu bi tako imeli povsod le 11.

Pri odgovoru B ne zamenjamo le 1 v 11, ampak tudi 11 v 1111. Nato pa zamenjamo še 11 nazaj v 1 (in 1111 nazaj v 11), torej je rezultat po vseh zamenjavah enak začetnemu besedilu.

Če uporabimo način pri odgovoru C, smo v bistvu vse 1 zamenjali z 11 ter nato vse 11 z 1. Tako na koncu dobimo povsod same 1.

Postopek D ne bi bil pravilen, če bi bil v besedilu že od prej kak znak $. Vendar naloga pravi, da mi je besedilo uspelo popraviti, torej vemo, da ga ni bilo. Ostali postopki so namreč gotovo napačni.

Računalniško ozadje

Če ste pravi heker, boste ostali heker tudi pri uporabi tako dolgočasnih orodij, kot je urejevalnik besedil.

Pa še to: ta zgodba je povsem resnična, le da je moje dolgo besedilo pravzaprav računalniški program, v katerem sem pomotoma zamenjal dve konstanti.

Delo na kmetiji

srednja šola

Alja in Bojan nameravata zadnjih deset dni šolskih počitnic pomagati pri delu na kmetiji in tako zaslužiti nekaj bonbonov.

- Na kmetiji Ajvar bi vse dni dobila vsak po 50 bonbonov na dan.
- Na kmetiji Brinček bi prvi dan Bojan dobil le 10 bonbonov, saj vejo, da je Bojan še neizkušen. Drugi dan bi dobil 20 bonbonov, nato 30 in tako dalje. Vendar pa bi na tej kmetiji Alja dobila vsak dan manj: prvi dan bi dobila 100 bonbonov, drugi dan le 90, saj kmet ve, da bi bila drugi dan že utrujena in bi zato delala manj. Tretji dan bi dobila 80, nato 70 in tako dalje.
- Kmet na kmetiji Cuker pa ne mara Bojana, zato bi mu dal le en sam bonbon prvi dan, nato pa nič več. Da bi Bojana še bolj razjezil, pa bi Alja dobila en bonbon prvi dan, 2 drugi dan, 4 tretji dan, 8 četrti dan in tako naprej. Vsak naslednji dan bi Alja dobila dvakratnik plačila predhodnega dne. (Seveda kmet ne ve, da Alja na skrivaj vse zaslužene bonbone deli z Bojanom.)

Alja in Bojan si bosta na koncu razdelila zaslužene bonbone. Za katero kmetijo naj se odločita, da bosta skupaj zaslužila čim več?

Rešitev

Za kmetijo Brinček.

Izračun za kmetijo Ajvar je enostaven. Vsak od njiju zasluži 50 bonbonov na dan, torej vsak dobi 500 bonbonov v 10 dneh.

Kmetija Brinček ima na prvi pogled bolj kompliciran izračun: sešteti moramo 100 + 90 + 80 + 70 + 60 + 50 + 40 + 30 + 20 + 10. Koliko bi to lahko bilo? Razmislimo lahko tudi takole: Alja in Bojan skupaj bi prvi dan zaslužila 110 bonbonov, drugi dan tudi 110 bonbonov in vse naslednje dni tudi. Torej bi v desetih dneh skupaj zaslužila 1100 bonbonov oziroma vsak 550 bonbonov.

Izračun za kmetijo Cuker pa izgleda še bolj zapleten. Tudi tu je lažje, če izračunamo skupen zaslužek Alje in Bojana. Takole gre:

1 + 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 =
2 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 =
4 + 4 + 8 + 16 + 32 + 64 + 128 + 256 + 512 =
8 + 8 + 16 ... tole se pa lepo sešteva ...
preskočimo ostalo,
končati se mora z: 512 + 512 = 1024

Če Alja in Bojan razdelita zasluženih 1024 bonbonov, vsak dobi 512 bonbonov.

Torej odločitev za kmetijo Brinček ni težka.

Računalniško ozadje

Računalničarji pogosto seštevajo podobne vsote, na primer, ko računajo čas izvajanja algoritmov. Takrat številke ne predstavljajo bonbonov, ampak število operacij, ki jih računalnik opravi za obdelavo vsakega kosa podatkov. Tu se najbolj bojimo zadnjega primera, kmetije Cuker, kjer se število operacij **podvoji** z vsakim novim kosom podatkov. Pomislite, če bi Alja in Bojan na kmetiji Cuker delala le en dan več, bi vsak od njiju zaslužil še dodatnih 512 bonbonov!

Ste opazili, kako učinkovito smo v rešitvi izračunali obe vsoti? Če, na primer, želimo sešteti vsa števila od 1 do n, jih lahko seštevamo sočasno dvakrat, od spredaj in od zadaj, ter rezultat delimo z dva.

Ravno zato imajo računalnikarji zelo radi matematiko!

Nagrobnik

srednja šola

Šifriranje, pri katerem vsako črko zamenjamo z naslednjo črko po abecedi, je stara finta. Tudi premik za tri črke - ali za pet ali za osem - poznamo.

Kaj pa takole: prvo črko sporočila pustimo pri miru. Drugo črko sporočila zamenjamo z naslednjo po abecedi. Tretjo pomaknemo za dve črki po abecedi naprej. Četrto za tri, peto za štiri ...

Kaj piše na tem nagrobniku?

Rešitev

HASTA LA VISTA.

Računalniško ozadje

Povezava z računalništvom? »Hasta la vista, baby«, so bile (skoraj) zadnje besed znanega robota iz filma Terminator 2. Robotika je del računalništva, ki ... Ne kupite? Poskusimo drugače: prej omenjenega robota je igral Arnold Schwarzenegger, ki je bil med leti 2003 in 2011 guverner Kalifornije, zvezne države, ki je znana po Silicijevi dolini, v kateri se nahajajo največja računalniška podjetja, kot so Google, Apple ... Tudi ne kupite?

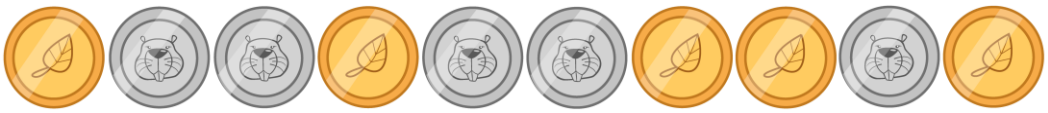
Prav, potem pa je naloga pač povezana s preprosto kriptografijo. Tako preprosto, da leta 2017 res sodi le še na nagrobnik in na takale tekmovanja.

Obračanje denarja

srednja šola

Imamo deset kovancev. Vsak kovanec je na eni strani zlat  , na drugi pa srebrn  .

Kovance postavimo v vrsto, kot prikazuje slika:



Kovance lahko obračamo, a vedno le v parih po dva sosednja. Želeli bi jih obrniti tako, da bodo vsi kazali zlato stran. Koliko potez (obračanj parov kovancev) potrebujemo za to?

Rešitev

Tako rešitev je nemogoče doseči. Pri postavljenih kovancih z obračanjem parov ne moremo doseči, da bi bili vsi kovanci obrnjeni enako.

Vsakokrat, ko obrnemo dva kovanca, število srebrnih ali ostane enako ali pa se poveča ali zmanjša za dva. V začetni postavitvi kovancev na sliki je liho število srebrnih kovancev. Torej bo tudi po vsakem obračanju število srebrnih kovancev liho. To pomeni, da nikoli ne moremo doseči postavitve, v kateri bi imeli nič srebrnih kovancev.

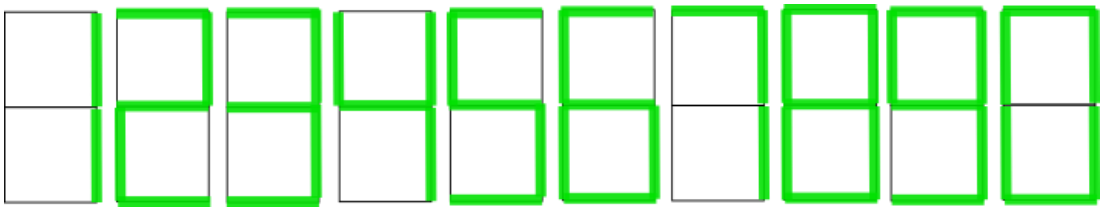
Računalniško ozadje

Parnost je zelo uporabna tudi v računalništvu. Uporablja se, na primer, pri prenosu in shranjevanju podatkov, kjer na konec bajta dodamo parnostni bit, ki pove, ali je število enic v bajtu liho ali sodo. S pomočjo tega dodatnega bita lahko enostavno preverjamo, ali je pri prenosu ali shranjevanju podatkov prišlo do napake.

