



Robot Contest

Сегед университетінің AI зерттеушілері роботтарды бағдарламалау байқауын өткізуде.

Досыңыз Ханга байқауға қатысуға шешім қабылдады. Мақсаты - әйгілі венгрлік бақташы иттердің Пули тұқымының керемет интеллектіне таң қала отырып, соңғы *Pulibot*-ты бағдарламалау.

Pulibot $(H + 2) \times (W + 2)$ ұяшықтар торынан тұратын лабиринтте сыналады. Тордың жолдары солтүстіктен оңтүстікке қарай -1 -ден H -ға дейін, ал тордың бағандары батыстан шығысқа қарай -1 -дан W -ға дейін нөмірленеді. Тордың r жолы мен c бағанында орналасқан ұяшықты $(-1 \leq r \leq H, -1 \leq c \leq W)$ (r, c) ұяшығы деп атаймыз.

$0 \leq r < H$ және $0 \leq c < W$ болатын (r, c) ұяшығын қарастырайық. (r, c) ұяшығына **іргелес** 4 ұяшықтары бар:

- $(r, c - 1)$ ұяшығы (r, c) ұяшығының **батыс** ұяшығы деп аталады;
- $(r + 1, c)$ ұяшығы (r, c) ұяшығының **оңтүстігіндегі** ұяшығы деп аталады;
- $(r, c + 1)$ ұяшығы (r, c) ұяшығының **шығыс** ұяшығы деп аталады;
- $(r - 1, c)$ ұяшығы (r, c) ұяшығының **солтүстігіндегі** ұяшығы деп аталады.

$r = -1$ немесе $r = H$ немесе $c = -1$ немесе $c = W$ орындалса, (r, c) ұяшығы лабиринттің **шекара** ұяшығы деп аталады. Лабиринттің шекаралық ұяшығы болып табылмайтын әрбір ұяшық **кедергі** ұяшық немесе **бос** ұяшық болып табылады. Сонымен қатар, әрбір бос ұяшықта 0 ден Z_{MAX} қоса алғанда, теріс емес бүтін санмен белгіленген **түсі** болады. Бастапқыда әрбір бос ұяшықтың түсі 0.

Мысал ретінде, $H = 4, W = 5$ және $(1, 3)$ кедергі ұяшығы бар лабиринтті қарастырайық:

	-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							

Жалғыз кедергі ұяшық крестпен белгіленген. Лабиринттің шекаралық ұяшықтар көлеңкеленген. Әрбір бос ұяшықтағы сан оның түсін білдіреді.

(r_0, c_0) ұяшығынан (r_ℓ, c_ℓ) ұяшығына дейінгі ℓ ($\ell > 0$) ұзындығы **жол** деп - әрбір i ($0 \leq i < \ell$) үшін (r_i, c_i) және (r_{i+1}, c_{i+1}) ұяшықтары іргелес болатын бос $(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$ ұяшықтардан тұратын тізбекті айтамыз.

Ұзындығы ℓ жолда $\ell + 1$ ұяшықтар бар екенін ескеріңіз.

Байқауда зерттеушілер $(0, 0)$ ұяшығынан $(H - 1, W - 1)$ ұяшығына дейін кем дегенде бір жол бар лабиринтті орнатты. Бұл $(0, 0)$ және $(H - 1, W - 1)$ ұяшықтарының бос болуына кепілдік беретінін ескеріңіз. Ханга лабиринттің қай ұяшықтары бос, ал қай ұяшықтар кедергі екенін білмейді.

Сіздің міндетіңіз зерттеушілер орнатқан лабиринтте $(0, 0)$ ұяшығынан $(H - 1, W - 1)$ ұяшығына дейін *ең қысқа жолды* (яғни ең аз ұзындықтағы жолды) таба алатындай етіп, Хангаға Pulibot-ты бағдарламалауға көмектесу. Pulibot-тың сипаттамасы және байқау ережелері төменде сипатталған.

Бұл есептің мәлімдемесінің соңғы бөлімінде Pulibot визуализациясы үшін пайдалануға болатын көрсету құралы сипатталатынын ескеріңіз.

