Robot Yarışı

Szeged Universitetinin Süni İntellekt tədqiqatçıları robot proqramlaşdırma yarışı keçirirlər. Sənin dostun Həmid də həmin yarışda iştirak etməyə qərar verdi. Əsas hədəf məşhur Macarıstan çoban iti olan Puli'dən ilhamlanaraq yaradılmış mükəmməl *Pulibot* robotu üçün proqram yazmaqdır.

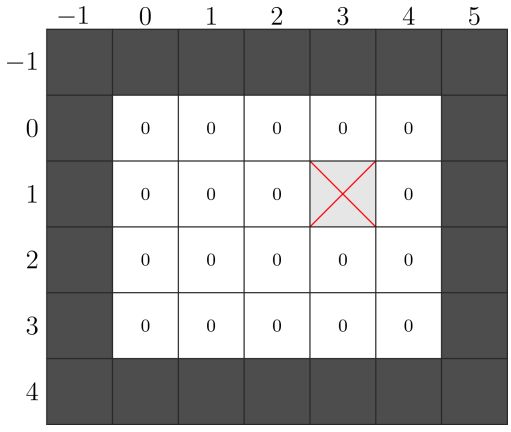
Pulibot $(H+2) \times (W+2)$ ölçülü cədvəl formasında olan bir labirintdə yoxlanılacaq. Cədvəlin sətirləri şimaldan cənuba doğru -1'dən H'ə qədər, sütunları isə qərbdən şərqə doğru -1'dən W'ya qədər nömrələnib. r'ci sətir və c'ci sütunda ($-1 \leq r \leq H$, $-1 \leq c \leq W$) olan xanaya (r, c) xanası deyəcəyik.

$0 \leq r < H$ və $0 \leq c < W$ şərtini ödəyən bir (r, c) xanasına nəzər yetirin. (r, c) xanasının 4 **qonşu** xanası var:

- $(r, c-1)$ xanasını (r, c) xanasının **qərbi** adlandıracağıq;
- $(r+1, c)$ xanasını (r, c) xanasının **cənubu** adlandıracağıq;
- $(r, c+1)$ xanasını (r, c) xanasının **şərqi** adlandıracağıq;
- $(r-1, c)$ xanasını (r, c) xanasının **şimalı** adlandıracağıq.

(r, c) xanası üçün əgər $r = -1$ və ya $r = H$ və ya $c = -1$ və ya $c = W$ şərtlərindən hər hansı biri ödənərsə, o zaman bu xananı **sərhəd** xana adlandıracağıq. Sərhəd olmayan hər bir xana ya **maneə**, ya da **boş** xanadır. Əlavə olaraq, hər bir boş xananın 0 və Z_{MAX} (0 və Z_{MAX} daxil) arasında mənfi olmayan tam ədəd **rəngi** var. Başda bütün boş xanaların rəngi 0'dır.

Məsələn, $H = 4$ və $W = 5$ olan, yalnızca $(1, 3)$ xanasında maneə olan labirintə nəzər yetirək:



Yeganə maneə olan xana üzərindən qırmızı X çəkilib. Labirintin sərhədləri tünd rəngdədir. Boş xanaların içindəki ədədlər onların rəngini göstərir.

Bir birindən müxtəlif $(r_0, c_0), (r_1, c_1), \ldots, (r_\ell, c_\ell)$ boş xanaları üçün əgər (r_i, c_i) və (r_{i+1}, c_{i+1}) ($0 \leq i < \ell$) xanaları qonşudursa, o zaman bu ardıcıllığı (r_0, c_0)'dan (r_ℓ, c_ℓ)'ə olan ℓ uzunluqlu **yol** adlandıraq. Diqqət edin ki, ℓ uzunluqlu yolun tam olaraq $\ell + 1$ sayda xanası var.

Yarış zamanı tədqiqatçılar labirinti ele qurur ki, $(0, 0)$ və $(H - 1, W - 1)$ arasında ən az bir yol olsun. Qeyd edək ki, bu o mənaya gəlir ki $(0, 0)$ və $(H - 1, W - 1)$ xanaları boşdur.

Həmid hansı xanalar boş, hansılar maneədir bilmir.

