



Tekmovanje robotov

Raziskovalci umetne inteligence Univerze v Szegedu organizirajo tekmovanje v programiranju robotov. Vaš prijatelj Hanga se je odločil, da bo sodeloval na tekmovanju. Cilj je sprogramirati vrhunski *Pulibot*, ki posnema veliko inteligenco znamenite madžarske pasme pastirskih psov Puli.

Pulibot bo testiran v blodnjaku, sestavljenem iz mreže $(H + 2) \times (W + 2)$ celic. Vrstice mreže so oštevilčene od -1 do H (od severa proti jugu), stolpci mreže pa so oštevilčeni od -1 do W (od zahoda proti vzhodu). Celico, ki se nahaja v r -ti vrstici in c -tem stolpcu mreže ($-1 \leq r \leq H$, $-1 \leq c \leq W$), imenujemo celica (r, c) .

Zamislimo si celico (r, c) , tako da velja $0 \leq r < H$ in $0 \leq c < W$. Celica (r, c) ima **4 sosednje** celice:

- celica $(r, c - 1)$ je celica **zahodno** od celice (r, c) ;
- celica $(r + 1, c)$ je celica **južno** od celice (r, c) ;
- celica $(r, c + 1)$ je celica **vzhodno** od celice (r, c) ;
- celica $(r - 1, c)$ je celica **severno** od celice (r, c) .

Celica (r, c) je **mejna** celica blodnjaka, če velja $r = -1$ ali $r = H$ ali $c = -1$ ali $c = W$. Vsaka celica, ki ni mejna celica, je bodisi **ovira** bodisi **prazna** celica. Poleg tega ima vsaka prazna celica **barvo**, ki jo predstavlja nenegativno število med 0 in vključno Z_{MAX} . Sprva je barva vsake prazne celice enaka 0.

Na primer, upoštevajmo blodnjak z $H = 4$ in $W = 5$, ki vsebuje eno oviro (1, 3):

| | -1 | 0 | 1 | 2 | 3 | 4 | 5 |
|----|----|---|---|---|---|---|---|
| -1 | | | | | | | |
| 0 | | 0 | 0 | 0 | 0 | 0 | |
| 1 | | 0 | 0 | 0 | | 0 | |
| 2 | | 0 | 0 | 0 | 0 | 0 | |
| 3 | | 0 | 0 | 0 | 0 | 0 | |
| 4 | | | | | | | |

Edina ovira je označena s križcem. Mejne celice blodnjaka so senčene. Številka v vsaki prazni celici predstavlja njeno barvo.

Pot dolžine ℓ ($\ell > 0$) od celice (r_0, c_0) do celice (r_ℓ, c_ℓ) je zaporedje paroma različnih *praznih* celic $(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$, v katerem sta za vsak i ($0 \leq i < \ell$) celici (r_i, c_i) in (r_{i+1}, c_{i+1}) sosednji.

Upoštevajte, da pot dolžine ℓ vsebuje natanko $\ell + 1$ celic.

Na tekmovanju so raziskovalci postavili blodnjak, v katerem obstaja vsaj ena pot od celice $(0, 0)$ do celice $(H - 1, W - 1)$. Upoštevajte, da sta celici $(0, 0)$ in $(H - 1, W - 1)$ zagotovo prazni.

Hanga ne ve, katere celice blodnjaka so prazne in v katerih so ovire.

Vaša naloga je pomagati Hangi sprogramirati Pulibota tako, da bo sposoben najti *najkrajšo pot* (to je pot z najmanjšo dolžino) od celice $(0, 0)$ do celice $(H - 1, W - 1)$ v neznanem blodnjaku, ki so ga postavili raziskovalci. Specifikacija Pulibota in pravila tekmovanja so opisane spodaj.

Upoštevajte, da zadnje poglavje te naloge opisuje orodje za prikaz, ki ga lahko uporabite za vizualizacijo Pulibota.

