



Natjecanje robota

Istraživači umjetne inteligencije na Sveučilištu u Szegedu održavaju natjecanje u programiranju robota. Vaša prijateljica, Mila, odlučila je sudjelovati na natjecanju. Njezin cilj je isprogramirati ultimativnog *Pulibot*-a, diveći se inteligenciji poznate mađarske pasmine, Puli.

Pulibot će biti isproban na labirintu koji se sastoji od $(H + 2) \times (W + 2)$ tablice polja. Retci tablice označeni su od -1 do H u smjeru sjever jug, a stupci od -1 do W u smjeru zapad istok. Polje koje se nalazi u retku r i stupcu c ($-1 \leq r \leq H$, $-1 \leq c \leq W$) označavat ćemo sa (r, c) .

Promatrajmo polje (r, c) takvo da je $0 \leq r < H$ i $0 \leq c < W$. Postoje 4 **susjedna** polja polju (r, c) :

- polje $(r, c - 1)$ se označava kao polje **zapadno** od polja (r, c) ;
- polje $(r + 1, c)$ se označava kao polje **južno** od polja (r, c) ;
- polje $(r, c + 1)$ se označava kao polje **istočno** od polja (r, c) ;
- polje $(r - 1, c)$ se označava kao polje **sjeverno** od polja (r, c) ;

Polje (r, c) se naziva **okvirnim** poljem labirinta ako je $r = -1$ ili $r = H$ ili $c = -1$ ili $c = W$. Svako polja koje nije okvirno je ili **prazno** ili sadrži **prepreku**. Dodatno, svako prazno polje ima svoju **boju**, označenu kao nenegativan broj između 0 i Z_{MAX} , uključivo. Na početku, boja svakog praznog polja je 0.

Primjerice, promatrajmo labirint s $H = 4$ i $W = 5$, koji sadrži točno jednu prepreku na polju $(1, 3)$:

	-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							

Jedina prepreka je označena prekrštenim poljem. Okvirna polja su zacrnjena. Brojevi napisani u svakom praznom polju predstavljaju pripadajuću boju.

Put duljine ℓ ($\ell > 0$) od polja (r_0, c_0) do polja (r_ℓ, c_ℓ) je niz različitih *praznih* polja $(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$ u kojem za svaki i ($0 \leq i < \ell$) polja (r_i, c_i) i (r_{i+1}, c_{i+1}) su susjedna.

Primijetite da put duljine ℓ sadrži točno $\ell + 1$ polje.

Na natjecanju, istraživači su postavili labirint u kojem postoji barem jedan put od polja $(0, 0)$ do polja $(H - 1, W - 1)$. Primijetite da to znači da su polja $(0, 0)$ i $(H - 1, W - 1)$ sigurno prazna.

Mila ne zna koja su polja labirinta prazna, a u kojima se nalaze prepreke.

Vaš je zadatak pomoći Mili napisati program za Pulibota koji je sposoban pronaći *najkraći put* (točnije, put koji je minimalne duljine) od polja $(0, 0)$ do polja $(H - 1, W - 1)$ u njoj nepoznatom labirintu kojeg su pripremili istraživači. Detalji Pulibota i pravila natjecanja nalaze se ispod.

Detalji Pulibota

Definirajmo **stanje** polja (r, c) za svaki $-1 \leq r \leq H$ i $-1 \leq c \leq W$ kao cijeli broj takav da

- ako je polje (r, c) okvirno, onda je stanje -2 ;
- ako je polje prepreka, onda je stanje -1 ;
- ako je polje prazno, onda je stanje boja polja.