Sizin vəzifəniz,tədqiqatçıların düzəltdiyi və sizin tərəfinizdən bilinməyən labirintdə Pulibot'un $(0, 0)$ xanasından $(H - 1, W - 1)$ xanasına ən qısa yolu tapması üçün proqram yazmağında Həmidə kömək etməkdir. Pulibot'un özəllikləri və yarışın qaydaları aşağıda izah olunub.

Qeyd edək ki bu məsələnin sonunda Pulibot'u vizualizasiya edə biləcəyiniz aləti izah edilir.

Pulibot'un Özəllikləri

$-1 \leq r \leq H$ və $-1 \leq c \leq W$ şərtini ödəyən bütün (r, c) xanaları üçün həmin xananın **vəziyyətini** belə bir ədədlə göstərək:

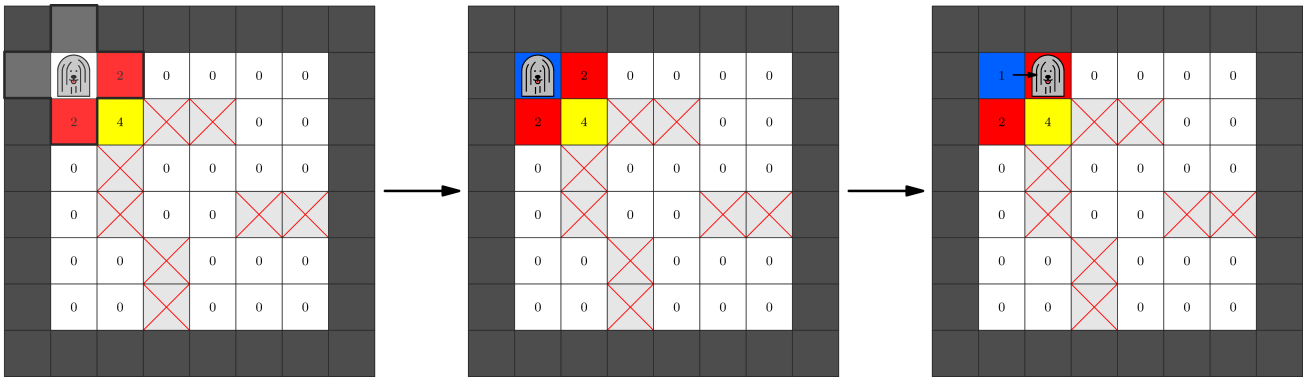
- əgər (r, c) xanası sərhəddirsə, o zaman onun vəziyyəti -2'dir;
- əgər (r, c) xanası maneədirsə, o zaman onun vəziyyəti -1'dir;
- əgər (r, c) xanası boşdursa, o zaman onun vəziyyəti həmin xananın rəngidir.

Pulibot'un proqramı addımlar ardıcıllığı ilə yerinə yetirilir. Hər bir addımda, Pulibot ətrafdakı xanaların vəziyyətlərinə nəzər yetirir və hansısa əməliyyatı yerinə yetirir. Yerinə yetirdiyi əməliyyat nəzər yetirdiyi xanaların vəziyyətlərindən asılıdır. Aşağıda daha dəqiq izah verilib.

Fərz edək ki, hazırki gedişin əvvəlində Pulibot (r, c) xanasındadır, hansı ki boş xanadır. Addım aşağıdakı kimi yerinə yetirilir:

1. Əvvəlcə, Pulibot hazırki **vəziyyət massivinə**, yəni hazırda üstündə durduğu (r, c) xanasının və qonşu xanaların vəziyyətlərini göstərən $S = [S[0], S[1], S[2], S[3], S[4]]$ massivinə nəzər yetirir:
    - $S[0]$ hazırki, yəni (r, c) xanasının vəzziyətini göstərir.
    - $S[1]$ qərbdəki xananın vəziyyətini göstərir.
    - $S[2]$ cənubdakı xananın vəziyyətini göstərir
    - $S[3]$ şərqdəki xananın vəziyyətini göstərir
    - $S[4]$ şimaldakı xananın vəziyyətini göstərir
2. Daha sonra Pulibot vəziyyət massivinə uyğun gələn (Z, A) **əməliyyatını** təyin edir.
3. Nəhayətdə, Pulibot əməliyyatı yerinə yetirir: (r, c) xanasını Z ilə rəngləyir, daha sonra isə A əməliyatını yerinə yetirir, hansı ki bu əməliyyat aşağıdakılardan biridir:
    - (r, c) xanasında *qalmaq*
    - qonşu 4 xanadan birinə *getmək*
    - *proqramı dayandırmaq*.

