



Роботско такмичење

Истраживачи вештачке интелигенције на Универзитету у Сегедину одржавају такмичење у програмирању робота. Твоја пријатељица, Марија, одлучила је да учествује у такмичењу. Циљ је програмирати ултимативног *Пулибота*, дивећи се великој интелигенцији познате пасмине мађарских пастирских паса, Пули.

Пулибот ће бити тестиран на лавиринту који се састоји од мреже ћелија димензија $(H + 2) \times (W + 2)$. Редови мреже су нумерисани бројевима од -1 до H од севера према југу, а колоне мреже су нумерисане бројевима од -1 до W од запада према истоку. Ћелију која се налази у реду r и колони c мреже ($-1 \leq r \leq H$, $-1 \leq c \leq W$) обележаваћемо са (r, c) .

Размотримо ћелију (r, c) такву да је $0 \leq r < H$ и $0 \leq c < W$. Постоје 4 ћелије **суседне** ћелији (r, c) :

- ћелија $(r, c - 1)$ **западно** од ћелије (r, c) ;
- ћелија $(r + 1, c)$ **јужно** од ћелије (r, c) ;
- ћелија $(r, c + 1)$ **источно** од ћелије (r, c) ;
- ћелија $(r - 1, c)$ **северно** од ћелије (r, c) .

Ћелија (r, c) се назива **гранична** ћелија лавиринта ако важи $r = -1$ или $r = H$ или $c = -1$ или $c = W$. Свака ћелија која није гранична ћелија лавиринта је или **ћелија с препрекама** или **празна** ћелија. Додатно, свака празна ћелија има **боју**, представљену ненегативним целим бројем између 0 и Z_{MAX} , укључујући. У почетку, боја сваке празне ћелије је 0.

На пример, посматрајмо лавиринт димензија $H = 4$ и $W = 5$, који садржи једну ћелију са препреком у ћелији $(1, 3)$:

	-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							

Једина ћелија с препреком је означена иксићем. Граничне ћелије лавиринта су осенчене. Број уписан у сваку празну ћелију представља њену боју.

Пут дужине ℓ ($\ell > 0$) од ћелије (r_0, c_0) до ћелије (r_ℓ, c_ℓ) је низ у паровима различитих *уразних* ћелија $(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$ таквих да за свако i ($0 \leq i < \ell$) ћелије (r_i, c_i) и (r_{i+1}, c_{i+1}) су суседне.

Имајте на уму да пут дужине ℓ садржи тачно $\ell + 1$ ћелија.

На такмичењу су истраживачи поставили лавиринт у којем постоји барем један пут од ћелије $(0, 0)$ до ћелије $(H - 1, W - 1)$. Примјетите да ово имплицира да су ћелије $(0, 0)$ и $(H - 1, W - 1)$ загарантовано празне.

Марија не зна које су ћелије лавиринта празне, а које су са препрекама.

Ваш задатак је помоћи Марији да програмира Пулибота тако да буде способан пронаћи *најкраћи ууш* (тј. пут минималне дужине) од ћелије $(0, 0)$ до ћелије $(H - 1, W - 1)$ у непознатом лавиринту који су поставили истраживачи. Спецификације Пулибота и правила такмичења су описани у наставку.

Обратите пажњу да последња секција поставке задатка описује алат за визуализацију који можете да користите да визуализијете Пулибота.

Пулиботова спецификација

Дефинишимо **стање** ћелије (r, c) ($-1 \leq r \leq H, -1 \leq c \leq W$) као цели број тако да:

- ако је ћелија (r, c) гранична ћелија тада је њено стање -2 ;
- ако је ћелија (r, c) ћелија са препреком онда је њено стање -1 ;
- ако је ћелија (r, c) празна ћелија онда је њено стање представља боју ћелије.

Пулиботов програм се извршава кроз низ корака. У сваком кораку, Пулибот препознаје стања оближњих ћелија па затим извршава инструкцију. Инструкција коју извршава је одређена стањима оближњих ћелија. Следи прецизнији опис.