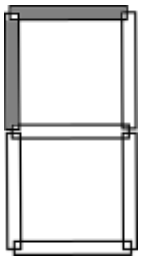
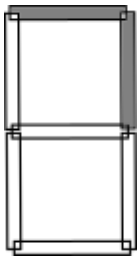
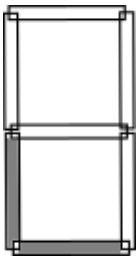
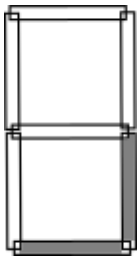
Prepoznavanje števil

srednja šola

Sistem za prepoznavanje števil prepozna števke, ki izgledajo, kot kaže slika:



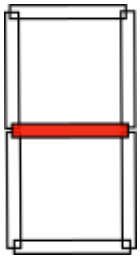
Vsaka števka je sestavljena iz največ sedmih delov (črtic). Za uspešno prepoznavanje posamezne števke ne potrebujemo vseh delov, ampak jo lahko uspešno prepoznamo tudi v primeru, če kakšen del ni viden. Kateri deli so lahko skriti (na slikah označeni sivo), pa bo sistem še vedno uspešno prepoznal vsako števko?

A. 
B. 
C. 
D. 

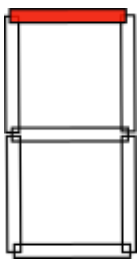
Rešitev

Pravilen odgovor je D.

Poskusimo najprej poiskati štiri dele, ki so potrebni, da razločimo vse števke. Ker se 0 in 8 razlikujeta le v enem delu, je ta del nujno potreben za uspešno prepoznavo.

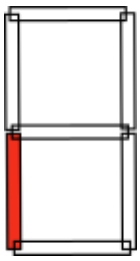


Nato lahko števke razdelimo v dve skupini glede na to, ali je pri njih ta del prisoten ali ne: pri {0,1,7} ni prisoten, pri {2,3,4,5,6,8,9} pa je. Ugotovimo tudi, da se števki 1 in 7 razlikujeta le v enem delu, torej moramo vključiti tudi ta del.

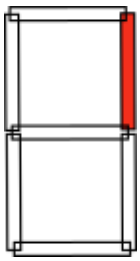


Sedaj lahko števke razdelimo na 4 skupine glede na to, ali vsebujejo ali ne oba posamezna dela: {1}, {0,7}, {4}, {2,3,5,6,8,9}. Vidimo, da oba izbrana dela že omogočata prepoznavo števk 1 in 4.

Sedaj poiščemo še druge števke, ki se razlikujejo le v enem delu. To sta 8 in 9. Torej moramo izbrati tudi ta del, ki loči števki 8 in 9.

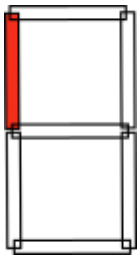


Torej lahko vse števke razdelimo na naslednje skupine z identičnimi elementi: {1}, {0}, {7}, {4}, {2,6,8}, {3,5,9}. Podobno se tudi števki 6 in 8 razlikujeta le v enem delu; tudi tega moramo izbrati.

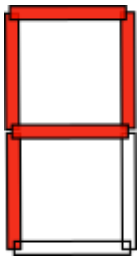


Z izbranimi štirimi deli lahko števke razdelimo v naslednje skupine: {1}, {0}, {7}, {4}, {2,8}, {6}, {3,9}, {5}. Vidimo, da le štirje deli ne zadostujejo, da bi razločili vse števke.

Števki 3 in 9 se razlikujeta le v enem delu. Ta del omogoča tudi razločevanje med števkama 2 in 8. Torej je to še zadnji del, ki ga moramo izbrati, da lahko ločimo med vsemi števkami.



Če torej sestavimo vse izbrane dele, vidimo, da so nujno potrebni vsi deli, ki so označeni rdeče.



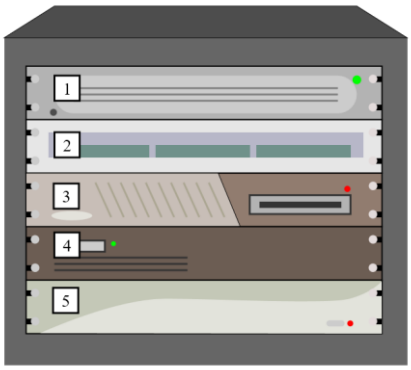
Verjetno ste tudi opazili, da je razmišljanje, ki nas je pripeljalo do rešitve, potrdilo tudi, da je to edina rešitev.

Računalniško ozadje

Algoritmi za prepoznavanje vzorcev v splošnem podajo logičen odgovor za vse vrste vhodov ter izvedejo najverjetnejše ujemanje vhoda, pri tem pa upoštevajo njihovo statistično variacijo. Prepoznavanje vzorcev je ena od vej strojnega učenja, ki se osredotoča na prepoznavo vzorcev in pravilnosti v podatkih. Eden od pristopov pri prepoznavanju je tudi tak, da izluščimo posebne lastnosti objektov, ki omogočajo njihovo enolično identifikacijo.

Računalniški center

srednja šola



V računalniškem centru imamo na voljo 5 strežnikov, ki so zloženi v ohišju, kot kaže slika. Potrebujemo nekaj strežnikov, ki bodo aktivno delovali, vendar ne moremo izbrati dveh sosednjih strežnikov, saj se v tem primeru preveč segrevata in odpovesta. Tako lahko na primer izberemo strežnika 1 in 3, ne moremo pa izbrati strežnikov 1 in 2.

Zanima nas, koliko je različnih konfiguracij, kjer lahko izberemo poljubno množico strežnikov (ta je lahko tudi prazna) tako, da noben par strežnikov iz te množice ni sosednji.

Če imamo le dva strežnika, potem lahko poiščemo tri take množice: v eni je prvi strežnik, v drugi je drugi strežnik, tretja množica pa je prazna.

Če imamo tri strežnike, lahko najdemo pet takih množic: tri vključujejo le po en strežnik, ena vsebuje strežnika 1 in 3, ena pa je prazna.

Koliko različnih množic lahko tvorimo s petimi strežniki?

Rešitev

Pravilen odgovor je 13 množic.

Tvorjene množice so naslednje: 1, 2, 3, 1 in 3, 1 in 4, 1 in 5, 1 in 3 in 5, 2 in 4, 2 in 5, 3 in 5, 4, 5 ter prazna množica.

Vedno lahko izberemo en sam strežnik ali pa nobenega, torej imamo najmanj 6 različnih množic: 1, 2, 3, 4, 5 ter množico brez strežnikov.

Nato lahko skupaj izberemo po dva strežnika, in sicer 1 skupaj v paru s 3, 4 ali 5; 2 skupaj v paru s 4 ali 5; ter 3 skupaj v paru s 5. Tako dobimo še 6 dodatnih množic: 1 in 3, 1 in 4, 1 in 5, 2 in 4, 2 in 5 ter 3 in 5.

Lahko pa izberemo skupaj tudi po tri strežnike, kjer je edina možnost združiti strežnike 1, 3 in 5.

Skupaj imamo torej možnih 13 različnih množic.

Računalniško ozadje

V nalogi smo računali, koliko množic lahko sestavimo, od prazne množice pa do množice, kjer se številke razlikujejo za 1. V nalogi smo morali izračunati odgovor le za 5 strežnikov, zato smo lahko

do rešitve prišli tako, da smo našteli vse možne kombinacije strežnikov. Kaj pa, če bi morali poiskati odgovor za 20 strežnikov? Ali pa morda za 100 strežnikov?

V tem primeru bi si morali pomagati z matematičnim izračunom. Pravzaprav je rezultat za n strežnikov enak (n+2)-temu Fibonaccijevemu številu. Fibonaccijeva števila so zelo znano zaporedje, za katero velja naslednja lastnost: n-to Fibonaccijevo število dobimo tako, da seštejemo (n – 1)-to in (n – 2)-to Fibonaccijevo število. Prvi dve števili v Fibonaccijevem zaporedju sta enaki 1.

Če želimo poiskati vse možne množice za strežnike 1, 2, …, n, poiščemo najprej vse možne množice za strežnike 1, 2, …, n – 1 ter vse možne množice za strežnike 1, 2, …, n – 2, katerim dodamo še strežnik n. Število vseh skupaj je torej enako vsoti vseh možnosti za (n–1) strežnikov in vseh možnosti za (n–2) strežnikov. Pri enem samem strežniku imamo dve množici, 2 pa je tudi tretje Fibonaccijevo število. Pri n strežnikih je torej število vseh možnih množic enako (n+2)-temu Fibonaccijevemu številu.

V računalništvu je zelo pomembno, da smo učinkoviti in da rezultat izračunamo na najboljši možni način. Za izračun n-tega Fibonaccijevega števila potrebujemo le rezultat dveh predhodnih Fibonaccijevih števil, ki ju seštejemo. Za izračun odgovora pri 20 strežnikih tako ni potrebno, da zapišemo vse možne množice (katerih je zelo veliko, več kot 10.000), saj lahko do rezultata pridemo z enostavnim seštevanjem.