Pulibot's Specification

Әрбір $-1 \leq r \leq H$ және $-1 \leq c \leq W$ үшін (r, c) ұяшығының **күйін** бүтін сан ретінде анықтаймыз:

- егер (r, c) ұяшығы шекаралық ұяшық болса, оның күйі -2 ;
- егер (r, c) ұяшығы кедергі ұяшығы болса, оның күйі -1 ;
- егер (r, c) ұяшығы бос ұяшық болса, оның күйі ұяшықтың түсі болып табылады.

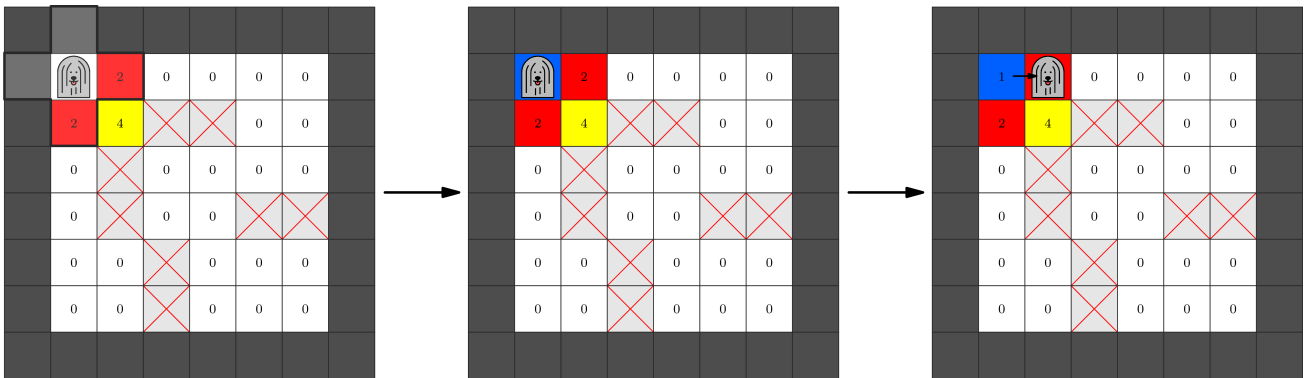
Pulibot бағдарламасы қадамдар тізбегі ретінде орындалады. Әрбір қадамда Pulibot жақын орналасқан ұяшықтардың күйлерін *таниды*, содан кейін нұсқауды орындайды. Ол орындайтын нұсқау *танылған* ұяшықтардың күйі арқылы анықталады. Нақтырақ сипаттама төменде жазылған.

Қадамның басында Pulibot (r, c) бос ұяшығында болсын делік. Қадам келесідей орындалады:

1. Біріншіден, Pulibot (r, c) ұяшығының және барлық көрші ұяшықтардың күйінен тұратын **күй массивін** таниды, яғни $S = [S[0], S[1], S[2], S[3], S[4]]$ массивін таниды, мұндағы:
 - $S[0]$ — (r, c) ұяшығының күйі.
 - $S[1]$ — батыстағы ұяшықтың күйі.
 - $S[2]$ — оңтүстіктегі ұяшықтың күйі.
 - $S[3]$ — шығыстағы ұяшықтың күйі.

- $S[4]$ — солтүстіктегі ұяшықтың күйі.
- 2. Содан кейін Pulibot танылған күй массивіне сәйкес келетін **нұсқауды** (Z, A) арқылы анықтайды.
- 3. Соңында Pulibot сол нұсқауды орындайды: ол (r, c) ұяшығының түсін Z деп орнатады, содан кейін ол келесі әрекеттердің бірі болып табылатын A әрекетін орындайды:
 - (r, c) ұяшығында қалу;
 - 4 іргілес ұяшықтардың біріне жылжу;
 - бағдарламаны тоқтату.

