



Robot Contest

Istraživači vještačke inteligencije na Univerzitetu u Segedinu održavaju takmičenje u programiranju robota. Tvoja prijateljica, Marija, odlučila je da učestvuje u takmičenju. Cilj je programirati ultimativnog *Pulibota*, diveći se velikoj inteligenciji poznate pasmine mađarskih pastirskih pasa, Puli.

Pulibot će biti testiran na lavirintu koji se sastoji od mreže ćelija dimenzija $(H + 2) \times (W + 2)$. Redovi mreže su numerisani brojevima od -1 do H od sjevera prema jugu, a kolone mreže su numerisane brojevima od -1 do W od zapada prema istoku. Ćeliju koja se nalazi u redu r i koloni c ($-1 \leq r \leq H$, $-1 \leq c \leq W$) obelježavamo sa (r, c) .

Razmotrimo ćeliju (r, c) takvu da je $0 \leq r < H$ i $0 \leq c < W$. Postoje 4 ćelije **susjedne** ćeliji (r, c) :

- ćelija $(r, c - 1)$ **zapadno** od ćelije (r, c) ;
- ćelija $(r + 1, c)$ **južno** od ćelije (r, c) ;
- ćelija $(r, c + 1)$ **istočno** od ćelije (r, c) ;
- ćelija $(r - 1, c)$ **sjeverno** od ćelije (r, c) .

Ćelija (r, c) se naziva **granična** ćelija lavirinta ako važi $r = -1$ ili $r = H$ ili $c = -1$ ili $c = W$. Svaka ćelija koja nije granična ćelija lavirinta je ili **ćelija sa preprekama** ili **prazna** ćelija. Dodatno, svaka prazna ćelija ima **boju**, predstavljenu nenegativnim cijelim brojem između 0 i Z_{MAX} , uključujući. U početku, boja svake prazne ćelije je 0.

Na primjer, posmatrajmo lavirint dimenzija $H = 4$ i $W = 5$, koji sadrži jednu ćeliju s preprekom u ćeliji $(1, 3)$:

	-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 ćelija s preprekom je označena iksićem. Granične ćelije lavirinta su osijenčene. Broj upisan u svaku praznu ćeliju predstavlja njenu boju.

Put dužine ℓ ($\ell > 0$) od ćelije (r_0, c_0) do ćelije (r_ℓ, c_ℓ) je niz u parovima različitih *praznih* ćelija $(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$ takvih da za svako i ($0 \leq i < \ell$) ćelije (r_i, c_i) i (r_{i+1}, c_{i+1}) su susjedne.

Imajte na umu da put dužine ℓ sadrži tačno $\ell + 1$ ćelija.

Na takmičenju su istraživači postavili lavirint u kojem postoji barem jedan put od ćelije $(0, 0)$ do ćelije $(H - 1, W - 1)$. Primjetite da ovo implicira da su ćelije $(0, 0)$ i $(H - 1, W - 1)$ zagarantovano prazne.

Marija ne zna koje su ćelije lavirinta prazne, a koje su sa preprekama.

Vaš zadatak je pomoći Mariji da programira Pulibota tako da bude sposoban pronaći *najkraći put* (tj. put minimalne dužine) od ćelije $(0, 0)$ do ćelije $(H - 1, W - 1)$ u nepoznatom lavirintu koji su postavili istraživači. Specifikacije Pulibota i pravila takmičenja su opisani u nastavku.

Obratite pažnju da posljenja sekcija postavke zadatka opisuje alat za vizualizaciju koji možete da koristite da vizualizirate Pulibota.

Pulibotova specifikacija

Definišimo **stanje** ćelije (r, c) ($-1 \leq r \leq H, -1 \leq c \leq W$) kao cijeli broj tako da:

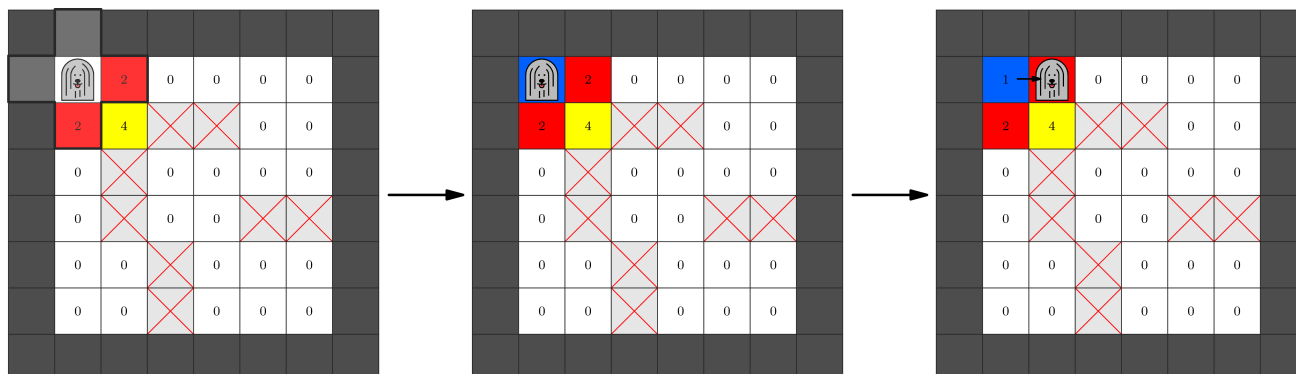
- ako je ćelija (r, c) granična ćelija tada je njeno stanje -2 ;
- ako je ćelija (r, c) ćelija sa preprekom onda je njeno stanje -1 ;
- ako je ćelija (r, c) prazna ćelija onda je njeno stanje predstavlja boju ćelije.

Pulibotov program se izvršava kroz niz koraka. U svakom koraku, Pulibot prepoznaje stanja obližnjih ćelija pa zatim izvršava instrukciju. Instrukcija koju izvršava je određena stanjima obližnjih ćelija. Slijedi precizniji opis.

Pretpostavimo da je na početku trenutnog koraka Pulibot u ćeliji (r, c) , koja je prazna ćelija. Korak se izvodi na sljedeći način:

1. Prvo, Pulibot provjeri trenutni **niz stanja**, to jest, niz $S = [S[0], S[1], S[2], S[3], S[4]]$, koji se sastoji od stanja ćelije (r, c) i svih susjednih ćelija:
 - $S[0]$ je stanje ćelije (r, c) .
 - $S[1]$ je stanje ćelije na zapadu.
 - $S[2]$ je stanje ćelije na jugu.
 - $S[3]$ je stanje ćelije na istoku.
 - $S[4]$ je stanje ćelije na sjeveru.
2. Zatim, Pulibot određuje **instrukciju** (Z, A) koja odgovara prepoznatom nizu stanja.
3. Konačno, Pulibot izvodi odgovarajuću instrukciju: postavlja boju ćelije (r, c) na boju Z i izvodi akciju A , koja je jedna od sljedećih akcija:
 - *ostani* u ćeliji (r, c) ;
 - *premjestiti se* u jednu od 4 susjedne ćelije;
 - *završi program*.

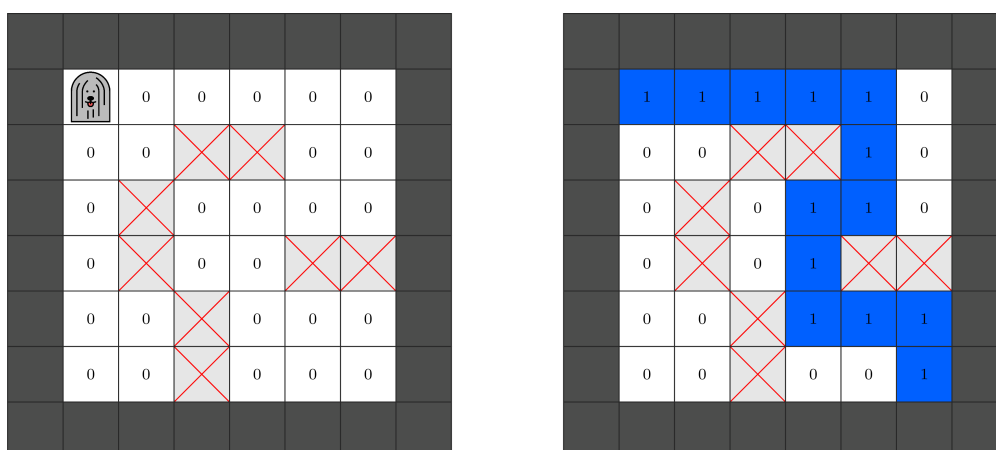
Na primjer, razmotrimo slučaj prikazan na lijevoj strani sljedeće slike. Pulibot je trenutno u ćeliji $(0,0)$ sa bojom 0. Pulibot prepoznaje niz stanja $S = [0, -2, 2, 2, -2]$. Pulibot može imati program koji, nakon prepoznavanja ovog niza, postavlja boju trenutne ćelije na $Z = 1$, a zatim se pomiče na istok, kao što je prikazano u sredini i desno od slike:



Pravila takmičenja robota

- Na početku, Pulibot se postavlja u ćeliju $(0,0)$ i počinje da izvršava svoj program.
- Pulibotu nije dozvoljeno da se pomjeri u ćeliju koja nije prazna.
- Pulibotov program se mora završiti nakon najviše 500 000 koraka.
- Nakon završetka Pulibotovog programa, prazne ćelije u lavirintu treba da budu obojene tako da:
 - Postoji najkraći put od $(0,0)$ do $(H-1, W-1)$ za koji je boja svake ćelije uključene u putanju 1.
 - Sve ostale prazne ćelije imaju boju 0.
- Pulibot može prekinuti svoj program u bilo kojoj praznoj ćeliji.

Na primjer, sljedeća slika prikazuje mogući lavirint sa dimenzijama $H = W = 6$. Početna konfiguracija je prikazana na lijevoj strani, a jedno prihvatljivo bojenje praznih ćelija nakon završetka je prikazano na desnoj strani:



Detalji implementacije

Potrebno je da implementirate sljedeću funkciju.

```
void program_pulibot()
```

- Ova funkcija bi trebala generisati Pulibotov program. Ovaj program bi trebao raditi ispravno za sve vrijednosti H i W i svaki lavirint koji ispunjava ograničenja zadatka.
- Ova funkcija se poziva tačno jednom za svaki test slučaj.

Ova funkcija može pozvati sljedeću funkciju za generisanje Pulibotovog programa:

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

- S : niz dužine 5 koji opisuje niz stanja.
- Z : nenegativan cijeli broj koji predstavlja boju.
- A : jedan znak koji predstavlja akciju Pulibota kako slijedi:
 - H: ostani u trenutnoj ćeliji;
 - W: pomjeri se na zapad;
 - S: pomjeri se na jug;
 - E: pomjeri se na istok;
 - N: pomjeri se na sjever;
 - T: završi program.
- Pozivanje ove funkcije upućuje Pulibot-a da nakon prepoznavanja niza stanja S treba izvršiti instrukciju (Z, A) .

Pozivanje ove funkcije više puta sa istim nizom stanja S rezultiraće izlazom 'Output isn't correct'.

Nije potrebno pozvati `set_instruction` sa svakim mogućim nizom stanja S . Međutim, ako Pulibot kasnije prepozna niz stanja za koje instrukcija nije postavljena, dobićete izlaz 'Output isn't correct'.

Nakon što se `program_pulibot` završi, grader poziva Pulibotov program nad jednim ili više lavirinta. Ovi pozivi *ne* se ubrajaju u vremensko ograničenje za vaše rješenje. Grejder *nije* adaptivan, to jest, skup lavirinta je unapred definisan u svakom testnom slučaju.

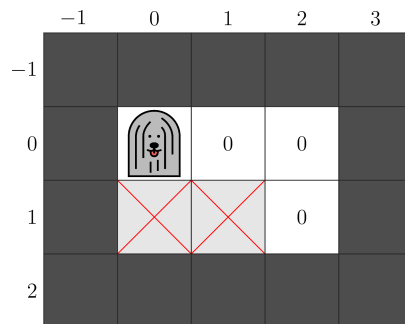
Ako Pulibot prekrši bilo koje od Pravila takmičenja robota prije nego što prekine svoj program, dobićete izlaz 'Output isn't correct'.