Poleg zanimivih matematičnih lastnosti so Fibonaccijeva števila zelo uporabna tudi v računalništvu. Veliko algoritmov in podatkovnih struktur uporablja vsaj nekatere matematične lastnosti Fibonaccijevih števil, kar algoritme naredi bolj učinkovite.

Prefiksna koda črke zakodira kot števila z eno ali več števkami, vendar se nobena koda ne začne s števkami, ki tvorijo neko drugo kodo. Na primer, če črko **A** zakodiramo kot **12**, potem lahko črko **B** zakodiramo kot **2** (ker se 12 ne začne z 2). Črko **C** lahko v tem primeru zakodiramo v **11** (ker se niti 12 niti 2 ne začneta z 11), ne bi pa mogli za črko C uporabiti kode 21 (ker smo 2 že uporabili za B) ali 121 (ker smo 12 že uporabili za A).

S prefiksno kodo – v kateri črk A in B ne zapišemo nujno tako, kot smo za primer napisali zgoraj – smo zakodirali besedo ABAKUS in dobili naslednje zaporedje števk:

12112233321

V kaj smo zakodirali črko K?

Rešitev

Pravilen odgovor je 22.

Če v zakodiran niz števk vstavimo presledke med posameznimi kodami črk, dobimo naslednje zaporedje: 1 21 1 22 33 321. Torej se črka K zakodira v 22.

In kako lahko niz razbijemo na posamezne kode črk? Začnimo s prvo črko. Če A zakodiramo z dvema števkama, torej kot 12, potem bi morala črka B imeti kodo 1, kar ni možno, saj bi se koda za A začela s kodo za B. Tudi ni možno, da bi bila koda za A daljša kot 2 števki (121, 1211, 12112 in tako naprej), saj se ta črka ponovi, a v zaporedju ni nobenih takih ponovitev. Torej je lahko koda črke A le 1.

Črki A sledi črka B, torej lahko B zakodiramo le kot 2, 21 ali 211223332. Koda za B ne more biti 2, saj bi se v tem primeru beseda začela z ABAA. Tudi 211223332 ne more biti, saj bi v tem primeru bila zakodirana beseda ABA. Torej je koda za B enaka 21.

Sedaj vemo, da je z 1 21 1 zakodiran začetek besede ABA. Ugotoviti moramo še pomen kod 2233321.

Na koncu besede imamo črko S. Njena koda ne more biti 1 ali 21, saj sta ti dve kodi že uporabljeni za A in B. Torej so možne kode za S le 321, 3321, 33321, 233321 ter 2233321. Koda za S ne more biti 2233321, saj bi bila v tem primeru naša beseda ABAS. Tudi ne more biti 233321, saj bi nam v tem primeru ostala le ena števka za kodiranje dveh črk, K in U. Ne more biti niti 33321, saj bi v tem primeru morali dve različni črki (K in U) zakodirati z isto kodo 2. Če je koda S 3321, potem sta črki KU zakodirani kot 223. Vendar pa koda za K ne more biti le 2 in koda za U ne more biti le 3, saj se ostale kode začnejo s tema dvema števkama (B se začne z 2, S pa s 3). Torej je koda za S lahko le 321. Tako 2233 zakodira KU in edina možnost je, da je K zakodiran kot 22 in U kot 33.

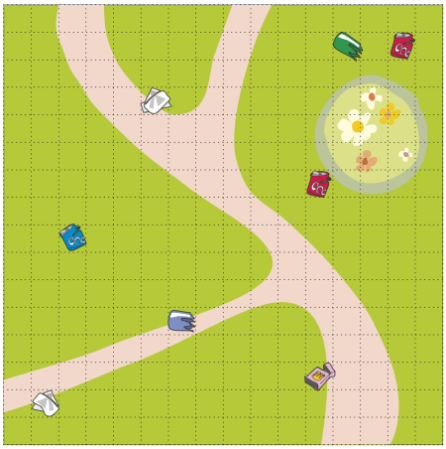
Računalniško ozadje

Naloga prikazuje primer kodiranja z uporabo prefiksne kode, kjer elemente zakodiramo s kodami različnih dolžin, pri tem pa se nobena koda ne začne z neko drugo kodo. Ta lastnost je uporabna pri dekodiranju informacij, saj med posameznimi kodami ne potrebujemo presledkov (ne potrebujemo označenega začetka posamezne kode).

Pogosto prefiksno kodo uporabimo tako, da za podano besedilo poiščemo tako kodiranje, da je zakodiran niz čimkrajši. To uporabimo pri arhiviranju podatkov, kjer zmanjšamo velikost podatkov, ne da bi izgubili kakršnokoli informacijo. Metoda, po kateri dobimo optimalno prefiksno kodo, se imenuje Huffmanovo kodiranje. Slednje se pogosto uporablja pri stiskanju in kodiranju, ne le za besedila (format zip), saj je tudi del popularnih medijskih formatov jpeg in mp3.

Robotski čistilci

srednja šola

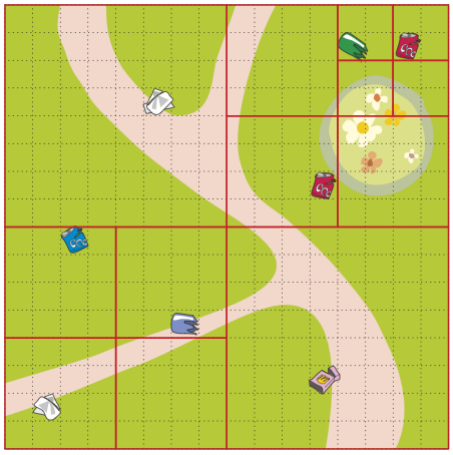


Mestne oblasti so se odločile, da bodo osrednji mestni park čistili robotski čistilci. Park so razdelili na 256 delov, v mrežo 16 x 16 kvadratkov, ter pripravili tudi množico robotov-čistilcev. Vsak robot lahko preišče določeno kvadratno področje v parku in najde ter pobere natanko eno smet. Vendar pa lahko roboti delujejo in pobirajo smeti, le če v vsako kvadratno območje parka postavimo robota, četudi tam ni nobenih smeti.

Park lahko razdelimo na kvadratna območja velikosti 1 x 1, 2 x 2, 4 x 4 in 8 x 8 kvadratkov mreže, v vsako kvadratno območje pa postavimo enega robota-čistilca. Mrežo in smeti v parku kaže slika.

Kakšno je najmanjše število robotov, ki jih moramo uporabiti, da poberemo vse smeti v parku?

Rešitev



Pravilen odgovor je 13 robotov.

Rešitev prikazuje slika, na kateri smo z rdečo črto označili, kako park razdelimo na kvadratna območja.

Verjetno ste opazili, da moramo uporabiti vsaj 8 kvadratnih območij, saj imamo v parku 8 smeti in za vsako od njih potrebujemo enega robota, da jo pobere. Če želimo dobiti najmanjše število kvadratov, moramo maksimizirati velikost kvadratnega območja okoli vsake smeti. V primeru rešitve na sliki vidimo, da ne obstaja nobeno večje kvadratno območje predpisanih velikosti, ki bi obkrožalo vsak kos smeti in se ne bi prekrivalo z ostalimi smetmi.

Računalniško ozadje

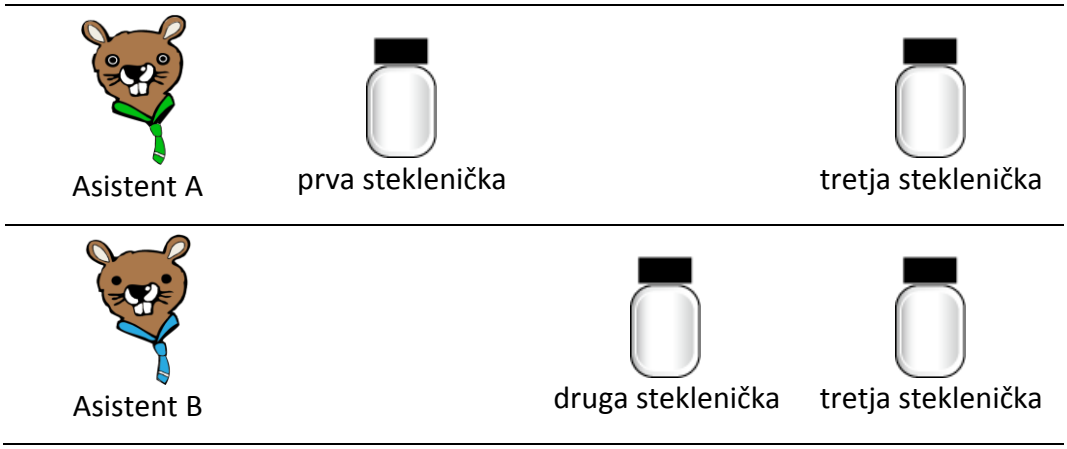
Tako predstavitev dvodimenzionalnih informacij imenujemo štiri-drevo (angleško *quad-tree*). Štiri-drevo razdeli dvodimenzionalno področje v štiri enake dele, kjer lahko vsak del nadalje delimo v nove štiri dele in tako naprej, dokler na koncu ne pridemo do enega samega elementa v vsakem delu (ki ga zato ne moremo več deliti). Uporaba štiri-dreves kot podatkovnih struktur omogoča shranjevanje in učinkovito iskanje grafičnih informacij, hkrati pa omogoča tudi povpraševanje po obsegu, kot je na primer odgovor na vprašanje, koliko objektov je v določenem delu slike.