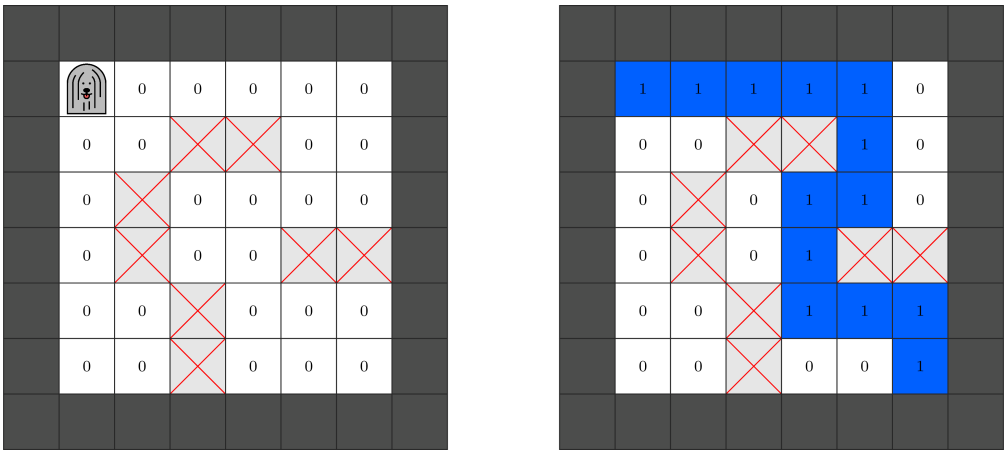
Məsələn, aşağıda verilən şəkillərdən soldakına nəzər yetirin. Pulibot hazırda $(0, 0)$ xanasındadır və həmin xananın rəngi 0'dır. Pulibot $S = [0, -2, 2, 2, -2]$ vəziyyət massivini görür. Pulibot'un ortada və sağdakı şəkillərdə göstərilən formada belə bir proqramı ola bilər ki, bu massivi gördükdən sonra hazırki xananı $Z = 1$'ə rəngləsin və şərqə getsin.



Robot yarışının qaydaları

- Başda, Pulibot $(0, 0)$ xanasındadır və proqramını işə salır.
- Pulibot boş olmayan xanaya gedə bilməz.
- Pulibot'un proqramı ən çox $500\,000$ əməliyyatdan sonra dayanmalıdır.
- Pullibot'un proqramı dayandıqdan sonra boş xanalar elə rənglənməlidir ki:
  - $(0, 0)$ xanasından $(H - 1, W - 1)$ xanasına gedən ən qısa yollardan biri 1 ilə rənglənib.
  - Digər bütün boş xanaların rəngi 0'dır.
  - Proqramı dayandıqdan sonra Pulibot hər hansı xanada dayana bilər.

Məsələn, aşağıdakı şəkil $H = W = 6$ üçün mümkün labirintlərdən birini göstərir. Başlanğıc versiyası solda, düzgün rəngləmələrdən biri sağda göstərilib:



İmplementasiya detalları

Aşağıdakı proseduru implement etməlisiniz

```
void program_pulibot()
```

- Bu prosedur Pulibot'un proqramını yaratmalıdır. Alınan proqram H və W'nin hər bir dəyəri və tapşırığın qaydalarına uyğun mümkün bütün labirintlər üçün düzgün işləməlidir.
- Bu prosedur hər test üçün yalnızca bir dəfə çağırılacaq.

Bu prosedur Pulibot'un proqramını yaratmaq üçün aşağıdakı proseduru çağıra bilər:

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

- S: vəziyyət massivini göstərən 5 uzunluqlu massiv
- Z: rəngi göstərən mənfi olmayan tam ədəd
- A: Pullibot'un yerinə yetirəcəyi əməliyyatı göstərən tək bir simvol:
  - `H`: olduğu yerdə dayan;
  - `W`: qərbə get;
  - `S`: cənuba get;
  - `E`: şərqə get;
  - `N`: şimala get;
  - `T`: proqramı dayandır.
- Bu proseduru çağırmaq Pulibot'a S vəziyyət massivini gördükdə (Z, A) əməliyyatını yerinə yetirməli olduğu instruksiyasını verir.