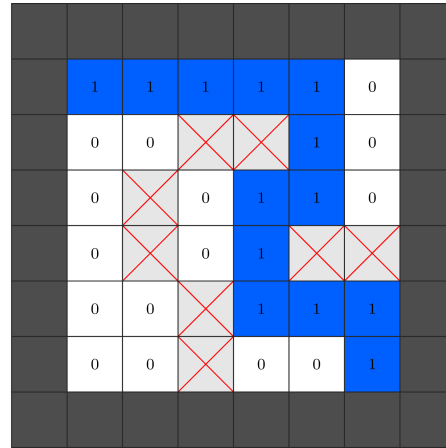
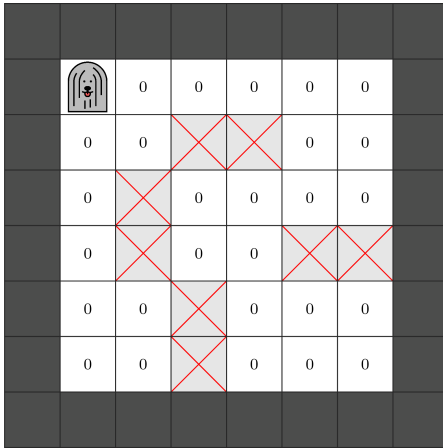
Мысалы, келесі суреттің сол жағында көрсетілген сценарийді қарастырайық. Pulibot қазір 0 түсі бар $(0, 0)$ ұяшығында тұр. Pulibot $S = [0, -2, 2, 2, -2]$ күй массивін таниды. Суреттің ортасында және оң жағында көрсетілгендей Pulibot осы массивті танығаннан кейін ұяшықтың түсін $Z = 1$ етіп орнататын, содан кейін шығысқа қарай жылжитын бағдарламасы болуы мүмкін.



Robot Contest Rules

- Бастапқыда Pulibot $(0, 0)$ ұяшығына орналасады және өз бағдарламасын орындауға кіріседі.
- Pulibotқа бос емес ұяшыққа өтуге рұқсат етілмейді.
- Pulibot бағдарламасы ең көбі 500 000 қадамдардан кейін аяқталуы керек.
- Pulibot бағдарламасы аяқталғаннан кейін лабиринттегі бос ұяшықтарды келесідей бояу керек:
 - $(0, 0)$ бастап $(H - 1, W - 1)$ дейінгі ең қысқа жолдағы әрбір ұяшықтың түсін 1ге
 - Барлық басқа бос ұяшықтардың түсі 0ге бояу қажет.
- Pulibot кез келген бос ұяшықта өз бағдарламасын тоқтата алады.

Мысалы, келесі суретте $H = W = 6$ болатын мүмкін лабиринт көрсетілген. Бастапқы конфигурация сол жақта және аяқтағаннан кейін бос ұяшықтардың бір қолайлы боялғаны оң жақта көрсетілген:



Implementation Details

Сізге келесі функцияны іске асыру қажет.

```
void program_pulibot()
```

- Бұл функция Pulibot бағдарламасын жасауы керек. Бұл бағдарлама H және W барлық мәндері және тапсырма шектеулеріне сәйкес келетін кез келген лабиринт үшін дұрыс жұмыс жасауы керек.
- Бұл функция әрбір сынақ жағдайы үшін бір рет шақырылады.

Бұл функция Pulibot бағдарламасын жасау үшін келесі функцияны шақыра алады:

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

- S : күй массивін сипаттайтын ұзындығы 5 массив.
- Z : түсті білдіретін теріс емес бүтін сан.
- A : төмендегідей Pulibot әрекетін білдіретін бір таңба:
 - H: өз орнында қалу;
 - W: батысқа қарай жылжу;
 - S: оңтүстікке жылжу;
 - E: шығысқа қарай жылжу;
 - N: солтүстікке жылжу;
 - T: бағдарламаны тоқтату.
- Бұл функция Pulibotқа S күй массивін тану кезінде (Z, A) нұсқаулығын орындауға нұсқау береді.

Бір күй массиві S ті бірнеше рет қайталап шақырса "Output isn't correct" деген вердикт алады.