Uspavalni napoj

srednja šola

Dr. Bober je pomotoma postavil stekleničko njegove najnovejše iznajdbe, uspavalnega napoja, med ducate stekleničk z vodo. Ker so vse stekleničke enake in ker sta tako voda kot tudi uspavalni napoj brez barve, vonja in okusa, ni mogoče, da bi uspavalni napoj našel na podlagi njegovega izgleda. Zato prosi svoje asistente, da spijejo manjšo količino tekočine, ki jo vzame iz ene ali več stekleničk (uspavalni napoj lahko mešamo z vodo), ter opazuje rezultate. Uspavalni napoj je namreč zelo močan in četudi spiješ le manjšo količino, boš zaspal prej kot v pol ure. Seveda pa voda ne učinkuje in če jo spiješ, boš ostal buden.

Na primer, če v testu sodelujeta **dva** asistenta, prvi (A) spije mešanico tekočin iz **prve** in **tretje** stekleničke, drugi (B) pa spije mešanico iz **druge** in **tretje** stekleničke.



Glede na rezultate testa lahko sklepamo, v kateri stekleničkі je uspavalni napoj:

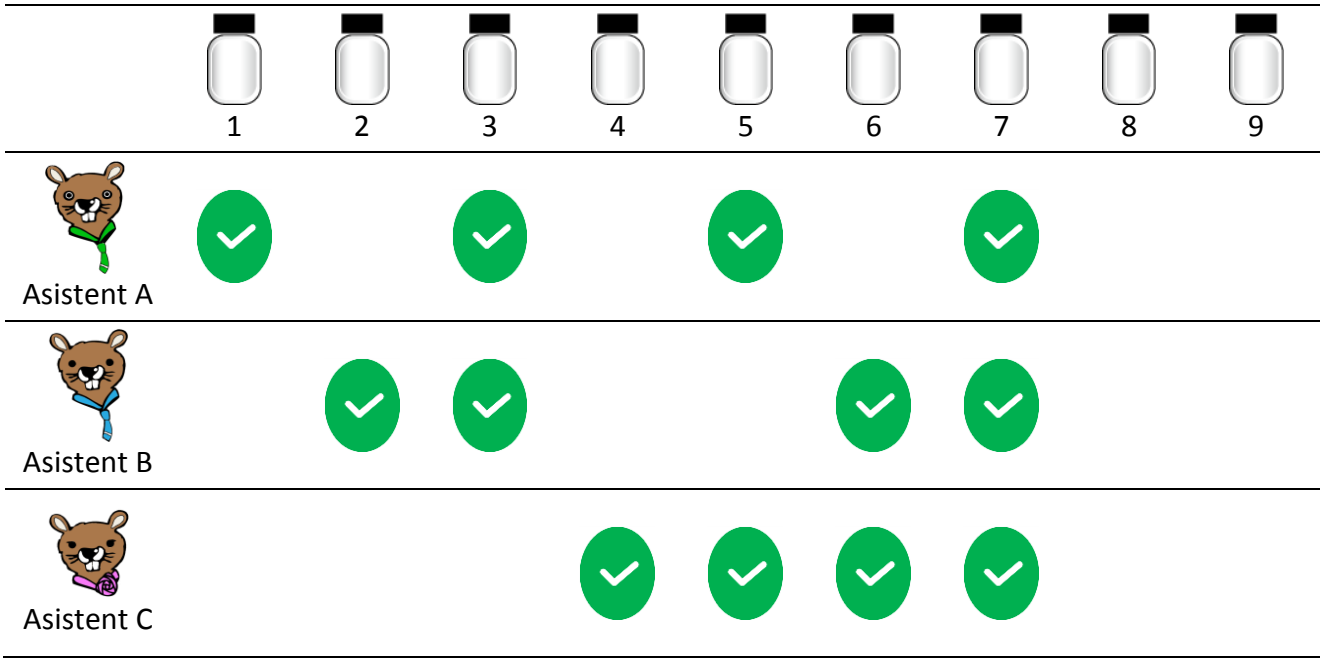
- Če zaspi le asistent A, vemo, da je napoj v prvi stekleničkі, v ostalih pa je voda.
- Če zaspi le asistent B, vemo, da je napoj v drugi stekleničkі.
- Če zaspita oba asistenta, A in B, vemo, da je napoj v tretji stekleničkі.
- Če ne zaspi nobeden od asistentov, potem vse tri stekleničke vsebujejo navadno vodo. V tem testu smo preverjali vsebino treh stekleničk.

Recimo, da v testu sodelujejo trije asistenti. Največ koliko stekleničk s tekočino lahko preverimo, če vsak od asistentov le enkrat spije mešanico tekočin?

Rešitev

Pravilen odgovor je sedem stekleničk.

Če v testu sodelujejo trije asistenti, lahko preverimo največ sedem stekleničk s tekočino ($2^3 - 1$).



- Če zaspi le asistent A, je v prvi stekleničkiuspavalni napoj, vse ostale pa vsebujejo vodo.
- Če zaspi le asistent B, je uspavalni napoj v drugi stekleničkiu.
- Če zaspita oba asistenta A in B, uspavalni napoj hrani tretja steklenička.
- Če zaspi le asistent C, je uspavalni napoj v četrti stekleničkiu.
- Če zaspita oba asistenta A in C, je uspavalni napoj v peti stekleničkiu.
- Če zaspita oba asistenta B in C, imamo uspavalni napoj v šesti stekleničkiu.
- Če zaspijo vsi trije asistenti A, B in C, potem je uspavalni napoj zagotovo v sedmi stekleničkiu.
- Če pa vsi trije asistenti ostanejo budni, to pomeni, da je v vseh sedmih stekleničkah navadna voda. V tem primeru smo tako preverili sedem (oziroma $2^3 - 1$) stekleničk s tekočino.

Računalniško ozadje

Pri reševanju naloge smo uporabili binarna števila, to so števila v dvojiškem sistemu (z bazo 2), ki za predstavitev uporabljajo le dve vrednosti, 0 in 1. V nalogi bi lahko rekli, da 0 predstavlja budnega asistenta, 1 pa predstavlja asistenta, ki je zaspal (po preizkušanju mešanice tekočin).

Torej bi lahko statuse vseh treh asistentov na kratko zapisali takole:

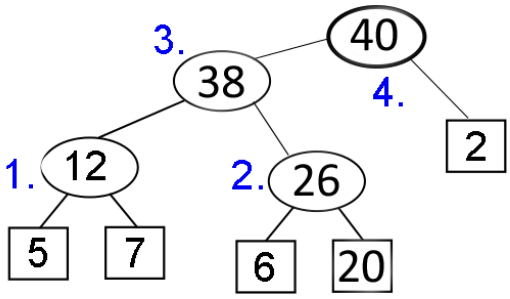
Zapis	Opis statusa	Rezultat
001	Zaspi asistent A.	Uspavalni napoj je v prvi stekleničkiu.
010	Zaspi asistent B.	Uspavalni napoj je v drugi stekleničkiu.
011	Zaspita oba asistenta A in B.	Uspavalni napoj je v tretji stekleničkiu.
100	Zaspi asistent C.	Uspavalni napoj je v četrti stekleničkiu.
101	Zaspita oba asistenta A in C.	Uspavalni napoj je v peti stekleničkiu.
110	Zaspita oba asistenta B in C.	Uspavalni napoj je v šesti stekleničkiu.
111	Zaspijo vsi trije asistenti.	Uspavalni napoj je v sedmi stekleničkiu.
000	Vsi trije asistenti ostanejo budni.	V vseh sedmih stekleničkah je voda.

Zlato

Zlatko je v bližnjem potoku našel več koščkov zlata. Rad bi jih pretopil v en sam kos. Prijatelj ima talilnico, v kateri lahko hkrati stali le dva kosa zlata, vsako taljenje pa stane en cent za gram zlata.

Če imamo pet koščkov zlata, ki tehtajo 5 g, 7 g, 6 g, 20 g in 2 g, jih lahko, na primer, pretopimo v en 40-gramski kos v naslednjem zaporedju (glej tudi sliko):

1. Pretopi skupaj kosa 5 g in 7 g, da dobiš 12 g. To taljenje stane 12 centov.
2. Pretopi 6 g in 20 g, da dobiš 26 g. To taljenje stane 26 centov.
3. Pretopi skupaj oba kosa po 12 g in 26 g, ki si ju ustvaril s predhodnima taljenjema, da dobiš 38 g. To taljenje stane 38 centov.
4. Na koncu pretopi skupaj 38 g in 2 g, da dobiš 40 g. To taljenje stane 40 centov.