Specifikacija Pulibota

Definirajmo **stanje** celice (r, c) za vsak $-1 \leq r \leq H$ in $-1 \leq c \leq W$ kot celo število, tako da:

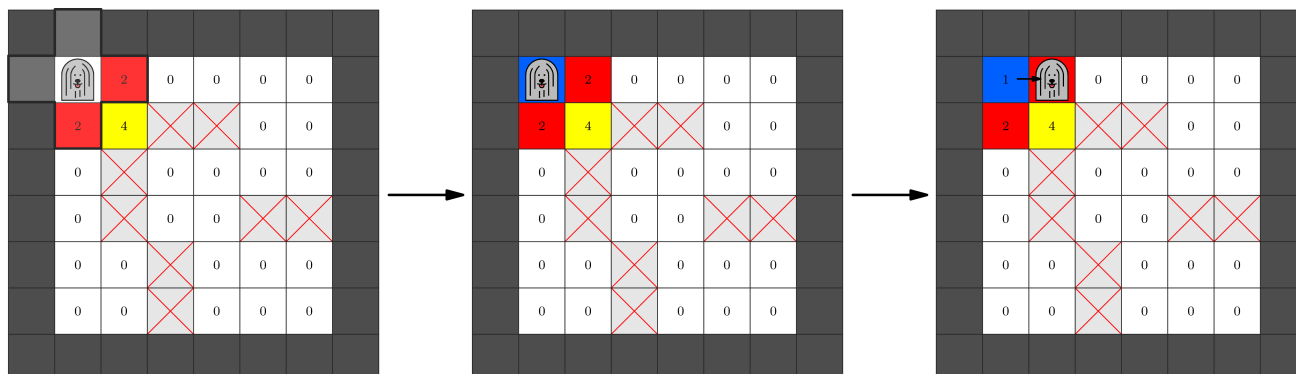
- če je celica (r, c) mejna celica, potem je njeno stanje -2 ;
- če je celica (r, c) ovira, potem je njeno stanje -1 ;
- če je celica (r, c) prazna celica, potem je njeno stanje barva celice.

Program Pulibota se izvaja kot zaporedje korakov. V vsakem koraku Pulibot prepozna stanja bližnjih celic in nato izvede navodilo. Navodilo, ki ga izvede, določajo prepoznana stanja. Sledi natančnejši opis.

Predpostavimo, da je Pulibot na začetku trenutnega koraka v celici (r, c) , ki je prazna celica. Korak se izvede takole:

1. Najprej Pulibot prepozna trenutno **polje stanj**, to je polje $S = [S[0], S[1], S[2], S[3], S[4]]$, ki vsebuje stanje celice (r, c) in vseh sosednjih celic:
 - $S[0]$ je stanje celice (r, c) .
 - $S[1]$ je stanje celice na zahodu.
 - $S[2]$ je stanje celice na jugu.
 - $S[3]$ je stanje celice na vzhodu.
 - $S[4]$ je stanje celice na severu.
2. Nato Pulibot določi **ukaz** (Z, A) , ki ustreza prepoznanim stanjem.
3. Končno Pulibot izvede ta ukaz: nastavi barvo celice (r, c) na barvo Z in nato izvede dejanje A , ki je eno od naslednjih:
 - *ostane* v celici (r, c) ;
 - *premakne* se v eno od 4 sosednjih celic;
 - *konča program*.

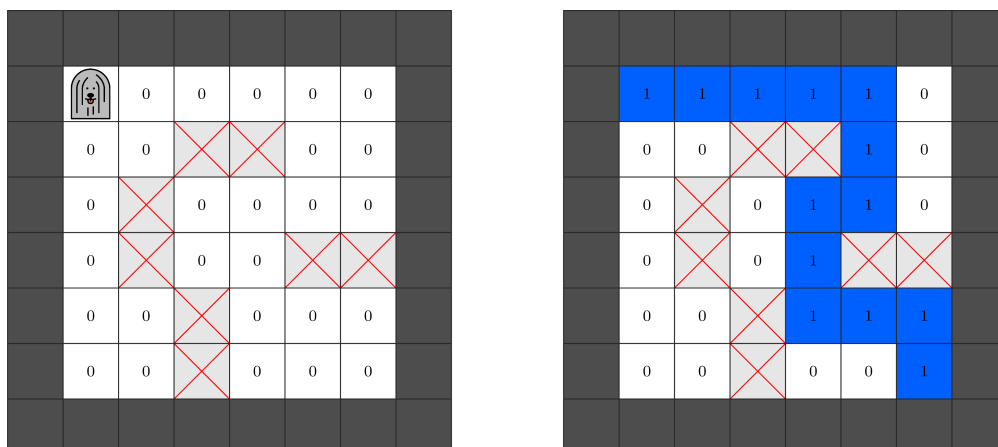
Na primer, upoštevajmo scenarij, prikazan na levi strani naslednje slike. Pulibot trenutno stoji v celici $(0,0)$ z barvo 0. Pulibot prepozna polje stanj $S = [0, -2, 2, 2, -2]$. Pulibot ima lahko program, ki ob razpoznavi tega polja nastavi barvo trenutne celice na $Z = 1$ in se nato premakne proti vzhodu, kot je prikazano na sredini in na desni strani slike:



Pravila tekmovanja z roboti

- Na začetku se Pulibot nahaja v celici $(0,0)$ in prične izvajati svoj program.
- Pulibot se ne sme premakniti v celico, ki ni prazna.
- Program Pulibota se mora končati v največ 500 000 korakih.
- Po končanem programu Pulibota morajo biti prazne celice v blodnjaku obarvane tako, da:
 - obstaja najkrajša pot od $(0,0)$ do $(H-1, W-1)$, pri kateri je barva vsake celice na poti 1.
 - vse ostale prazne celice imajo barvo 0.
- Pulibot lahko svoj program konča v kateri koli prazni celici.

Na primer, naslednja slika prikazuje možen blodnjak s $H = W = 6$. Začetna konfiguracija je prikazana na levi in eno izmed veljavnih barvanj praznih celic po zaključku je prikazano na desni:



Podrobnosti implementacije

Implementirajte naslednjo metodo.

```
void program_pulibot()
```

- Ta metoda predstavlja program Pulibota. Ta program naj bi pravilno deloval za vse vrednosti H in W ter za vsak blodnjak, ki ustreza omejitvam naloge.
- Ta metoda se pokliče natanko enkrat za vsak testni primer.

Ta metoda lahko kliče naslednjo metodo:

```
void set_instruction(int[] S, int Z, char A)
```

- S : polje dolžine 5, ki opisuje polje stanj.
- Z : nenegativno celo število, ki predstavlja barvo.
- A : en sam znak, ki predstavlja dejanje Pulibota, kot sledi:
 - H: ostani;
 - W: premakni se na zahod;
 - S: premakni se na jug;
 - E: premakni se na vzhod;
 - N: premakni se na sever;
 - T: končaj program.
- Klic te metode naroči Pulibotu, da ob razpoznavi polja stanj S izvede navodilo (Z, A) .

Če večkrat pokličete ta postopek z istim stanjem polja S , boste dobili Output `isn't correct`.

Ni potrebno klicati `set_instruction` za vsako možno polje stanj S . Ampak, če Pulibot kasneje razpozna polje stanj, za katero ni bilo določeno navodilo, boste dobili Output `isn't correct`.

Po zaključku `program_pulibot` ocenjevalnik sproži program Pulibota na enem ali večih blodnjakih. Ti klici *ne štejejo* k časovni omejitvi za vašo rešitev. Ocenjevalnik *ni* prilagodljiv, kar pomeni, da je nabor blodnjakov v vsakem testnem primeru vnaprej določen.

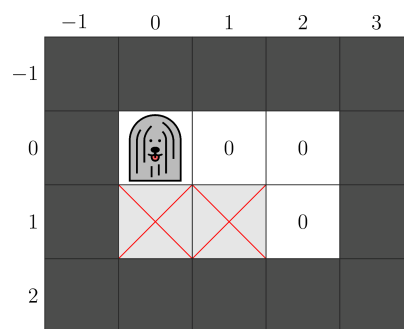
Če Pulibot krši katero koli izmed pravil tekmovanja, preden konča svoj program, boste dobili Output `isn't correct`.

Primer

Metoda `program_pulibot` lahko pokliče `set_instruction` na naslednji način:

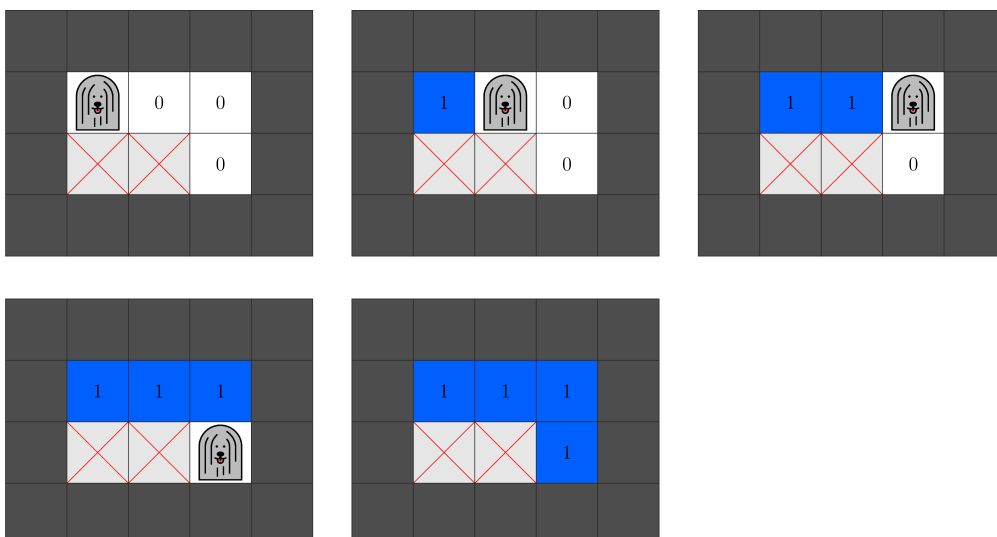
| Klic | Navodilo za polje stanj S |
|--|--|
| <code>set_instruction([0, -2, -1, 0, -2], 1, E)</code> | Nastavi barvo na 1 in se premakni na vzhod |
| <code>set_instruction([0, 1, -1, 0, -2], 1, E)</code> | Nastavi barvo na 1 in se premakni na vzhod |
| <code>set_instruction([0, 1, 0, -2, -2], 1, S)</code> | Nastavi barvo na 1 in se premakni na jug |
| <code>set_instruction([0, -1, -2, -2, 1], 1, T)</code> | Nastavi barvo na 1 in končaj program |

Upoštevajmo scenarij, kjer je $H = 2$ in $W = 3$, blodnjak pa je prikazan na naslednji sliki.



Za ta poseben blodnjak se program Pulibota izvede v štirih korakih. Polja stanj, katera Pulibot prepozna, in ustrezna navodila ustrezajo štirim klicem `set_instruction`, ki so opisani zgoraj, v enakem vrstnem redu. Zadnje od teh navodil konča program.

Naslednja slika prikazuje blodnjak pred vsakim od štirih korakov in končne barve po zaključku.



Vendar pa upoštevajte, da ta program s 4 navodili morda ne bo našel najkrajše poti v drugih veljavnih blodnjakih. Za to bo prejel sodbo `Output isn't correct`, če bo oddan.

Omejitve

$Z_{MAX} = 19$. Torej lahko Pulibot uporablja barve od 0 do vključno 19.

Za vsak blodnjak, uporabljen za testiranje Pulibota, velja:

- $2 \leq H, W \leq 15$
- Obstaja vsaj ena pot od celice $(0, 0)$ do celice $(H - 1, W - 1)$.

Podnaloge

1. (6 točk) V blodnjaku ni ovire.
2. (10 točk) $H = 2$
3. (18 točk) Med vsakim parom praznih celic obstaja natanko ena pot.
4. (20 točk) Vse najkrajše poti od celice $(0, 0)$ do celice $(H - 1, W - 1)$ imajo dolžino $H + W - 2$.
5. (46 točk) Ni dodatnih omejitev.

Če v katerem koli od testnih primerov klici metode `set_instruction` ali program Pulibota med izvajanjem ne ustrezajo omejitvam, opisanim v Podrobnostih implementacije, bo ocena vaše rešitve za to podnalogo 0.

V vsaki podnalogi lahko dobite delno oceno tako, da proizvedete obarvanje, ki je skoraj pravilno.

Formalno:

- Rešitev testnega primera je **popolna**, če končno obarvanje praznih celic ustreza pravilom tekmovanja z roboti.
- Rešitev testnega primera je **delna**, če je končno obarvanje videti takole:
 - Obstaja najkrajša pot od $(0, 0)$ do $(H - 1, W - 1)$, pri kateri je barva vsake celice na poti 1.
 - V mreži ni druge prazne celice z barvo 1.
 - Nekatere prazne celice v mreži imajo barvo, ki ni 0 ali 1.

Če vaša rešitev testnega primera ni ne popolna ne delna, bo vaša ocena za ustrezen testni primer 0.

V podnalogah 1—4 je točkovanje popolne rešitve 100%, točkovanje delne rešitve pa 50% točk za podnalogo.

V podnalogi 5 je vaša ocena odvisna od števila barv, uporabljenih v programu Pulibota. Natančneje, označimo z Z^* največjo vrednost Z izmed vseh klicev `set_instruction`. Ocena testnega primera se izračuna po naslednji tabeli:

| Pogoj | Ocena (popolna) | Ocena (delna) |
|-----------------------|-------------------|-------------------|
| $11 \leq Z^* \leq 19$ | $20 + (19 - Z^*)$ | $12 + (19 - Z^*)$ |
| $Z^* = 10$ | 31 | 23 |
| $Z^* = 9$ | 34 | 26 |
| $Z^* = 8$ | 38 | 29 |
| $Z^* = 7$ | 42 | 32 |
| $Z^* \leq 6$ | 46 | 36 |

Ocena za vsako podnalogo je minimum točk za testne primere v podnalogi.