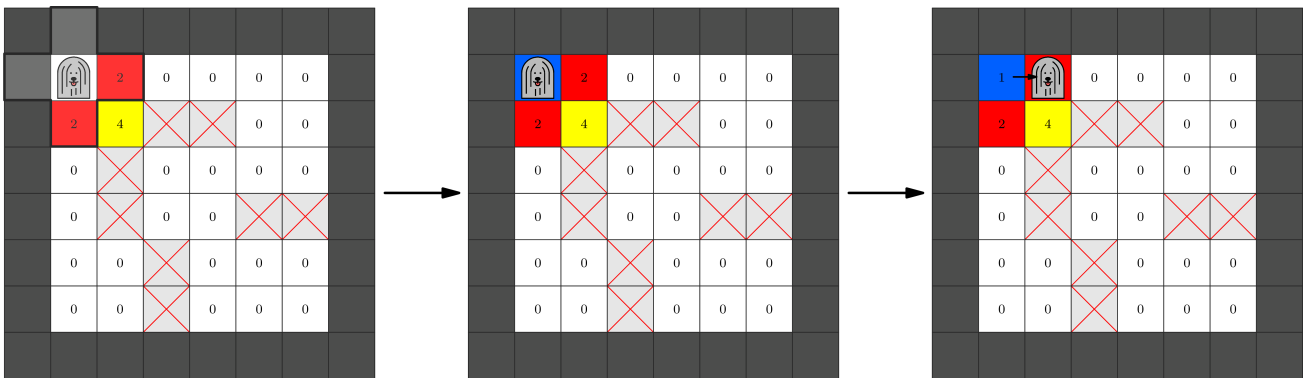
Program Pulibota pokreće se u nekoliko koraka. U svakom koraku, Pulibot prepoznaje stanje okolna polja te odradi jednu instrukciju. Instrukcija je određena prepoznatim stanjima. Precizniji opis slijedi:

Pretpostavimo da na početku trenutnog koraka, Pulibot se nalazi u polju (r, c) , koje je prazno, Korak se odvija na sljedeći način:

1. Prvo, Pulibot prepoznaje trenutni **niz stanja**, točnije niz $S = [S[0], S[1], S[2], S[3], S[4]]$, koji se sastoji od stanja polja (r, c) i susjednih polja:
 - $S[0]$ je stanje polja (r, c) .
 - $S[1]$ je stanje polja zapadno od polja (r, c)
 - $S[2]$ je stanje polja južno od polja (r, c)
 - $S[3]$ je stanje polja istočno od polja (r, c)
 - $S[4]$ je stanje polja sjeverno od polja (r, c)
2. Onda, Pulibot određuje **instrukciju** (Z, A) koje pripada prepoznatom polju stanja.
3. Konačno, Pulibot čini instrukciju: postavlja boju polja (r, c) na boju Z i čini akciju A , koja je jedno od sljedećih mogućnosti:
 - *ostaje* na polju (r, c) ;
 - *pomiče* se na jedno od 4 susjedna polja;
 - *završava izvršavanje programa*

Primjerice, promatrajte slučaj prikazan na lijevoj slici. Pulibot je trenutno na polju $(0, 0)$ s bojom 0. Pulibot prepoznaje niz stanja $S = [0, -2, 2, 2, -2]$. Pulibot možda ima program koji, pri

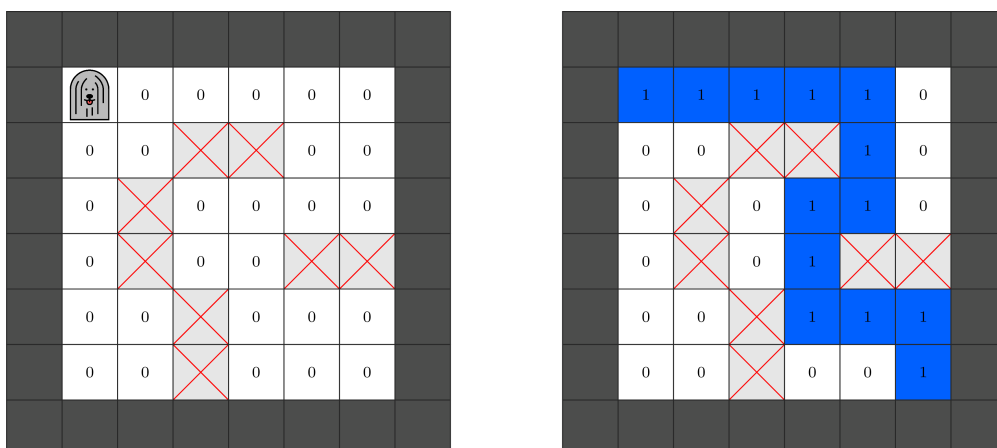
prepoznavanju niza, postavlja boju trenutnog polja na $Z = 1$ i pomiče se istočno, kao što je prikazano na srednjoj i desnoj slici.