Po četrtem taljenju smo skupaj stopili vseh pet koščkov zlata in sestavili en sam kos. Skupna cena taljenja pa je 12 + 26 + 38 + 40 = 116 centov.

Enak rezultat, to je en 40-gramski kos zlata, bi prav tako s štirimi topljenji lahko dobili tudi tako, da bi sestavljali posamezne kose na naslednji način: 5 g + 7 g = 12 g, 6 g + 2 g = 8 g, 12 g + 20 g = 32 g ter 32 g + 8 g = 40 g. V tem primeru je skupni strošek taljenja le 12 + 8 + 32 + 40 = 92 centov.

Zlatko ima 8 koščkov zlata naslednjih tež:

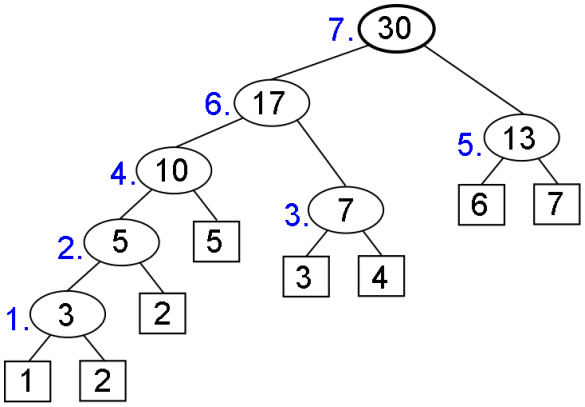
7 g, 1 g, 3 g, 2 g, 6 g, 2 g, 5 g, 4 g

Najmanj koliko ga bo stalo taljenje, da iz njih dobi en sam kos zlata?

Rešitev

Pravilen odgovor je 85 centov.

Vse kose lahko pretopimo skupaj z minimalno ceno 85 centov, kot to prikazuje spodnja slika. Kvadrati na sliki predstavljajo začetne koščke zlata, elipse pa pretopljene kose. Številka znotraj njih prikazuje težo koščka. Vrstni red, v katerem pretapljamo skupaj koščke zlata, je na sliki označen z modrimi številkami poleg elips. Skupna cena je vsota vseh tež stopljenih koščkov zlata (številke znotraj elips), kar je 3 + 5 + 7 + 10 + 13 + 17 + 30 = 85 centov.



Teža vsakega začetnega koščka zlata se šteje enkrat pri vsakem topljenju, v katerega je vključen. Če želimo najnižjo skupno ceno, moramo minimizirati število topljenj najtežjih koščkov. Tako razmišljanje nas vodi do optimalne rešitve, ki je prikazana na zgornji sliki: na vsakem koraku stopimo skupaj dva najlažja koščka zlata.

Računalniško ozadje

S problemom optimizacije se zelo pogosto srečamo tudi v vsakdanjem življenju. Ta naloga je primer oblike kombinatorične optimizacije, kjer moramo poiskati najmanjši strošek obdelave podane množice elementov, skupen strošek pa je odvisen od vrstnega reda obdelave. Probleme optimizacije lahko rešimo na različne načine. Eden najbolj enostavnih načinov, ki deluje pravilno tudi na primeru v naši nalogi, je uporaba požrešnega algoritma, ki na vsakem koraku obdela elemente z najmanjšo posamezno ceno.

Igra s fižoli

srednja šola, državno

Ana in Borut igrata igro s fižoli, ki ima naslednja pravila. Na mizi sta dva kupčka fižolovih zrn. Ana in Borut si izmenjujeta poteze:

1. Igralec na potezi najprej z mize odstrani enega od kupčkov.
2. Nato preostali kupček razdeli na dva kupčka.

Zmaga tisti igralec, ki mu uspe na mizi pustiti dva kupčka, v vsakem pa je natanko en fižol. Prva je na vrsti Ana.

Igro začnemo s 24 fižoli, ki so razdeljeni na dva kupčka. Katera od navedenih začetnih razdelitev fižolov omogoča Ani, da bo s pametno igro prav gotovo zmagala?

A. 11 in 13 B. 3 in 21 C. 7 in 17 D. 8 in 16

Rešitev

Pravilni odgovor je D.

Začetna razdelitev 24 fižolov je lahko na dva kupa z lihim številom fižolov ali na dva kupa s sodim številom fižolov, saj 24 fižolov ne moremo razdeliti na dva kupa, kjer ima eden sodo, drugi pa liho število fižolov. Dobra zmagovalna strategija igre je, da v vsaki potezi poskusimo narediti dva kupa z lihim številom fižolov.

Poglejmo, zakaj. Če ima igralec na potezi na mizi dva liha kupa (tj. kupa z lihim številom fižolov), lahko preostali kup razdeli le na lihi in na sodi kup. Naslednji igralec tako lahko z mize odstrani lihi kup in sodi kup razdeli na dva liha. Če sta to kupa s po enim fižolom, se igra zaključi in igralec, ki je naredil to potezo, zmaga. Če pa je v vsakem kupu več fižolov, nasprotni igralec ne more zmagati v naslednji potezi, saj lahko lihi kup razdeli le na en sodi in en lihi, kar ne ustreza pogoju za zmago.

Kot vidimo, v tej igri dobra strategija igranja ne zadostuje za gotovo zmago. Potrebujemo tudi pravo začetno razdelitev fižolov.

Računalniško ozadje

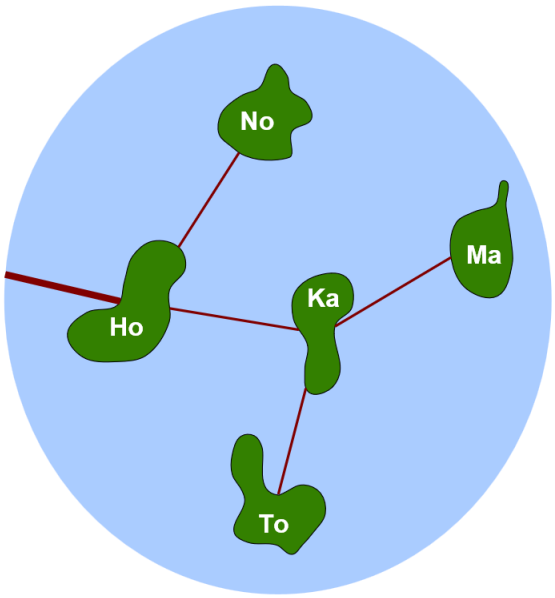
Zmagovalno strategijo lahko sestavimo tako, da poiščemo invariantno (tj. nespremenljivo) lastnost: ta je lahko resnična po naši potezi, vendar je nasprotni igralec ne more doseči, izvesti in je resnična tudi ob koncu igre, v kateri zmagamo.

Pri večini iger takšnih strategij ne moremo poiskati. Pač pa lahko za posamezno pozicijo ocenimo, kako dobra je. Računalniški programi zato navadno igrajo tako, da pred vsako potezo pregledajo možna nadaljevanja in med možnimi potezami izberejo tisto, ki vodi v čim boljšo pozicijo.

Honomakato

srednja šola, državno

Arhipelag Honomakato sestavlja pet rajskih otokov Ho, No, Ma, Ka in To. Največji otok Ho je povezan v internet z velikim podmorskim kablom. Ostale otoke povezujejo manjši kabli, in sicer Ho in No, Ho in Ka, Ka in Ma ter Ka in To. Preko teh kablov so otoki povezani s Ho in tudi naprej v internet.



Prebivalci arhipelaga Honomakato želijo prilagodljivo omrežje, ki bi delovalo tudi v primeru napake na katerem od manjših kablov, ki povezujejo otoke.

Če lahko med otoki položimo še dva kabla in tako zagotovimo prilagodljivost omrežja, med katerimi otoki moramo položiti kabla?

A. Med Ho in To ter med No in Ma.
B. Med Ho in To ter med Ma in To.
C. Med Ka in No ter med No in Ma.
D. Dva dodatna kabla nista dovolj, da bi dobili prilagodljivo omrežje.

Rešitev

Pravilen odgovor je A: kabla moramo položiti med Ho in To ter med No in Ma.

Če dodamo kabel Ho-To, je To še vedno povezan, če izpade kabel Ho-Ka ali kabel Ka-To. Tudi Ka je še vedno povezan preko To, če je napaka na kablu Ho-Ka.

Če dodamo kabel No-Ma, je Ma povezan preko No v primeru izpada kabla Ho-Ka ali kabla Ka-Ma. V primeru izpada kabla Ho-No je tudi No je povezan preko Ma in Ka. Poleg tega je povezan tudi Ka preko Ma in No, čeprav je okvara na kablu Ho-Ka.

Če bi kabla potegnili med Ma in To ter med Ho in To, bi bil No odrezan od omrežja, če bi izpadel kabel Ho-No.

