



Робот мелдеши

Сегед университетинин AI(жасалма интеллект) изилдөөчүлөрү роботторду программалоо боюнча сынак өткөрүп жатышат. Досуңуз Ханга мелдешке катышууну чечти. Максаты - атактуу венгриялык койчунун жардамчы ит тукумунун Пулидин улуу интеллекте суктанган акыркы *Pulibot* программалоо.

Pulibot $(H + 2) \times (W + 2)$ торчолордон турган лабиринтте сыналат. Тордун катарлары түндүктөн түштүккө -1 ден H ге чейин, ал эми тордун мамылары батыштан чыгышка чейин -1 ден W ге чейин номерленген. Тордун r сабында жана c тилкесинде жайгашкан клетканы $(-1 \leq r \leq H, -1 \leq c \leq W)$ (r, c) клеткасы деп атайбыз.

$0 \leq r < H$ жана $0 \leq c < W$ болгон (r, c) клеткасын карап көрөлү. (r, c) клеткасына **коңшу** 4 клеткалары бар:

- $(r, c - 1)$ клеткасы (r, c) клеткасынын **батышындагы** клеткасы деп аталат;
- $(r + 1, c)$ клеткасы (r, c) клеткасынын **түштүгүндөгү** клеткасы деп аталат;
- $(r, c + 1)$ клеткасы (r, c) клеткасынын **чыгышындагы** клеткасы деп аталат;
- $(r - 1, c)$ клеткасы (r, c) клеткасынын **түндүгүндөгү** клеткасы деп аталат.

$r = -1$ же $r = H$ же $c = -1$ же $c = W$ орун алса, (r, c) клеткасы лабиринттин **чек ара** клеткасы деп аталат. Лабиринттин чек ара клеткасы болбогон ар бир клетка же **тоскоолдук** клетка же **бош** клетка. Кошумчалай кетсек, ар бир бош клетканын **түсү** бар, ал 0 жана Z_{MAX} ортосундагы терс эмес бүтүн сан менен көрсөтүлгөн. Башында, ар бир бош клетканын түсү 0 болуп саналат.

Мисалы, $H = 4$ жана $W = 5$ болгон лабиринтти карап көрөлү, анда $(1, 3)$ бир тоскоолдук клеткасы бар:

	-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							

Жалгыз тоскоолдук клеткасы крест менен белгиленген. Лабиринттин чек ара клеткалары карага боёлгон. Ар бир бош клеткадагы сан анын түсүн билдирет.

(r_0, c_0) клеткасынан (r_ℓ, c_ℓ) клеткасына ℓ ($\ell > 0$) узундуктагы **жол** экиден айырмаланган *бош* клеткалардын ырааттуулугу $(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$ ар бир i ($0 \leq i < \ell$) үчүн (r_i, c_i) жана (r_{i+1}, c_{i+1}) клеткалары жана.

ℓ узундуктагы жол так $(\ell + 1)$ клеткаларын камтыганын эске алыңыз.

Мелдеште изилдөөчүлөр $(0, 0)$ клеткасынан $(H - 1, W - 1)$ клеткасына чейин жок дегенде бир жол бар болгон лабиринт түзүштү. Бул $(0, 0)$ жана $(H - 1, W - 1)$ клеткаларынын бош болушуна кепилдик берилгенин эске алыңыз.

Ханга лабиринттин кайсы клеткалары бош, кайсы клеткалар тоскоолдуктар экенин билбейт.

Сиздин милдетиңиз Хангага *Pulibot*тун программалоого жардам берүү, ал $(0, 0)$ клеткасынан $(H - 1, W - 1)$ клеткасына чейин *эң кыска жолду* (б.а. минималдуу узундуктагы жолду) таба алат. Изилдөөчүлөр тарабынан орнотулган белгисиз лабиринтте. *Pulibot*тун спецификациясы жана мелдештин эрежелери төмөндө баяндалган.

Бул көйгөй билдирүүсүнүн акыркы бөлүмүндө сиз *Pulibot*тун визуалдаштыруу үчүн колдоно турган дисплей куралы сүрөттөлгөнүнө көңүл буруңуз.

Pulibotтун спецификациясы

Ар бир $-1 \leq r \leq H$ жана $-1 \leq c \leq W$ үчүн (r, c) клеткасынын **абалын** бүтүн сан катары аныктаңыз:

- эгерде (r, c) клеткасы чектик клетка болсо, анда анын абалы -2 ;
- эгерде (r, c) клеткасы тоскоолдук клеткасы болсо, анда анын абалы -1 ;
- эгерде (r, c) клеткасы бош клетка болсо, анда анын абалы клетканын түсү болуп саналат.