Primjer

funkcija `program_pulibot` može pozivati `set_instruction` na sljedeći način:

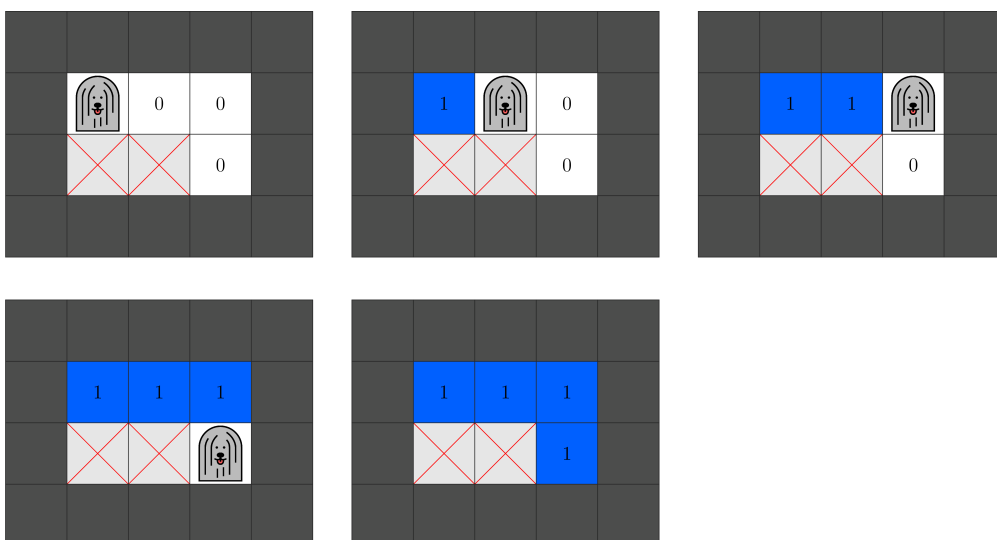
Pozovi	Instrukcija za niz stanja S
<code>set_instruction([0, -2, -1, 0, -2], 1, E)</code>	Postavi boju na 1 i pomakni se istočno
<code>set_instruction([0, 1, -1, 0, -2], 1, E)</code>	Postavi boju na 1 i pomakni se istočno
<code>set_instruction([0, 1, 0, -2, -2], 1, S)</code>	Postavi boju na 1 i pomakni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

Razmotrimo slučaj gdje je $H = 2$ i $W = 3$, a lavirint je prikazan na sljedećoj slici.



Za ovaj konkretan lavirint Pulibotov program se izvršava u četiri koraka. Nizovi stanja koja Pulibot prepoznaje odgovaraju pozivima funkcije `set_instruction` prikazanih u tabeli. Posljednja od ovih instrukcija prekida program.

Sljedeća slika prikazuje stanje lavirinta prije svakog od četiri koraka i njegovo konačno stanje nakon završetka.



Međutim, imajte na umu da ovaj program od 4 instrukcije možda neće pronaći najkraći put u drugim validnim lavirintima. Stoga, ako se submituje, dobiće poruku 'Output isn't correct'.

Ograničenja

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

Za svaki lavirint korišten za testiranje Pulibota:

- $2 \leq H, W \leq 15$
- Postoji najmanje jedna putanja od ćelije $(0, 0)$ do ćelije $(H - 1, W - 1)$.

Podzadaci

1. (6 bodova) U lavirintu nema ćelije s preprekama.
2. (10 bodova) $H = 2$
3. (18 bodova) Postoji tačno jedan put između svakog para praznih ćelija.
4. (20 bodova) Svaki najkraći put od ćelije $(0, 0)$ do ćelije $(H - 1, W - 1)$ ima dužinu $H + W - 2$.
5. (46 bodova) Nema dodatnih ograničenja.

Ako, u bilo kojem od test slučajeva, pozivi funkciji `set_instruction` ili Pulibotovom programu tokom njenog izvršavanja nisu u skladu s ograničenjima opisanim u Detaljima implementacije, vaše rješenje će biti ocijenjeno sa 0 bodova na tom podzadatku.

U svakom podzadatku možete dobiti dio bodova tako što ćete naći skoro tačno bojenje.

Formalno:

- Rješenje test slučaja je **potpuno** ako konačno bojenje praznih ćelija zadovoljava Pravila takmičenja robota.
- Rješenje testnog slučaja je **djelimično** ako konačno bojanje izgleda ovako:
 - Postoji najkraći put od $(0, 0)$ do $(H - 1, W - 1)$ za koji je boja svake ćelije uključene u putanju 1.
 - Nema druge prazne ćelije u mreži sa bojom 1.
 - Neka prazna ćelija u mreži ima boju koja nije 0 i 1.

Ako vaše rješenje test primjera nije ni potpuno ni djelimično, vaše rješenje će biti ocijenjeno sa 0 bodova na tom test primjeru.