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod naslednje oblike:

- vrstica 1: $H \ W$
- vrstica $2 + r$ ($0 \leq r < H$): $m[r][0] \ m[r][1] \ \dots \ m[r][W - 1]$

Tukaj je m polje H polj W celih števil, ki opisujejo ne-mejne celice blodnjaka. $m[r][c] = 0$, če je celica (r, c) prazna celica in $m[r][c] = 1$, če je celica (r, c) ovira.

Vzorčni ocenjevalnik najprej pokliče `program_pulibot()`. Če vzorčni ocenjevalnik zazna kršitev protokola, izpiše `Protocol Violation: <MSG>` in zaključi izvajanje. `<MSG>` je eno izmed naslednjih sporočil:

- `Invalid array`: $-2 \leq S[i] \leq Z_{MAX}$ ni izpolnjen za nek i ali dolžina S ni 5.
- `Invalid color`: $0 \leq Z \leq Z_{MAX}$ ne velja.
- `Invalid action`: znak A ni eden izmed H, W, S, E, N ali T.
- `Same state array`: `set_instruction` je bil poklican z istim poljem S vsaj dvakrat.

Sicer, ko se `program_pulibot` konča, vzorčni ocenjevalnik izvede program Pulibota v blodnjaku, ki je opisan na vhodu.

Vzorčni ocenjevalnik ustvari dva izhoda.

Vzorčni ocenjevalnik najprej zapiše dnevnik dejanj Pulibota v datoteko `robot.bin` v trenutni delovni mapi. Ta datoteka služi kot vhod orodju za prikaz, ki je opisano v naslednjem razdelku.

Če se program Pulibota ne zaključi uspešno, vzorčni ocenjevalnik izpiše eno izmed naslednjih sporočil:

- `Unexpected state`: Pulibot je razpoznal polje stanj, za katerega ni bil poklican `set_instruction`.

- `Invalid move`: izvajanje programa je povzročilo, da se je Pulibot premaknil na neprazno celico.
- `Too many steps`: Pulibot je izvedel 500 000 korakov, preden bi se zaključil.

Sicer naj bo $e[r][c]$ stanje celice (r, c) po zaključku programa Pulibota. Vzorčni ocenjevalnik izpiše H vrstic v naslednji obliki:

- Vrstica $1 + r$ ($0 \leq r < H$): $e[r][0] \ e[r][1] \ \dots \ e[r][W - 1]$

Orodje za prikaz

Pripomoček za to nalogo vsebuje datoteko z nazivom `display.py`. Ko ga pokličete, ta Python skripta prikaže dejanja Pulibota v blodnjaku, ki so zapisana v dnevniku dejanj, ki je ustvaril vzorčni ocenjevalnik. Za to mora biti binarna datoteka `robot.bin` prisotna v trenutni delovni mapi.

Za klic skripte izvedite naslednji ukaz:

```
python3 display.py
```

Prikaže se preprost grafični vmesnik. Glavne značilnosti so naslednje:

- Lahko opazujete stanje celotnega blodnjaka. Trenutna lokacija Pulibota je označena s pravokotnikom.
- Po korakih Pulibota se lahko sprehajate ali z uporabo gumbov s puščicami, lahko pa tudi z uporabo tipkovnice. Možno je tudi skočiti na poljuben korak.
- Na dnu je prikazan naslednji korak programa. Prikazuje trenutno polje stanj in navodilo, ki se bo izvedlo. Po zadnjem koraku prikaže bodisi eno od sporočil o napakah, bodisi `Terminated`, če se program uspešno konča.
- Vsaki številki, ki predstavlja barvo, lahko dodelite tako vizualno barvo ozadja kakor tudi besedilno oznako. Oznaka je kratek niz, ki se bo pojavil v vsaki celici s to barvo. Barve in oznake lahko dodelite na eden izmed naslednjih načinov:
 - Nastavite jih v pogovornem oknu, dosegljivem preko gumba `Colors`.
 - Uredite vsebino datoteke `colors.txt`.
- Za ponovno nalaganje `robot.bin`, uporabite gumb `Reload`. To je uporabno, če se vsebina `robot.bin` spremeni.