Pravila natjecanja robota

- Na početku, Pulibot se nalazi na polju $(0, 0)$ i započinje s izvršavanjem programa.
- Pulibotu nije dozvoljeno kretanje na polja koja nisu prazna.
- Pulibotov program mora završiti s izvršavanjem nakon najviše 500 000 koraka.
- Nakon završetka izvršavanja Pulibotovog programa, prazna polja labirinta moraju biti obojana tako da:
 - Postoji najkraći put od polja $(0, 0)$ do polja $(H - 1, W - 1)$ tako da je boja svih uključenih polja 1.
 - Sva druga prazna polja imaju boju 0.
- Pulibot može završiti izvršavanje svog programa u bilo kojem praznom polju.

Primjerice, na sljedećoj slici prikazan je mogući labirint s $H = W = 6$. Početna postava prikazana na lijevoj slici, a jedno moguće bojanje praznih polja nakon završetka izvršavanja programa na desnoj slici.



Implementacijski detalji

Morate implementirati sljedeće funkcije.

```
void program_pulibot()
```

- Ova funkcija treba napraviti Pulibotov program. Program mora biti točan za sve moguće vrijednosti H i W i sve moguće labirinte koji zadovoljavaju ograničenja.
- Ova funkcija će se pozvati točno jednom po primjeru.

Ova funkcija smije zvati sljedeću funkciju kako bi napravila Pulibotov program:

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

- S : niz duljine 5 koji opisuje niz stanja
- Z : nenegativan cijeli broj koji opisuje boju
- A : jedan znak koji opisuje akciju Pulibota koja slijedi:
 - H: ostani;
 - W: pomakni se na zapad;
 - S: pomakni se na jug;
 - E: pomakni se na istok;
 - N: pomakni se na sjever;
 - T: završi s izvršavanjem programa.
- Pozivanje ove funkcije naređuje Pulibotu da ako prepozna niz stanja S onda treba odraditi instrukciju (Z, A) .

Pozivanje ove funkcije više puta s istim nizom stanja S biti će evaluirano kao `Output isn't correct`.

Nije potrebno pozvati funkciju `set_instruction` sa svim mogućim nizovima stanja S . Ali, ako Pulibot prepozna niz stanja za koji instrukcija nije određena, dobit ćete `Output isn't correct` evaluaciju.

Nakon što `program_pulibot` završi s izvršavanjem, ocjenjivač će pokrenuti Pulibotov program na jednom ili više labirinata. Ti pozivi se *neće* računati u vremensko ograničenje vašeg rješenja. Ocjenjivač *nije* adaptivan, točnije, skup labirinata je unaprijed određen u svakom primjeru.

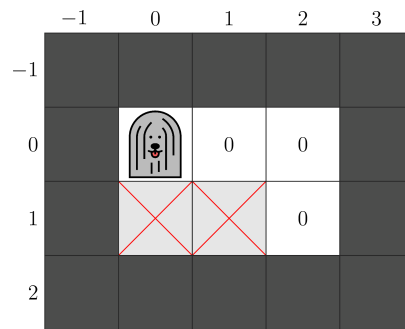
Ako Pulibot prekrši ijedno od pravila natjecanja prije nego što program završi s izvršavanjem, Vaše rješenje bit će evaluirano kao `Output isn't correct`.

Primjer

Funkcija `program_pulibot` može pozvati `set_instruction` na sljedeći način:

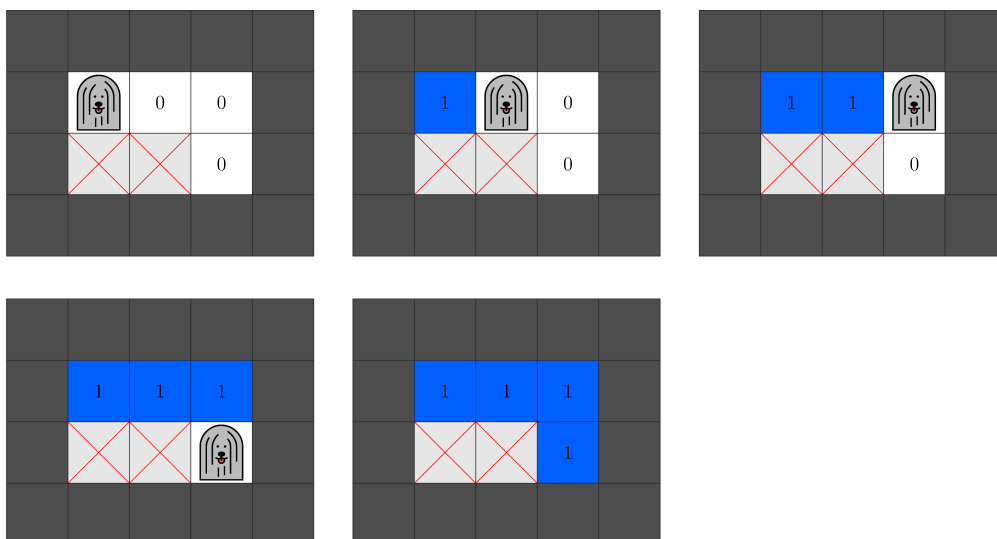
Poziv	Instrukcija za niz stanja S
<code>set_instruction([0, -2, -1, 0, -2], 1, E)</code>	Postavi boju na 1 i pomakni se istočno
<code>set_instruction([0, 1, -1, 0, -2], 1, E)</code>	Postavi boju na 1 i pomakni se istočno
<code>set_instruction([0, 1, 0, -2, -2], 1, S)</code>	Postavi boju na 1 i pomakni se južno
<code>set_instruction([0, -1, -2, -2, 1], 1, T)</code>	Postavi boju na 1 i završi s izvršavanjem.

Promatrajmo slučaj gdje je $H = 2$ i $W = 3$, te labirint je postavljen kao na sljedećoj slici.



Za ovaj labirint, Pulibotov program završava izvršavanje u četiri koraka. Niz stanja koja Pulibot prepoznaje i pripadajuće instrukcije koje odradi točno su četiri poziva `set_instruction` gore prikazana, tim redoslijedom. Zadnja od tih instrukcija završava izvršavanje.

Sljedeća slika prikazuje labirint nakon svakog od četiri koraka i konačne boje nakon završetka izvršavanja.



Iako, primijetite da ovaj program od 4 instrukcije možda neće pronaći najkraći put za sve labirinte koji zadovoljavaju dana ograničenja. Stoga, ako poslano, program će biti evaluiran kao `Output isn't correct`.

Ograničenja

$Z_{MAX} = 19$. Stoga, Pulibot može koristiti boje od 0 do 19, uključivo.

Za svaki labirint na kojem se pokreće Pulibot:

- $2 \leq H, W \leq 15$
- Postoji barem jedan put od polja $(0, 0)$ do polja $(H - 1, W - 1)$.

Podzadaci

1. (6 bodova) Postoji točno jedna prepreka.
2. (10 bodova) $H = 2$
3. (18 bodova) Postoji točno jedan put između svaka dva prazna polja.
4. (20 bodova) Svaki najkraći put od $(0, 0)$ do $(H - 1, W - 1)$ ima duljinu točno $H + W - 2$.
5. (46 bodova) Nema dodatnih ograničenja.

Ako, u bilo kojem od primjera, pozivi funkcije `set_instruction` ili Pulibotov program tijekom svog izvršavanja, ne zadovoljavaju sva ograničenja opisana u Implementacijskim detaljima, broj bodova Vašeg rješenja na tom podzadatku biti će 0.

U svakom podzadatku, moguće je dobiti parcijalne bodovi ako je završno bojanje skoro točno.

Točnije:

- Rješenje za primjer je **potpuno** ako konačno bojanje svih praznih polja zadovoljava pravila natjecanja.
- Rješenje za primjer je **parcijalno** ako konačno bojanje ima sljedeća svojstva:
 - Postoji najkraći put od polja $(0, 0)$ do polja $(H - 1, W - 1)$ tako da je boja svih polja na putu 1.
 - Ne postoji drugo prazno polje s bojom 1.
 - Neka prazna polja mogu imati boju koja nije ni 0 ni 1.

Ako vaše rješenje na nekom primjeru nije ni potpuno ni parcijalno, tada je broj bodova na pripadajućem primjeru 0.

U podzadacima 1-4, broj bodova koji nosi potpuno rješenje je 100%, a broj bodova koji nosi parcijalno rješenje je 50%, od broja bodova na tom podzadatku.