Әрбір мүмкін S күй массивіне `set_instruction` шақыру қажет емес. Дегенмен, Pulibot кейінірек нұсқау орнатылмаған күй массивін танитын болса, онда «Output isn't correct» деген вердикт аласыз.

program_pulibot аяқталғаннан кейін, грейдер бір немесе бірнеше лабиринт бойынша Pulibot бағдарламасын шақырады. Бұл шақырулар шешіміңіздің уақыт шегіне *есептелмейді*. Грейдер *бейімделмейді*, яғни лабиринттер жиынтығы әрбір сынақ жағдайында алдын ала анықталған.

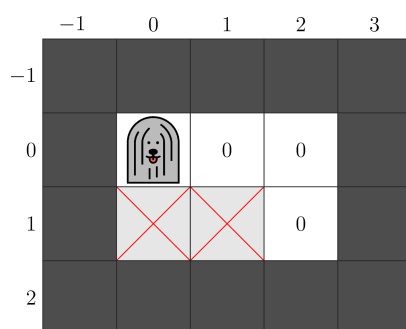
Егер Pulibot өз бағдарламасын тоқтатпас бұрын робот байқауы ережелерінің кез келгенін бұзса, сіз «Output isn't correct» деген вердикт аласыз.

Example

program_pulibot функциясы set_instruction-ға келесідей шақыруы мүмкін:

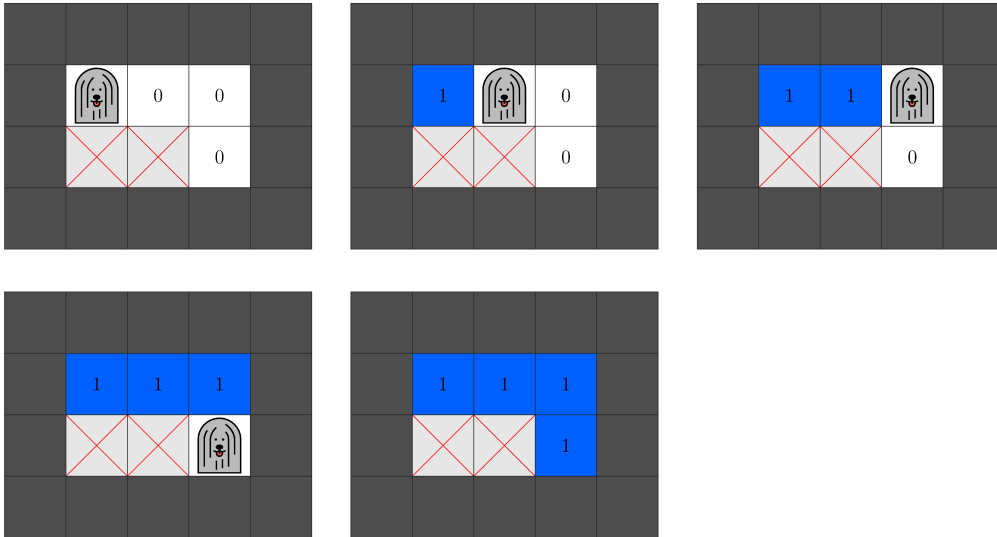
Шақыру	S күй массивіне арналған нұсқаулық
set_nұсқау([0, -2, -1, 0, -2], 1, E)	Түсті 1 етіп орнатып, шығысқа қарай жылжытыңыз
set_nұсқау([0, 1, -1, 0, -2], 1, E)	Түсті 1 етіп орнатып, шығысқа қарай жылжытыңыз
set_nұсқау([0, 1, 0, -2, -2], 1, S)	Түсті 1 етіп орнатып, оңтүстікке қарай жылжытыңыз
set_nұсқау([0, -1, -2, -2, 1], 1, T)	Түсті 1 етіп орнатыңыз және бағдарламаны аяқтаңыз

$H = 2$ және $W = 3$ және лабиринт келесі суретте көрсетілетін сценарийді қарастырайық.