Podobno tudi kabla Ka-No in No-Ma ne omogočata povezanosti To, če se pojavi napaka na kablu Ka-To.

Računalniško ozadje

Kabelsko omrežje arhipelaga Honomakato ne predstavlja le majhnega delčka interneta, temveč je na podoben način sestavljen celoten internet: vsi računalniki, pametni telefoni, televizije, hladilniki in druge naprave, ki so dandanes priključene v internet, predstavljajo vozlišča tega omrežja, kot so vozlišča tudi posamezni otoki omrežja arhipelaga Honomakato.

Internet je nastal prav zato, ker so v šestdesetih letih prejšnjega stoletja razmišljali, kako zgraditi prilagodljivo omrežje, ki bo prožno in robustno ter bo delovalo tudi v primeru, če odpove posamezna povezava tega omrežja. Nedelujoče povezave med posameznimi vozlišči naj torej ne bi onesposobile celotnega omrežja. Podobno velja tudi za druga omrežja, kot so cestna omrežja ali preskrbovalna omrežja za elektrilo in vodo: zelo pomembno je, da v omrežju ni ene same točke ali povezave, katere nedelovanje lahko onesposobi delovanje večjega dela omrežja.

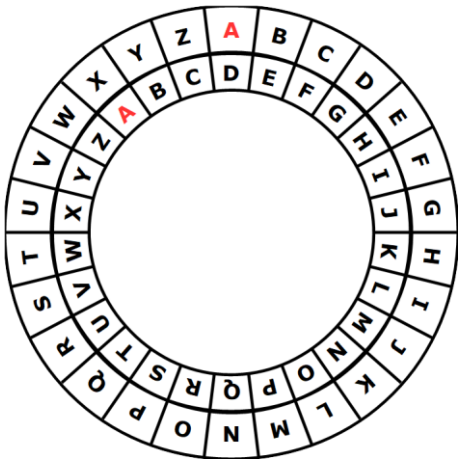
Računalnikarji pri razlagi omrežij uporabljajo *teorijo grafov*. Graf je omrežje, ki ga sestavljajo vozlišča in povezave med vozlišči. Graf je *povezan*, če sta povezani kateri koli dve vozlišči A in B v grafu (na primer, če lahko vozlišče A dosežemo iz vozlišča B preko ene ali več povezav). Če imamo v grafu tako povezavo, da mora ta povezava obstajati, da je graf povezan, tako povezavo imenujemo *most*. Če želimo imeti prilagodljivo omrežje, se moramo izogniti mostovom. Na srečo v računalništvu poznamo algoritme, ki znajo poiskati mostove v grafu; Robert Tarjan je iznašel zelo učinkovit algoritem za odkrivanje mostov (pa tudi kar nekaj drugih uporabnih algoritmov za grafe).

Kaj je za kosilo?

srednja šola, državno

Bibo in Biba si izmenjujeta sporočila, zašifrirana s šifrirnim diskom. Ta ima notranji in zunanji disk. Vsak dan Biba v zašifriranem sporočilu pošlje Bibu, kaj si želi za kosilo. Sporočilo je zašifrirano tako:

1. Biba napiše ime jedi, na primer »SOLATA«.
2. Pod vsako črko napiše številko med 1 in 9 in obrne notranji disk na levo za toliko mest od začetne poravnane pozicije. Slika na desni kaže šifrirni disk, ko je notranji disk obrnjen za 3 mesta v levo.
3. Biba napiše ustrezno črko z notranjega diska na papir. Slika na desni kaže, kako je »S« zakodiran kot »V«.
4. Biba pošlje zašifrirano sporočilo Bibu. Ta potem odšifrira sporočilo, da izve, kaj si Biba želi za kosilo.



Če Biba želi sporočiti SOLATA, napiše:

Sporočilo	**S**	**O**	**L**	**A**	**T**	**A**
Obrni levo	3	1	4	3	5	2
Zašifrirano sporočilo	**V**	**P**	**P**	**D**	**Y**	**C**

Kaj želi za kosilo danes?

Sporočilo	**?**	**?**	**?**	**?**	**?**	**?**	**?**
Obrni levo	3	5	1	7	4	2	8
Zašifrirano sporočilo	**O**	**F**	**A**	**H**	**R**	**L**	**I**

Rešitev

LAZANJA

Računalniško ozadje

V nalogi si spoznal preprosto metodo šifriranja. Takšne metode se danes ne uporabljajo več, saj jih današnji računalniki brez težav razbijejo, razen če si računalnika pred pošiljanjem sporočila pošljeta geslo, ki je enako dolgo kot sporočilo – kar pa seveda nima smisla.

Kovanci

srednja šola, državno

V šoli skrbijo za zdravo prehrano, zato so namestili avtomat z jabolki. Če kdo plača preveč, mu avtomat vrne razliko v kovancih. Vrednosti kovancev so 31, 17, 10 in 1 BEB. Avtomat vrača razliko v skladu z naslednjimi pravili:

- Vzemi največji kovanec, ki je manjši ali enak vrednosti, ki jo je potrebno vrniti, vrni kovanec in zmanjšaj razliko, ki jo je potrebno vrniti za vrednost tega kovanca.
- Ponavljaj prejšnji korak, dokler razlika ni enaka 0.

Na primer, če mora avtomat vrniti 33 BEB, avtomat vrne en kovanec za 31 BEB in dva kovanca za 1 BEB.

Ta postopek ne zagotavlja, da bo avtomat vrnil najmanjše število kovancev za vsako vrednost. Pri katerem znesku bi avtomat lahko vrnil tri kovance, vendar jih vrne več?

A. 21 BEB B. 36 BEB C. 12 BEB D. 52 BEB

Rešitev

Pravilni odgovor je 21 BEB.

21 BEB bi lahko vrnil s 3 kovanci: 10 BEB + 10 BEB + 1 BEB, vendar vrne 5 kovancev: 17 BEB + 1 BEB + 1 BEB + 1 BEB + 1 BEB.

Za 36 BEB potrebuje 4 kovance: 17 BEB + 17 BEB + 1 BEB + 1 BEB.

Pri 12 BEB lahko vrne tri kovance – in tudi jih: 10 BEB + 1 BEB + 1 BEB.

Za 52 BEB potrebuje 4 kovance: 31 BEB + 10 BEB + 10 BEB + 1 BEB.

Računalniško ozadje

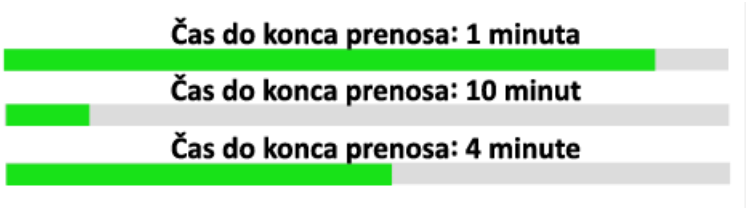
Naloga opisuje instinktivno metodo vračanja drobiža, ki jo uporablja vsak. Ta problem poznamo pod imenom problem vračanja kovancev, ki zahteva, da za dani znesek vrnemo najmanjše število kovancev. Vračanje kovancev je lahko kar zapleten problem.

Prenos datotek

srednja šola, državno

Pri prenosu datotek s strežnika imamo omejeno skupno hitrost prenosa. Če na primer sočasno prenašamo 10 datotek, je čas prenosa vsake datoteke desetkrat počasnejši, kot če bi prenašali eno samo datoteko naenkrat.

Uporabnik s strežnika hkrati prenaša 3 datoteke. Slika prikazuje trenutno stanje prenosov. Čas do konca prenosa je izračunan le glede na trenutno hitrost prenosa in ni odvisen od preteklih prenosov.

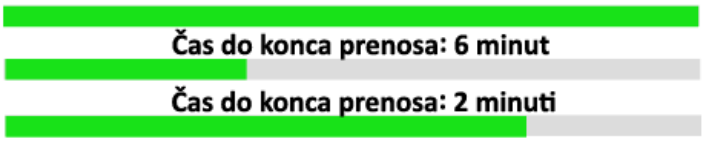


Koliko časa še potrebujemo, da se prenesejo vse tri datoteke?

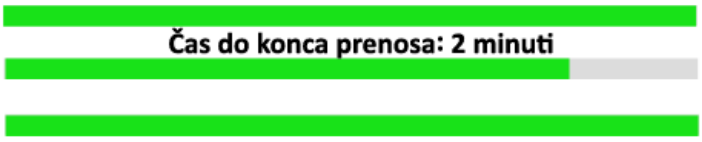
Rešitev

Pravilen odgovor je 5 minut.

Po eni minuti bo prva datoteka prenešena v celoti, zato se bo hitrost prenosa za ostali dve datoteki povečala za 3/2 (tj. namesto 3 datotek bomo prenašali le 2). Zato se bo v tistem trenutku tudi ustrezno skrajšal čas prenosa (to je za 2/3): namesto 9 minut za drugo datoteko, bo potrebno le še 6 minut, namesto 3 minute za tretjo datoteko pa le še 2 minuti. Napredovanje prenosa bo takšno:



Po dveh minutah bo v celoti prenešena tudi tretja datoteka, hitrost prenosa preostale datoteke pa se bo zato podvojila. Tako do konca prenosa druge datoteke potrebujemo le še 2 minuti.