U podzadacima 1-4, potpuno rješenje će biti ocijenjeno sa 100%, a djelimično rješenje će biti ocijenjeno sa 50% bodova predviđenih za taj podzadatak.

U podzadatku 5, vaš rezultat zavisi od broja boja korištenih u Pulibotovom programu. Preciznije, označite sa Z^* maksimalnu vrijednost Z za sve pozive funkcije `set_instruction`. Rezultat test primjera se izračunava:

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

Rezultat za svaki podzadatak je minimum bodova za test slučajeve u tom podzadatku.

Sample Grader

The sample grader čita ulaz u sledećem formatu:

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

m predstavlja niz od H nizova dužine W koji opisuju negranične ćelije lavirinta. $m[r][c] = 0$ ako je ćelija (r, c) prazna i $m[r][c] = 1$ ako je ćelija (r, c) ćelija prepreka.

Sample grader prvo poziva `program_pulibot()`. Ako sample grader otkrije kršenje protokola, ispisuje se poruka `Protocol Violation: <MSG>` i završava, gdje je ` jedna od sljedećih poruka o grešci:

- `Invalid array`: $-2 \leq S[i] \leq Z_{MAX}$ nije zadovoljeno i ili dužuna za S nije 5.
- `Invalid color`: $0 \leq Z \leq Z_{MAX}$ nije zadovoljeno.
- `Invalid action`: karakter A nije ni jedan od karaktera H, W, S, E, N, T.
- `Same state array`: `set_instruction` je pozvana sa istim nizom stanja S najmanje dva puta.

U suprotnom, kada se `program_pulibot` završi, sample grader izvršava Pulibotov program u lavirintu opisanom samim inputom.

The sample grader daje dva izlaza.

Sample grader upisuje spisak Pulibotovih akcija u datoteku `robot.bin` u radnom direktorijumu. Ova datoteka služi kao ulaz za alat za vizualizaciju opisan u sljedećem odjeljku.

Drugo, ako se Pulibotov program ne završi uspješno, Sample grader ispisuje jednu od sljedećih poruka o grešci:

- `Unexpected state`: Pulibot je našao nis stanja za koje `set_instruction` nije pozvana.

- `Invalid move`: izvršavanje akcije dovodi Pulibot na polje koje nije slobodno
- `Too many steps`: Pulibot je izvršio 500 000 koraka i nije završio izvršavanje.

Inače, neka $e[r][c]$ bude stanje ćelije (r, c) nakon što se Pulibotov program izvrši. Sample grader štampa H linija u sledećem formatu:

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

Alat za prikaz

Paket priloga za ovaj zadatak sadrži datoteku pod nazivom `display.py`. Kada se pozove, ova Python skripta prikazuje Pulibotove akcije u lavirintu opisanom unosom sample gradera. Za ovo, binarna datoteka `robot.bin` mora biti prisutna u radnom direktorijumu.

Da biste pozvali skriptu, izvršite sljedeću naredbu.

```
python3 display.py
```

Pojavljuje se jednostavan grafički interfejs. Glavne mogućnosti su sljedeće:

- Možete pratiti status čitavog lavirinta. Trenutna lokacija Pulibota je istaknuta pravougaonikom.
- Možete pretraživati korake Pulibota tako što ćete kliknuti na dugmad sa strelicama ili pritisnuti strelice na tastaturi. Takođe možete skočiti na određeni korak.
- Predstojeći korak u Pulibotovom programu prikazan je na dnu. Prikazuje se niz trenutnih stanja i instrukcija koje će se izvršiti. Nakon završne instrukcije, prikazuje ili jednu od poruka o grešci grejdera ili `Terminated` ako se program uspješno završi.
- Svakom broju koji predstavlja boju, možete dodijeliti vizuelnu boju pozadine, kao i tekst na ekranu. Prikazani tekst je kratak niz koji se upisuje u svaku ćeliju iste boje. Možete dodijeliti boje pozadine i prikazati tekstove na jedan od sljedećih načina:
 - Postavite ih u dijalog prozor nakon što kliknete na dugme `Colors`.
 - Uredite sadržaj datoteke `colors.txt`.
- Za ponovno učitavanje `robot.bin` koristite dugme `Reload`. Ovo je korisno ako se promijenio sadržaj datoteke `'robot.bin'`.