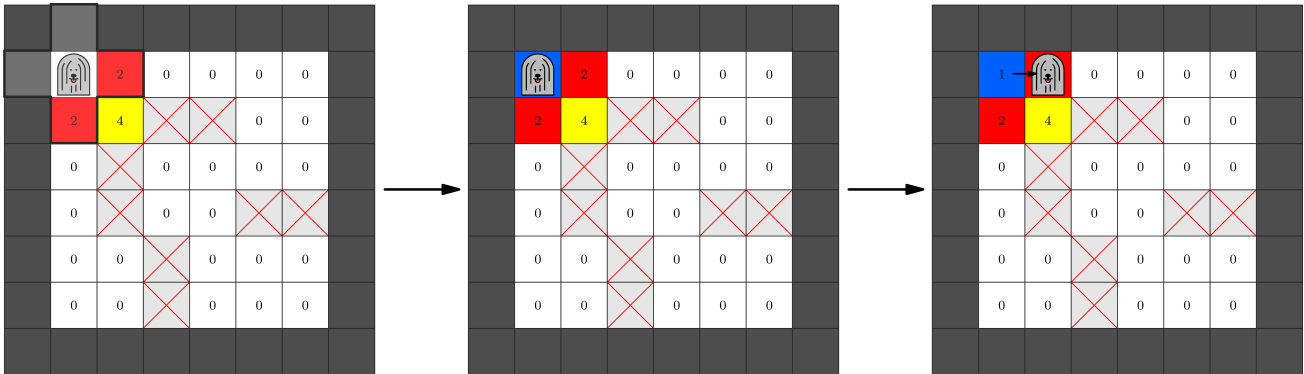
Претпоставимо да је на почетку тренутног корака Пулибот у ћелији (r, c) , која је празна ћелија. Корак се изводи на следећи начин:

1. Прво, Пулибот провери тренутни **низ стања**, то јест, низ $S = [S[0], S[1], S[2], S[3], S[4]]$, који се састоји од стања ћелије (r, c) и свих суседних ћелија:
 - $S[0]$ је стање ћелије (r, c) .
 - $S[1]$ је стање ћелије на западу.
 - $S[2]$ је стање ћелије на југу.
 - $S[3]$ је стање ћелије на истоку.
 - $S[4]$ је стање ћелије на сјеверу.
2. Затим, Пулибот одређује **инструкцију** (Z, A) која одговара препознатом низу стања.

3. Коначно, Пулибот изводи одговарајучу инструкцију: поставља боју ћелије (r, c) на боју Z и изводи акцију A , која је једна од следећих акција:

- *остати* у ћелији (r, c) ;
- *преместити се* у једну од 4 сусједне ћелије;
- *заврши програм*.

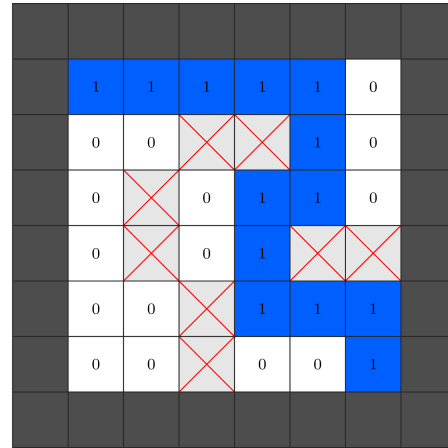
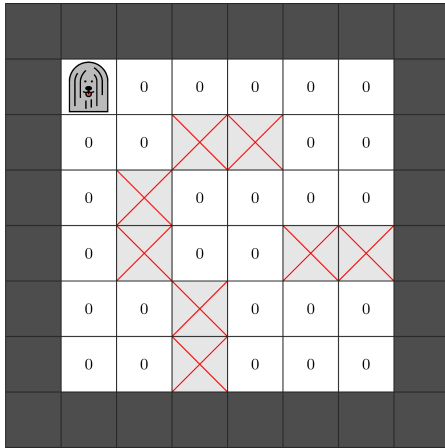
На пример, размотримо случај приказан на левој страни следеће слике. Пулибот је тренутно у ћелији $(0, 0)$ са бојом 0. Пулибот препознаје низ стања $S = [0, -2, 2, 2, -2]$. Пулибот може имати програм који, након препознавања овог низа, поставља боју тренутне ћелије на $Z = 1$, а затим се помиче на исток, као што је приказано у средини и десно од слике:



Правила такмичења робота

- На почетку, Пулибот се поставља у ћелију $(0, 0)$ и почиње да извршава свој програм.
- Пулиботу није дозвољено да се помери у ћелију која није празна.
- Пулиботов програм се мора завршити након највише 500 000 корака.
- Након завршетка Пулиботовог програма, празне ћелије у лавиринту треба да буду обојене тако да:
 - Постоји најкраћи пут од $(0, 0)$ до $(H - 1, W - 1)$ за који је боја сваке ћелије укључене у путању 1.
 - Све остале празне ћелије имају боју 0.
- Пулибот може прекинути свој програм у било којој празној ћелији.

На примјер, следећа слика приказује могући лавиринт са димензијама $H = W = 6$. Почетна конфигурација је приказана на левој страни, а једно прихватљиво бојење празних ћелија након завршетка је приказано на десној страни:



Детаљи имплементације

Потребно је да имплементирате следећу функцију.

```
void program_pulibot()
```

- Ова функција би требала генерисати Пулиботов програм. Овај програм би требао радити исправно за све вредности H и W и сваки лавиринт који испуњава ограничења задатка.
- Ова функција се позива тачно једном за сваки тест случај.

Ова функција може позвати следећу функцију за генерисање Пулиботовог програма:

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

- S : низ дужине 5 који описује низ стања.
- Z : ненегативан цели број који представља боју.
- A : један знак који представља акцију Пулибота како следи:
 - Н: остани у тренутној ћелији;
 - W: помери се на запад;
 - S: помери се на југ;
 - E: помери се на исток;
 - N: помери се на сјевер;
 - T: заврши програм.
- Позивање ове функције упућује Пулибот-а да након препознавања низа стања S треба извршити инструкцију (Z, A) .

Позивање ове функције више пута са истим низом стања S резултираће излазом 'Output isn't correct'.

Није потребно позвати `set_instruction` са сваким могућим низом стања S . Међутим, ако Пулибот касније препозна низ стања за које инструкција није постављена, добићете излаз

'Output isn't correct'.

Након што се `program_pulibot` заврши, грејдер позива Пулиботов програм над једним или више лавиринта. Ови позиви се *не* убрајају у временско ограничење за ваше решење. Грејдер *није* адаптиван, то јест, скуп лавиринта је унапред дефинисан у сваком тестном случају.

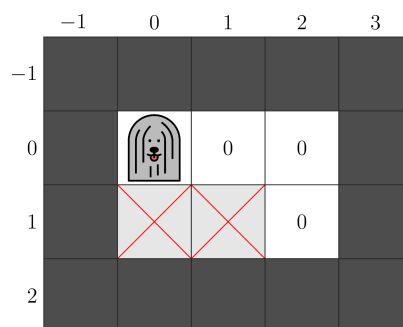
Ако Пулибот прекрши било које од Правила такмичења робота пре него што прекине свој програм, добићете излаз 'Output isn't correct'.

Пример

Функција `program_pulibot` може позивати `set_instruction` на следећи начин:

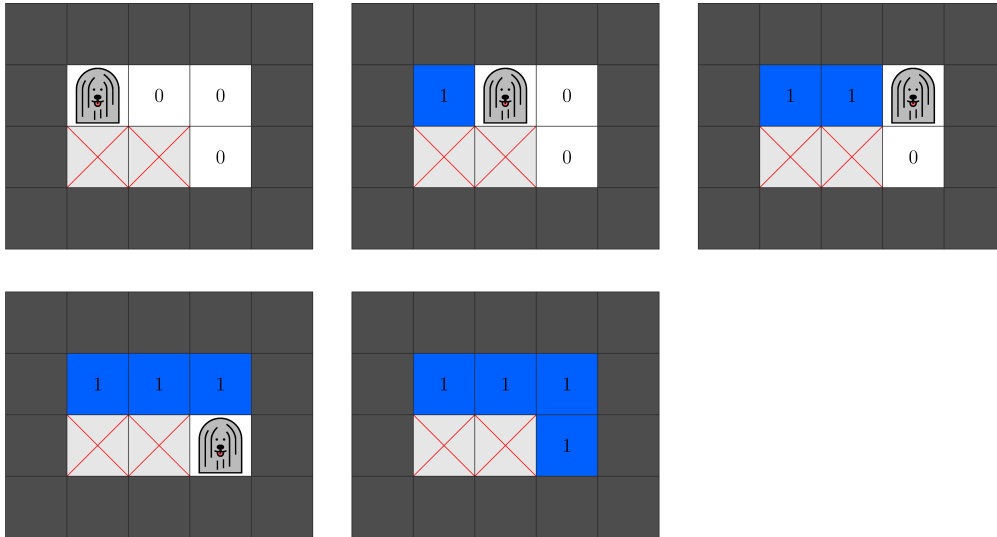
Позиви	Инструкција за низ стања S
<code>set_instruction([0, -2, -1, 0, -2], 1, E)</code>	Постави боју на 1 и помакни се источно
<code>set_instruction([0, 1, -1, 0, -2], 1, E)</code>	Постави боју на 1 и помакни се источно
<code>set_instruction([0, 1, 0, -2, -2], 1, S)</code>	Постави боју на 1 и помакни се јужно
<code>set_instruction([0, -1, -2, -2, 1], 1, T)</code>	Постави боју на 1 и заврши програм

Размотримо случај где је $H = 2$ и $W = 3$, а лавиринт је приказан на следећој слици:



За овај конкретан лавиринт Пулиботов програм се извршава у четири корака. Низови стања која Пулибот препознаје одговарају позивима функције `set_instruction` приказаних у табели. Последња од ових инструкција прекида програм.

Следећа слика приказује стање лавиринта пре сваког од четири корака и његово коначно стање након завршетка.



Међутим, имајте на уму да овај програм од 4 инструкције можда неће пронаћи најкраћи пут у другим валидним лавиринтима. Стога, ако се субмитује, добиће поруку 'Output isn't correct'.

Ограничења

$Z_{MAX} = 19$. Дакле, Пулибот може користити боје од 0 до 19, укључујући.

За сваки лавиринт кориштен за тестирање Пулибота:

- $2 \leq H, W \leq 15$
- Постоји најмање једна путања од ћелије $(0, 0)$ до ћелије $(H - 1, W - 1)$.

Подзадаци

1. (6 бодова) У лавиринту нема ћелије с препрекама.
2. (10 бодова) $H = 2$
3. (18 бодова) Постоји тачно један пут између сваког пара празних ћелија.
4. (20 бодова) Сваки најкраћи пут од ћелије $(0, 0)$ до ћелије $(H - 1, W - 1)$ има дужину $H + W - 2$.
5. (46 бодова) Нема додатних ограничења.

Ако, у било којем од тест случајева, позиви функцији `set_instruction` или Пулиботовом програму током њеног извршавања нису у складу с ограничењима описаним у Детаљима имплементације, ваше решење ће бити оцењено са 0 бодова на том подзадатку.

У сваком подзадатку можете добити део бодова тако што ћете наћи скоро тачно бојење.

Формално:

- Решење тест случаја је **потпуно** ако коначно бојење празних ћелија задовољава Правила такмичења робота.
- Решење тестног случаја је **делимично** ако коначно бојење изгледа овако:

- Постоји најкраћи пут од $(0,0)$ до $(H-1, W-1)$ за који је боја сваке ћелије укључене у путању 1.
- Нема друге празне ћелије у мрежи са бојом 1.
- Нека празна ћелија у мрежи има боју која није 0 и 1.

Ако ваше решење тест примјера није ни потпуно ни делимично, ваше решење ће бити оцењено са 0 бодова на том тест примеру.