U podzadatku 5, broj bodova ovisi o broju boja koje koristi Pulibotov program. Točnije, neka je Z^* najveća vrijednost broja Z po svim pozivima funkcije `set_instruction`. Broj bodova na tom primjeru izračunat je po sljedećoj tablici:

Uvjet	Broj bodova (potpuno)	Broj bodova (parcijalno)
$11 \leq Z^* \leq 19$	$20 + (19 - Z^*)$	$12 + (19 - Z^*)$
$Z^* = 10$	31	23
$Z^* = 9$	34	26
$Z^* = 8$	38	29
$Z^* = 7$	42	32
$Z^* \leq 6$	46	36

Broj bodova na podzadatku minimum je broja bodova po svim primjerima podzadatka.

Probni ocjenjivač

Probni ocjenjivač učitava ulaz u sljedećem obliku:

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

Ovdje, m je niz od H nizova W brojeva, koji opisuju neokvirna polja labirinta. $m[r][c] = 0$ ako je polje (r, c) prazno i $m[r][c] = 1$ ako je na polju (r, c) prepreka.

Probni ocjenjivač prvo poziva `program_pulibot()`. Ako probni ocjenjivač primijeti kršenje protokola, ispisuje `Protocol Violation: <MSG>` i završava s izvršavanjem, gdje `<MSG>` označava jednu od sljedećih poruka.

- `Invalid array`: $-2 \leq S[i] \leq Z_{MAX}$ nije zadovoljeno za neki i ili duljina od S nije 5.
- `Invalid color`: $0 \leq Z \leq Z_{MAX}$ ne vrijedi.
- `Invalid action`: znak A nije jedan od H, W, S, E, N ili T.
- `Same state array`: `set_instruction` pozvana je za isti niz stanja S barem dvaput.

Inače, kad `program_pulibot` završi s izvršavanjem, probni ocjenjivač pokreće Pulibotov program na labirintu opisanom u ulazu.

Probni ocjenjivač ispisuje dva izlaza,

Prvo, ocjenjivač piše zapis Pulibotovih akcija u datoteku `robot.bin` u trenutnoj mapi. Ta datoteka služi kao ulaz za vizualizacijski alat opisan u sljedećem odjeljku.

Drugo, ako Pulibotov program ne završi izvršavanje uspješno, ocjenjivač ispisuje jednu od sljedećih poruka greške:

- `Unexpected state`: Pulibot prepoznaje niz stanja s kojim funkcija `set_instruction` nije pozvana.

- `Invalid move`: ako Pulibot odradi akciju završit će u polju koje nije prazno.
- `Too many steps`: Pulibot izvršava preko 500 000 koraka bez završetka izvršavanja.

Inače, neka je $e[r][c]$ stanje polja (r, c) nakon što Pulibot završi s izvršavanjem. Probni ocjenjivač ispisuje H redaka u sljedećem obliku:

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

Alat za vizualizaciju

U prilogu je paket koji sadrži datoteku pod imenom `display.py`. Kad je pozvana, Python skripta prikazuje Pulibotove akcije u labirintu opisane u ulazu ocjenjivača. Za to, binarna datoteka `robot.bin` mora biti prisutna u radnoj mapi.

Kako biste pozvali skriptu, pokrenite sljedeću naredbu.

```
python3 display.py
```

Pojavit će se jednostavno grafičko sučelje. Glavne mogućnosti su sljedeće:

- Možete promatrati stanje cijelog labirinta. Trenutna pozicija Pulibota označena je pravokutnikom.
- Možete prolaziti kroz Pulibotove korake pritiskavši tipke za strelice ili vruće tipke. Također se možete odmah pomaknuti do nekog specifičnog koraka.
- Sljedeći Pulibotov korak u programu prikazan je na dnu. Prikazuje trenutni niz stanja i instrukciju koju će odraditi. Nakon završnog koraka, prikazuje jednu od poruka grešaka ocjenjivača, ili `Terminated` ako je program uspješno završio s izvršavanjem.
- Svakom broju koji predstavlja boju, moguće je pridružiti neku pozadinsku boju, kao i neki tekst. Odabrani tekst je kratak niz znakova koji će se pojaviti na svakom polju te boje. Moguće je odabrati boju i tekst na jedan od sljedećih načina:
 - Postaviti ih u posebnom prozoru nakon odabira gumba `Colors`.
 - Urediti sadržaj datoteke `colors.txt`.
- Kako biste ponovno učitali `robot.bin`, iskoristite `Reload` gumb. Korisno je ako se sadržaj datoteke `robot.bin` promijenio.