Bu proseduru eyni S massivi ilə bir neçə dəfə çağırsanız `Output isn't correct` cavabı alacaqsınız.

S massivinin bütün mümkün halları üçün `set_instruction` prosedurunu çağırmaq məcburiyyətində deyilsiniz. Lakin əgər Pulibot elə bir vəziyyətə gəlsə ki həmin vəziyyət üçün yerinə yetirilməli olan instruksiya təyin olunmayıb, o zaman `Output isn't correct` cavabı alacaqsınız.

`program_pulibot` bitdikdən sonra qreyder Pulibot'un proqramını bir və ya bir neçə labirint üzərində test edir.

Bu testlər sizin zaman limitinizə *hesablanmır*. Qreyder adaptiv *deyil*, yəni hər test üçün yoxlanılacaq labirintlər bəri başdan təyin olunub.

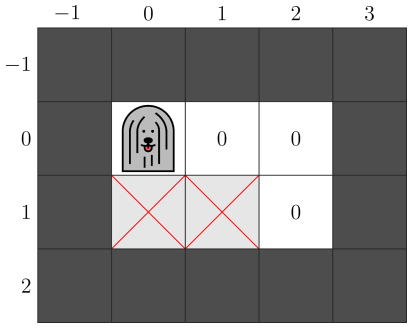
Əgər Pulibot Robot Yarış Qaydalarının hər hansı birinə əməl etməsə, o zaman `Output isn't correct` cavabı alacaqsınız.

Nümunə

`program_pulibot` proseduru `set_instruction` proseduruna aşağıdakı kimi çağırışlar edə bilər:

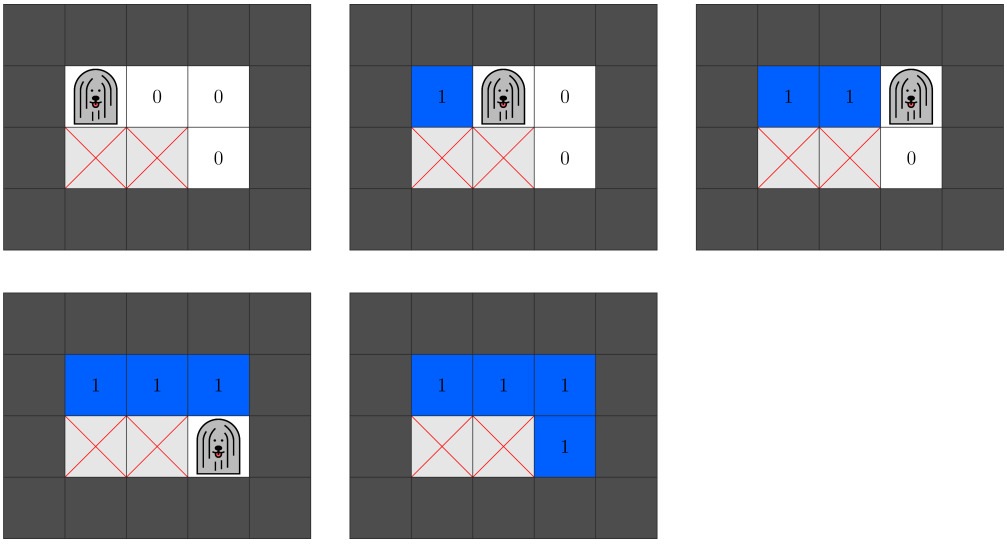
Çağırış	S massivi üçün verilən instruksiya
`set_instruction([0, -2, -1, 0, -2], 1, E)`	Xananı 1 ilə rənglə və şərqə get
`set_instruction([0, 1, -1, 0, -2], 1, E)`	Xananı 1 ilə rənglə və şərqə get
`set_instruction([0, 1, 0, -2, -2], 1, S)`	Xananı 1 ilə rənglə və cənuba get
`set_instruction([0, -1, -2, -2, 1], 1, T)`	Xananı 1 ilə rənglə və proqramı dayandır

$H = 2$ və $W = 3$ olan və labirintin aşağıdakı kimi olduğu vəziyyətə nəzər yetirin.