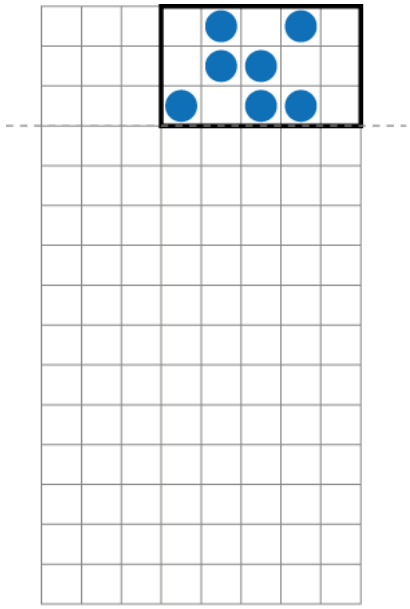
Torej bodo po 1 + 2 + 2 = 5 minutah prenešene vse tri datoteke.

Računalniško ozadje

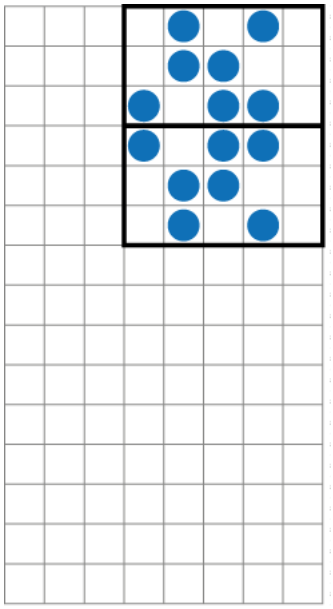
V vrstici napredka (angleško *progress bar*) velikost zapolnjenega dela predstavlja delež časa, ki je minil od začetka operacije, napram celotnemu času, potrebnemu za izvedbo. Navadno je ta čas poznan ali izračunan le približno. Dobra ocena preostalega časa pa je precej zahtevna naloga.

Prepogibi

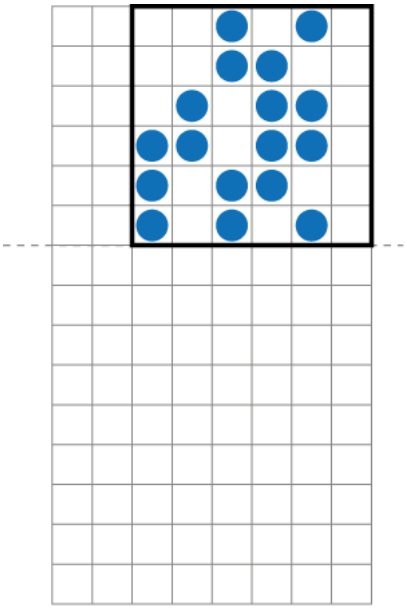
Vzemi kos karirastega papirja in označi pravokotno območje desno zgoraj, recimo tri vrstice in pet stolpcev. Popackaj naključno izbrane kvadrate.



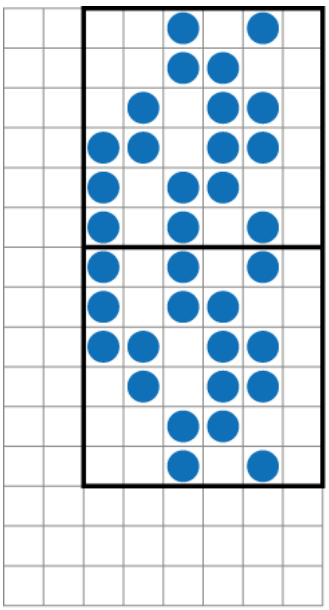
Prepogni tik pod pravokotnikom. Ker je črnilo še sveže, dobiš zrcalno sliko pack.



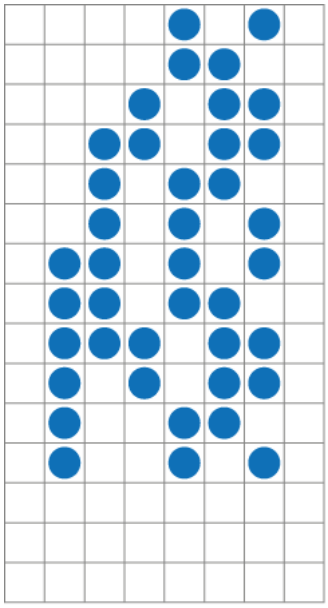
Popackaj kvadrate v stolpcu levo od pravokotnika, a le na zrcalni sliki.



Spet prepogni; tokrat pod zrcalno sliko.

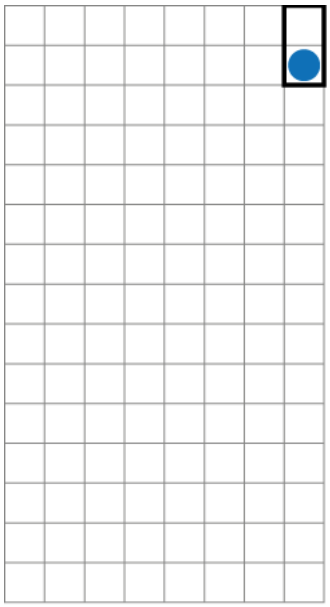


Spet dodaj packe na zrcalno sliko v stolpcu levo od stolpca, v katerem si dodajal v prejšnjem koraku.



Dobil si vzorec vrstic.

Nekatere vrstice so si med seboj precej podobne, kot na primer tretja in četrta, ki se razlikujeta le na enem mestu. Druge so si različne: druga in tretja se razlikujeta na treh mestih.



Začnimo od začetka, s preprostejšim vzorcem z enim stolpcem in dvema vrsticama.

Prepogibanje ponovimo sedemkrat in dobimo krasen dolg vzorec. Ko končamo: na koliko mestih se razlikujeta najbolj različni zaporedni vrstici?

Rešitev

Vsak par sosednjih vrstic se razlikuje na točno enem mestu.

Najprej: upam, da nisi res narisal slike s sedmimi prepogibanji. Vzorec je namreč dolg 256 vrstic!

Vrstici v začetnem vzorcu se očitno razlikujeta na enem samem mestu. Pokazali bomo, da se tudi po prepogibanju vsak par zaporednih vrstic razlikuje le na enem mestu.

- Gornji del pri vsakem prepogibanju ostane enak. Če so se vrstice v njem prej razlikovale na enem mestu, se zdaj tudi.
- Spodnji del je le zrcalna slika gornjega, z eno dodatno piko. Ker imajo vse vrstice v spodnjem delu to dodatno piko, se razlikujejo na toliko mestih, na kolikor so se njihovi originali v gornjem delu – torej na enem samem.
- Ostaneta še vrstici tik nad in pod prepogibom. Gornja vrstica spodnjega dela je slika spodnje vrstice zgornjega dela. Razlikujeta se točno po tem, da smo spodnji dodali še eno piko. Torej se tudi ti dve razlikujeta točno na enem mestu.

Ker se torej prvi vrstici razlikujeta le na enem mestu, bodo to lastnost ohranili tudi vsi prepogibi.

Računalniško ozadje

Če bi vrstice predstavljale dvojiško zapisana števila, bi s sedmimi prepogibi dobili 256 različnih vrstic (kako vemo, da so različne?), ki predstavljajo števila med 0 in 255. Ko gremo prek vseh vrstic, naštejemo vsa števila tako, da vedno spremenimo le en bit.

Triku rečemo Grayevo kodiranje (https://en.wikipedia.org/wiki/Gray_code). Uporabno je na številnih nepričakovanih mestih, od načrtovanja zanesljive strojne opreme do reševanja ugank, kot so kitajski prstani (https://en.wikipedia.org/wiki/Baguenaudier) in Hanojski stolpi (https://en.wikipedia.org/wiki/Tower_of_Hanoi).

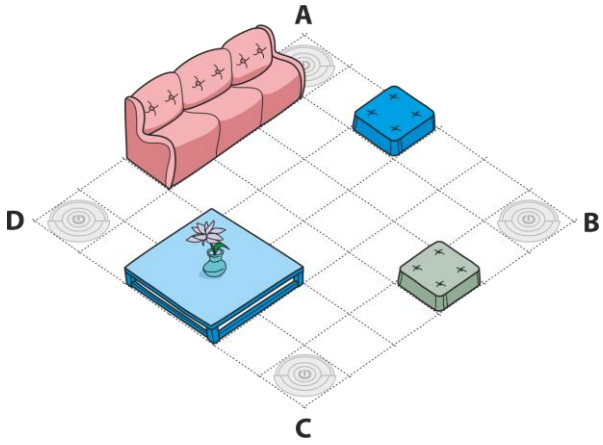
Robotski čistilec

srednja šola, državno

Robot za pomivanje kvadratnih talnih ploščic uporablja naslednje ukaze:

- N – premakni se naprej za eno ploščico (za premik potrebuje 1 minuto),
- D – obrni se za 90° v desno (obrat izvede v trenutku),
- O – operi ploščico (za pranje potrebuje 1 minuto).