Бұл арнайы лабиринт үшін Pulibot бағдарламасы төрт қадаммен жұмыс істейді. Пулибот танитын күй массивтері және олар орындайтын нұсқаулар жоғарыда ретпен жасалған set_instruction төрт шақыруына дәл сәйкес келеді. Осы нұсқаулардың соңғысы бағдарламаны тоқтатады.

Келесі суретте төрт қадамның әрқайсысының алдындағы лабиринт және аяқталғаннан кейінгі соңғы түстер көрсетілген.



Дегенмен, 4 нұсқауларынан тұратын бұл бағдарлама басқа жарамды лабиринттерде ең қысқа жолды таба алмауы мүмкін екенін ескеріңіз. Сондықтан, бұл шешім "Output isn't correct" деген вердикт алады.

Constraints

$Z_{MAX} = 19$. Демек, Pulibot 0-ден 19-ға дейінгі түстерді қолдана алады.

Pulibotты сынау үшін пайдаланылатын әрбір лабиринт үшін:

- $2 \leq H, W \leq 15$
- $(0, 0)$ ұяшығынан $(H - 1, W - 1)$ ұяшығына кемінде бір жол бар.

Subtasks

1. (6 ұпай) Лабиринтте кедергі ұяшығы жоқ.
2. (10 ұпай) $H = 2$
3. (18 ұпай) Әр жұп бос ұяшықтардың арасында дәл бір жол бар.
4. (20 ұпай) $(0, 0)$ ұяшығынан $(H - 1, W - 1)$ ұяшығына дейінгі әрбір ең қысқа жолдың ұзындығы $H + W - 2$ болады.
5. (46 ұпай) Қосымша шектеулер жоқ.

Егер сынақ жағдайларының кез келгенінде `set_instruction` процедурасына немесе Pulibot бағдарламасына оның орындалу барысындағы шақырулар Implementation Details шектеулеріне сәйкес келмесе, сол ішкі үшін шешіміңіздің ұпайы 0 болады.

Әрбір ішкі есепте дұрыс бояуды шығару арқылы жартылай ұпай алуға болады.

Ресми түрде:

- Сынақ жағдайының шешімі **толық** болып табылады, егер бос ұяшықтардың соңғы бояуы роботтар байқауының ережелеріне сәйкес келсе.

- Сынақ жағдайының шешімі **жартылай** болып табылады, егер соңғы бояу келесідей көрінсе:
 - $(0, 0)$ бастап $(H - 1, W - 1)$ дейінгі ең қысқа жол бар, ол үшін жолға енгізілген әрбір ұяшықтың түсі 1 болады.
 - Торда 1 түсі бар басқа бос ұяшық жоқ.
 - Тордағы кейбір бос ұяшықтардың түсі 0 және 1 емес.

Сынақ жағдайына сіздің шешіміңіз толық немесе жартылай болмаса, сәйкес сынақ жағдайы үшін сіздің ұпайыңыз 0 болады.

1-4 ішкі есепте толық шешімнің бағасы 100%, ал жартылай шешім оның ішкі есептің ұпайының 50% құрайды.

5-ші ішкі есепті сіздің ұпайыңыз Pulibot бағдарламасында қолданылатын түстер санына байланысты. Дәлірек айтқанда, Z^* арқылы `set_instruction` үшін жасалған барлық қоңыраулардағы Z тің максималды мәнін белгілейік. Сынақ жұмысының баллы келесі кесте бойынша есептеледі:

Шарты	Ұпай (толық)	Ұпай (жартылай)
$11 \leq Z^* \leq 19$	$20 + (19 - Z^*)$	$12 + (19 - Z^*)$
$Z^* = 10$	31	23
$Z^* = 9$	34	26
$Z^* = 8$	38	29
$Z^* = 7$	42	32
$Z^* \leq 6$	46	36

Әрбір ішкі есептің ұпайы - бұл ішкі тапсырмадағы сынақ жағдайлары үшін ұпайлардың ең азы.