Bu labirint üçün Pulibot'un proqramı dörd addım edir. Pulibot'un təyin etdiyi vəziyyət massivləri və yerinə yetirdiyi əməliyyatlar yuxarıda göstərilən instruksiyalar ilə sırayla üst-üstə düşür. Bunlardan sonuncusu proqramı dayandırır.

Aşağıdakı şəkil labirintin hər bir addımdan əvvəlki və proqram dayandıqdan sonrakı vəziyyətini göstərir.



Lakin, qeyd edək ki bu 4 addımdan ibarət proqram digər labirintlər üçün düzgün yolu tapmaya bilər. Buna görə də bu həll submit olunduqda `Output isn't correct` cavabı alacaq.

Məhdudiyyətlər

$Z_{MAX} = 19$. Yəni Pulibot 0'dan 19'a qədər (0 və 19 daxil) ədədlər ilə xanaları rəngləyə bilər.

Pulibot'u yoxlayan hər bir labirint üçün:

- $2 \leq H, W \leq 15$
- $(0, 0)$ xanasından $(H-1, W-1)$ xanasına ən az bir yol var.

Alt tapşırıqlar

1. (6 bal) Labirintdə maneə yoxdur.
2. (10 bal) $H = 2$
3. (18 bal) İstənilən iki boş xana arasında tam olaraq bir yol var.
4. (20 bal) $(0, 0)$ xanasından $(H-1, W-1)$ xanasına olan ən qısa yolun uzunluğu $H + W - 2$'dir.
5. (46 bal) Əlavə məhdudiyyət yoxdur.

Əgər alt tapşırığın hansısa testində `set_instruction` proseduruna olan çağırışlar zamanı və ya Pulibot'un proqramı işləyən zaman İmplementasiya Detallarından kənara çıxmalar yaşanarsa, o zaman həmin alt tapşırıq üçün sizin balınız 0 olacaq.

Hər alt tapşırıq üçün düzgün rəngləməyə yaxın bir həll tapsanız, həmin tapşırığın balının bir hisəsini ala bilərsiniz. Belə ki:

- Boş xanaların yekun rənglənməsi Robot Yarış Qaydalarına tam uyğundursa, o zaman həmin test üçün həll **tamdır**
- Əgər aşağıdakılar doğrudursa o zaman alt tapşırığın həlli **yarımçıqdır**:
  - $(0, 0)$'dan $(H-1, W-1)$ xanasına olan ən qısa yollardan birinin hər bir xanası 1 ilə rənglənib.
  - Başqa heç bir xana 1 ilə rənglənməyib.
  - Bəzi boş xanaların 0 və 1'dən başqa rəngi var.

Əgər sizin hansısa test üçün həlliniz nə tam nə yarımçıqdırsa, o zaman həmin test üçün balınız 0 olacaq.

1-4'cü alt tapşırıqlarda əgər həlliniz yarımçıqdırsa 50%, tamdırsa 100% bal alacaqsınız.

5'ci alt tapşırıqda sizin balınız Pulibot'un istifadə etdiyi rənglərin sayından asılıdır. Daha dəqiq desək, $Z^\star$ `set_instruction` proseduruna olan çağırışlardakı Z'lərin ən böyüyünü göstərsin. Testin balı aşağıdakı kimi təyin olunur:

Şərt	Bal (tam)	Bal (yarımçıq)
$11 \le Z^\star \le 19$	$20 + (19 - Z^\star)$	$12 + (19 - Z^\star)$
$Z^\star = 10$	31	23
$Z^\star = 9$	34	26
$Z^\star = 8$	38	29
$Z^\star = 7$	42	32
$Z^\star \le 6$	46	36

Alt tapşırıqların balı həmin tapşırıqdakı ən az bal verən testin balına bərabərdir.

Nümunə Qreyder

Nümunə qreyder girişi aşağıdakı formada oxuyur:

- sətir 1: H W
- sətir $2 + r$ ($0 \le r < H$): $m[r][0]$ $m[r][1]$ $\ldots$ $m[r][W-1]$

Burada m hər bir elementi W ədəddən ibarət massiv olan H uzunluqlu massivdir və labirintin sərhəd olmayan xanalarını təmsil edir. Əgər (r, c) xanası boş olarsa $m[r][c] = 0$, əks halda $m[r][c] = 1$.