У подзадацима 1-4, потпуно решење ће бити оцењено са 100%, а делимично решење ће бити оцењено са 50% бодова предвиђених за тај подзатак.

У подзатку 5, ваш резултат зависи од броја боја кориштених у Пулиботовом програму. Прецизније, означите са Z^* максималну вредност Z за све позиве функције `set_instruction`. Резултат тест примера се израчунава:

Стање	Резултат (потпуно)	Резултат (делимичан)
$11 \leq Z^* \leq 19$	$20 + (19 - Z^*)$	$12 + (19 - Z^*)$
$Z^* = 10$	31	23
$Z^* = 9$	34	26
$Z^* = 8$	38	29
$Z^* = 7$	42	32
$Z^* \leq 6$	46	36

Резултат за сваки подзатак је минимум бодова за тест примере у том подзатку.

Пример оцењивача (sample grader)

Оцењивач чита улаз у следећем формату:

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

m представља низ од H низова дужине W који описују неграничне ћелије лавиринта. $m[r][c] = 0$ ако је ћелија (r, c) празна и $m[r][c] = 1$ ако је ћелија (r, c) ћелија препрека.

Оцењивач прво позива `program_pulibot()`. Ако оцењивач открије кршење протокола, исписује се порука `Protocol Violation: <MSG>` и завршава, где је `<MSG>` једна од следећих порука о грешци:

- `Invalid array`: $-2 \leq S[i] \leq Z_{MAX}$ није задовољено i или дужина за S није 5.
- `Invalid color`: $0 \leq Z \leq Z_{MAX}$ није задовољено.
- `Invalid action`: карактер A није ни један од карактера H, W, S, E, N, T.

- Same state array: `set_instruction` је позвана са истим низом стања S најмање два пута.

У супротном, када се `program_pulibot` заврши, оцењивач извршава Пулиботов програм у лавиринту описаном самим улазом.

Оцењивач даје два излаза.

Оцењивач уписује списак Пулиботових акција у датотеку `robot.bin` у радном директоријуму. Ова датотека служи као улаз за алат за визуализацију описан у следећем одељку.

Друго, ако се Пулиботов програм не заврши успешно, оцењивач исписује једну од следећих порука о грешци:

- Unexpected state: Пулибот је нашао низ стања за који `set_instruction` није позвана.
- Invalid move: извршавање акције доводи Пулибот на поље које није слободно.
- Too many steps: Пулибот је извршио 500 000 корака и није завршио извршавање.

Иначе, нека $e[r][c]$ буде стање ћелије (r, c) након што се Пулиботов програм изврши. Оцењивач штампа H линија у следећем формату:

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

Алат за приказ

Пакет прилога за овај задатак садржи датотеку под називом `display.py`. Када се позове, ова Python скрипта приказује Пулиботове акције у лавиринту описаном уносом оцењивача. За ово, бинарна датотека `robot.bin` мора бити присутна у радном директоријуму.

Да бисте позвали скрипту, извршите следећу наредбу.

```
python3 display.py
```

Појављује се једноставан графички интерфејс. Главне могућности су следеће:

- Можете пратити статус читавог лавиринта. Тренутна локација Пулибота је истакнута правоугаоником.
- Можете претраживати кораке Пулибота тако што ћете кликнути на дугмад са стрелицама или притиснути стрелице на тастатури. Такође можете скочити на одређени корак.
- Предстојећи корак у Пулиботовом програму приказан је на дну. Приказује се низ тренутних стања и инструкција које ће се извршити. Након завршне инструкције, приказује или једну од порука о грешци оцењивача или `Terminated` ако се програм успешно заврши.

- Сваком броју који представља боју, можете доделити визуелну боју позадине, као и текст на екрану. Приказани текст је кратак низ који се уписује у сваку ћелију исте боје. Можете доделити боје позадине и приказати текстове на један од следећих начина:
 - Поставите их у дијалог прозор након што кликнете на дугме `Colors`.
 - Уредите садржај датотеке `colors.txt`.
- За поновно учитавање `robot.bin` користите дугме `Reload`. Ово је корисно ако се променио садржај датотеке `'robot.bin'`.