Robot začne in konča na ploščici v enem od kotov (A, B, C ali D), a ne nujno na isti ploščici.



Najmanj koliko časa potrebuje robot, da očisti vse ploščice, ki jih ne pokriva pohištvo?

Rešitev

Pravilen odgovor je 55 minut.

V sobi je 36 – 9 = 27 talnih ploščic, torej samo pomivanje ploščic traja 27 minut.

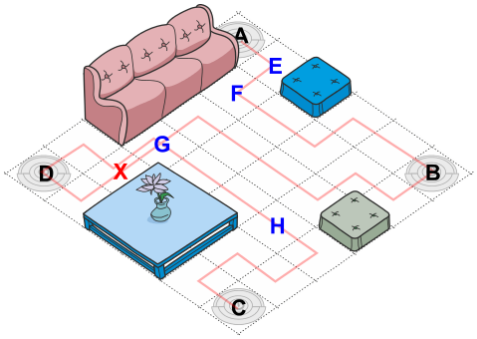
Če bi se robot na vsako ploščico postavil le enkrat, bi za premike potreboval 26 minut (ker se mu ni potrebno premakniti na prvo ploščico). Torej moramo le minimizirati število ploščic, na katere se robot prestavi večkrat.

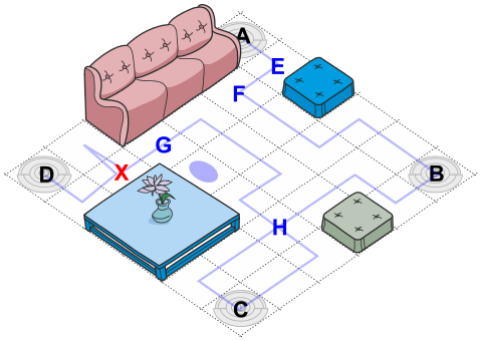
Na levi sliki je z znakom X označena ploščica, na katero se mora robot prestaviti dvakrat, ne glede na to, katero pot ubere. Na sliki so s črkami E, F, G in H označene ploščice, ki so »ozka grla« na poti proti kotnim ploščicam A, B, C in D. Ploščici E in F sta ozko grlo pri A, ploščica G je ozko grlo pri D, ploščica H pa je ozko grlo pri C. Kotna ploščica B nima ozkega grla.

Če ploščica v kotu ni ne začetna ne končna ploščica, se mora robot dvakrat premakniti preko vseh ploščic, ki so ozko grlo te kotne ploščice. Če pa robot začne ali konča v nekem kotu, lahko ozka grla tega kota prehodi le enkrat. Torej moramo pot robota zastaviti tako, da bo robot prehodil čim manj ploščic, ki so ozko grlo.

Če robot začne in konča v istem kotu, potem mora vse ploščice v njegovem ozkem grlu prehoditi dvakrat.

Ker ozko grlo pri A vključuje dve ploščici (E in F), bi bila dobra strategija, da A izberemo za začetno (ali končno) ploščico robota. Nato se moramo odločiti še za končno ploščico, kjer lahko izbiramo med ploščicama C in D (obe imata po eno ploščico na ozkem grlu, to je G oz. H).





Pot, ki je narisana na levi sliki, potrebuje 28 premikov robota naprej (končamo v C, vključuje X in G). Pot od A do D (na sliki desno) pa potrebuje liho število premikov, torej po 28 premikih naprej ni obiskana še najmanj ena ploščica (na desni sliki je označena z modro elipso), zato je ta pot daljša kot pot od A do C.

Najkrajši čas, ki ga robot potrebuje za čiščenje tal, je torej 27 (čiščenje) + 28 (premiki) minut, kar je skupaj 55 minut.

Računalniško ozadje

To je poseben primer obhoda trgovskega potnika – problem, pri katerem moramo poiskati najkrajšo pot in pri tem obiskati vsa vozlišča grafa, ali pa razširjen problem Hamiltonove poti – ali v grafu obstaja pot, ki obišče vsako vozlišče grafa natanko enkrat. Za reševanje teh dveh problemov ne obstaja noben učinkovit algoritem, zato bi iskanje najboljše rešitve naše naloge pri večji sobi zahtevalo veliko časa, čeprav bi imeli na voljo zmogljive računalnike.

Šola v Naughtinghamu

srednja šola, državno

Naughtinghamska šola ima hodnik s petimi vrati ... in kup porednih mulcev. Ko pridejo na hodnik, tečejo mimo učilnic in zaloputnejo vsaka odprta vrata, dokler ne pridejo do zaprtih vrat. Ta, zaprta vrata odprejo, vstopijo ... in jih seveda ne zaprejo za sabo.

Če mulec pride na hodnik, ravno ko je vseh pet vrat odprtih, jih pozaloputa, si čestita in tisti dan šprica šolo.

Na začetku so vsa vrata zaprta.

Vsi vemo, da je tako obnašanje neprimerno, in nihče noče špricati šole. V hipotetičnem primeru, da bi si to vendarle želeli: kateri po vrsti bi morali vstopiti?

Rešitev

Naj . predstavlja zaprta, ○ pa odprta vrata. Recimo, da dijaki vstopajo z desne in tečejo proti levi. Tabela kaže, kaj se dogaja z vrati.

0	. . . . .	Vsa vrata so zaprta.
1	. . . . ○	Že prva vrata so zaprta, zato jih odpre in vstopi.
2	. . . ○ .	Zapre prva, odpre druga.
3	. . . ○○	Odpre prva in vstopi.
4	. . ○ . .	Zaloputne prva, zaloputne druga, odpre tretja in vstopi.
5	. . ○ . ○	Odpre prva in vstopi.
6	. . ○○ .	Zaloputne prva, vstopi skozi druga.
7	. . ○○○	Vstopi skozi prva.
8	. ○ . . .	Zaloputne prva tri in vstopi skozi četrta.

Če te to spominja na dvojiško štetje, imaš prav. Imeli bomo

31	○○○○○

Dvaintrideseti bo zaloputnil vsa vrata in šel na sladoled.

Računalniško ozadje

Zakaj dvojiško štetje deluje tako?

Najprej razmislimo, kako sešteješ 40300996 + 4?

```
  40300996
+        4
= 40301000
```

6 + 4 je 0, 1 prenesemo na naslednjo števko. 9 + 1 je 0, 1 prenesemo naprej ... in tako naprej.

V dvojiškem zapisu je podobno:

```
  01011
+     1
= 01100
```

Zapiranje vrat je kot spreminjanje 1 v 0, kar počnemo, dokler ne naletimo na 0, ki jo spremenimo v 1 in se ustavimo.

Še razpolavljanja ogrlice

srednja šola, državno

Andreja ima zelo dolgo ogrlico, ki jo odpeto sestavlja zaporedje1024 perlic: diamant, kocka, kroglica, diamant, kocka, kroglica ... in tako dalje.

Ker je predolga, jo razpolovi in obdrži le desni del. Ker je še vedno predolga, jo razpolovi in spet obdrži desni del. Ko jo razpolovi tretjič, obdrži levi del.

To ponavlja: vedno po dvakrat obdrži desni del in enkrat levega. Konča, ko ji ostane le še ena perlica. Kakšna je?

Rešitev

Oštevilčimo perlice v začetni ogrlici: prva perlica naj ima številko 0, druga 1, tretja 2 in tako do 1023. Diamanti imajo številke, deljive s 3, pri kockah je ostanek pri deljenju s 3 enak 1, pri kroglicah pa 2. Izvedeti moramo torej številko preostale perlice.

	številka prve perlice	dolžina ogrlice
začetek	0	1024
DESNO	512 (1024 / 2)	512
DESNO	768 (512 + 512 / 2)	256
LEVO	768	128
DESNO	832 (768 + 128 / 2)	64
DESNO	864 (832 + 64 / 2)	32
LEVO	864	16
DESNO	872 (864 + 16 / 2)	8
DESNO	876 (872 + 8 / 2)	4
LEVO	876	2
DESNO	877 (876 + 2 / 2)	1

Ostala ji bo perlica, ki je bila na 877. mestu. Ostanek pri deljenju s 3 je 1, torej gre za kocko.

Računalniško ozadje

Ko smo računali številko preostale perlice, smo pretvarjali število 1101101101 iz dvojiškega zapisa v desetiškega. Zakaj? Vse perlice na levi strani ogrlice imajo prvo števko 0 (v dvojiškem zapisu) in vse na desni strani imajo 1. Ko vzamemo preostanek, imajo vse perlice v levi polovici spet prvo števko 0, druge 1 ...