Nümunə qreyder əvvəlcə `program_pulibot()` prosedurunu çağırır. Əgər nümunə qreyder qayda pozuntusu aşkarlayarsa, o zaman `Protocol Violation: <MSG>` çıxışa verir və dayanır. Burada `<MSG>` aşağıdakı error mesajlarından biridir:

- `Invalid array`: hansısa i üçün $-2 \le S[i] \le Z_{MAX}$ şərti ödənməyib və ya S massivinin uzunluğu 5 deyil.
- `Invalid color`: $0 \le Z \le Z_{MAX}$ şərti ödənməyib.
- `Invalid action`: A simvolu `H`, `W`, `S`, `E`, `N`, `T` hərflərindən heç biri deyil.
- `Same state array`: `set_instruction` proseduru eyni S massivi ilə ən az 2 dəfə çağırılıb.

Əks halda `program_pulibot` bitdikdə nümunə qreyder Pulibot'un proqramını girişdə verilən labirintdə işə salır.

Nümunə qreyder çıxışa iki məlumat verir.

Birinci, nümunə qreyder Pulibot'un hərəkətlərini işlədiyiniz qovluqda `robot.bin` faylına yazır. Bu fayl aşağıda izah olunan vizualizasiya alətinə giriş olaraq işlədiləcək.

İkinci, Pulibot'un proqramı düzgün dayanmasa qreyder aşağıdakı mesajlardan birini çıxışa verir:

- `Unexpected state`: Pulibot elə bir vəziyyət massivi gördü ki `set_instruction` proseduru həmin massiv ilə çağırılmayıb.
- `Invalid move`: Əməliyyatı yerinə yetirdikdən sonra Pulibot boş olmayan xanaya getdi.
- `Too many steps`: Pulibot'un proqramı dayanmadan $500\,000$ əməliyyat yerinə yetirdi.

Əks halda $e[r][c]$ Pulibot'un proqramı dayandıqdan sonra (r, c) xanasının vəziyyəti olsun. Nümunə qreyder aşağıdakı formada H sətir çıxışa verir:

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

Vizualizasiya aləti

Bu sualın əlavə paketində `display.py` adlı fayl var. İşlətdiyiniz zaman bu script nümunə qreyderin verdiyi girişə əsasən Pulibot'un əməliyyatlarını göstərir. Bunun üçün `robot.bin` binary faylı işlədiyiniz qovluqda olmalıdır.

Script'i çalışdırmaq üçün aşağıdakı əmri yerinə yetirin

```
python3 display.py
```

Sadə qrafik interfeys ekrana gəlir. Əsas özəllikləri bunlardır:

- Labirintin bütün halına nəzər yetirə bilərsiniz. Pulibut'un hazırda durduğu xana işarələnib.
- Pulibot'un əməliyyatlarını ox düymələri və ya hotkey'lər ilə gəzə bilərsiniz. Həmçinin hansısa spesifik əməliyyata bir başa gedə bilərsiniz.
- Pulibot'un proqramındakı növbəti addım aşağıda göstərilir. Orada hazırki vəziyyət massivi və yerinə yetirəcəyi əməliyyat göstərilir. Sonuncu əməliyyatdan sonra ya qreyderin error mesajlarından birini göstərir, ya da `Terminated` yazır (əgər proqram uğurlu şəkildə dayanıbsa).
- Rəngi təmsil edən hər bir ədədə arxa plan rəngi və yazı təyin edə bilərsiniz. Bu yazı həmin rəngdə olan xanaların üstünə yazılacaq qısa bir string'dir. Rəng və yazı təyin etmək üçün aşağıdakı yollardan istifadə edə bilərsiniz:
  - `Colors` düyməsini sıxdıqdan sonra çıxan dialoq pəncərəsində təyin edin.
  - `colors.txt` faylında dəyişiklik edin.
- `robot.bin` faylını yeniləmək üçün `Reload` düyməsindən istifadə edin. Əgər həmin faylın içi dəyişibsə bu yararlıdır.