Sample Grader

Үлгі грейдер енгізуді келесі форматта оқиды:

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

Мұнда m — лабиринттің шекаралық емес ұяшықтарын сипаттайтын W бүтін сандарының тұратын H массивтерінің массиві. $m[r][c] = 0$, егер (r, c) ұяшығы бос ұяшық болса және $m[r][c] = 1$, егер (r, c) ұяшығы кедергі ұяшық болса.

Үлгі грейдер басында `program_pulibot()` шақырады. Грейдер үлгісі хаттаманың бұзылуын анықтаса, үлгілік бағалаушы `Protocol Violation: <MSG>` басып шығарады және

тоқтатады, мұндағы <MSG> келесі қате туралы хабарлардың бірі болып табылады:

- Invalid array: $-2 \leq S[i] \leq Z_{MAX}$ кейбір i үшін сәйкес келмейді немесе S ұзындығы 5 емес.
- Invalid color: $0 \leq Z \leq Z_{MAX}$ орындалмайды.
- Invalid action: A таңбасы H, W, S, E, N немесе T ға тең емес.
- Same state array: set_instruction бірдей S үшін екі рет шақырылады.

Әйтпесе, program_pulibot аяқталған кезде, үлгі грейдер кіріспен сипатталған лабиринтте Pulibot бағдарламасын орындайды.

Үлгі грейдер екі нәтиже шығарады.

Алдымен грейдер үлгісі Pulibot әрекеттерінің журналын жұмыс каталогындағы robot.bin файлына жазады. Бұл файл келесі бөлімде сипатталған визуализация құралының кірісі ретінде қызмет етеді.

Екіншіден, Pulibot бағдарламасы сәтті аяқталмаса, үлгі грейдер келесі қате туралы хабарлардың бірін басып шығарады:

- Unexpected state: Pulibot set_instruction шақырылмаған күй массивін таныды.
- Invalid move: әрекетті орындау Pulibotтың бос емес ұяшыққа баруына әкелді.
- Too many steps: Pulibot бағдарламасын тоқтатпай 500 000 қадамдарын орындады.

Әйтпесе, Pulibot бағдарламасы аяқталғаннан кейін $e[r][c]$ (r, c) ұяшығының күйі болсын. Үлгі грейдер H жолдарын келесі пішімде басып шығарады:

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

Display Tool

Бұл тапсырманың тіркеме архивы display.py деп аталатын файлды қамтиды. Қолданылған кезде бұл Python сценарийі үлгі грейдер енгізуімен сипатталған лабиринттегі Pulibot әрекеттерін көрсетеді. Ол үшін robot.bin екілік файлы жұмыс каталогында болуы керек.

Сценарийді шақыру үшін келесі команданы орындаңыз.

```
python3 display.py
```

Қарапайым графикалық интерфейс көрсетіледі. Негізгі ерекшеліктері келесідей:

- Толық лабиринттің күйін байқауға болады. Pulibotтың қазіргі орны тіктөртбұрышпен ерекшеленген.
- Сіз Pulibot қадамдарын көрсеткі түймелерді басу немесе олардың жылдам пернелерін басу арқылы шолуға болады. Сондай-ақ белгілі бір қадамға өтуге болады.

- Pulibot бағдарламасындағы алдағы қадам төменгі жағында көрсетілген. Ол ағымдағы күй массивін және ол орындайтын нұсқауды көрсетеді. Соңғы қадамнан кейін ол бағалаушының қате туралы хабарларының бірін немесе бағдарлама сәтті аяқталса, «Terminated» көрсетеді.
- Түсті білдіретін әрбір санға көрнекі өң түсін, сондай-ақ дисплей мәтінін тағайындай аласыз. Дисплей мәтіні - бұл түсі бар әрбір ұяшықта пайда болатын қысқа жол. Фондық түстерді тағайындауға және мәтіндерді келесі жолдардың кез келгенімен көрсетуге болады:
 - «Colors» түймесін басқаннан кейін оларды диалогтық терезеде орнатыңыз.
 - `colors.txt` файлының мазмұнын өңдеңіз.
- `robot.bin` қайта жүктеу үшін `Reload` түймесін пайдаланыңыз. `robot.bin` мазмұны өзгерсе, бұл пайдалы.