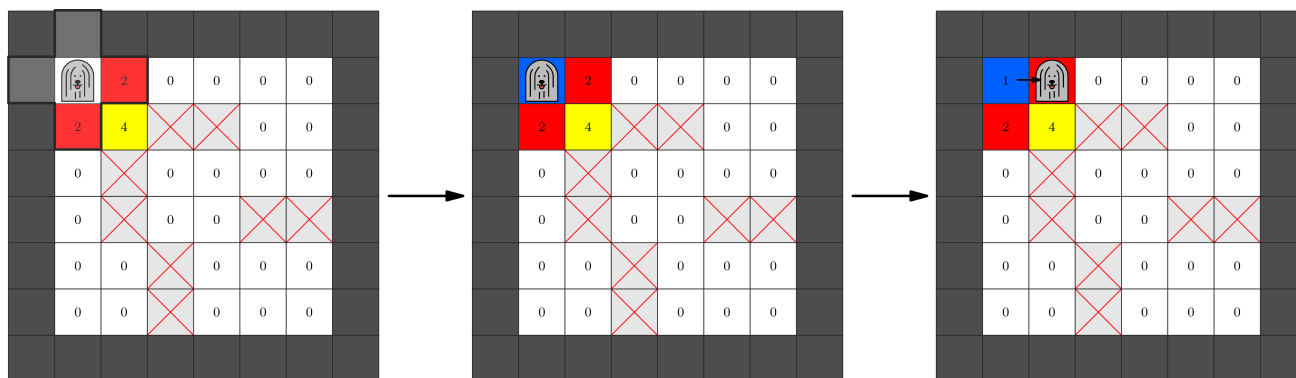
*Pulibot*тун программасы кадамдардын ырааттуулугу катары аткарылат. Ар бир кадамда *Pulibot* коңшу жайгашкан клеткалардын абалын таанып, андан кийин көрсөтмөлөрдү аткарат. Ал аткарган көрсөтмө таанылган абалдар тарабынан аныкталат. Төмөндө дагы так аныктамасы келтирилген.

Учурдагы кадамдын башында *Pulibot* (r, c) клеткасында, ал бош клеткада. Кадам төмөнкүдөй жүзөгө ашырылат:

1. Биринчиден, *Pulibot* учурдагы **абалдардын массивин**, башкача айтканда $S = [S[0], S[1], S[2], S[3], S[4]]$ массивдерин тааныйт. (r, c) клеткасынын жана бардык коңшу клеткалардын абалынан турат:
 - $S[0]$ — (r, c) клеткасынын абалы.
 - $S[1]$ — батыштагы клетканын абалы.

- $S[2]$ — түштүктөгү клетканын абалы.
 - $S[3]$ — чыгыштагы клетканын абалы.
 - $S[4]$ — түндүктөгү клетканын абалы.
2. Андан кийин, *Pulibot* таанылган абалдардын массивине туура келген **инструкцияны** (Z, A) аныктайт.
 3. Акырында, *Pulibot* бул көрсөтмөнү аткарат: (r, c) клеткасынын түсүн Z түсүнө өзгөртөт. андан кийин ал A аракетин аткарат, бул төмөнкү аракеттердин бири:
 - (r, c) клеткасында *калуу*;
 - 4 коңшу клеткалардын бирине *жылуу*;
 - *программаны токтотуу*.

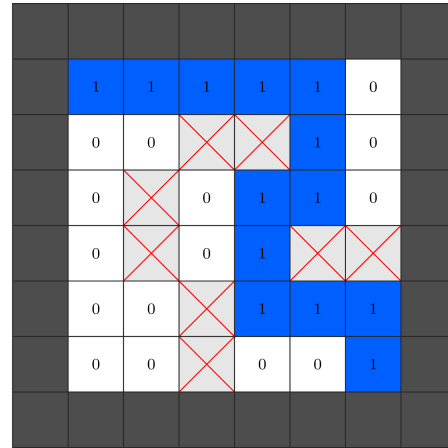
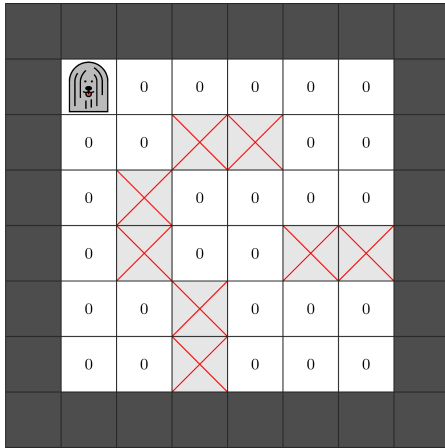
Мисалы, төмөнкү сүрөттүн сол жагында көрсөтүлгөн сценарийди карап көрөлү. *Pulibot* учурда 0 түсү менен $(0, 0)$ клеткасында. *Pulibot* $S = [0, -2, 2, 2, -2]$ абалдардын массивин тааныйт. *Pulibot*то бул массивди тааныгандан кийин учурдагы клетканын түсүн $Z = 1$ кылып коюп, андан кийин чыгышка жыла турган программа болушу мүмкүн, сүрөттүн ортосунда жана оң жагында көрсөтүлгөндөй:



Робот мелдешинин эрежелери

- Башында *Pulibot* $(0, 0)$ клеткасына жайгаштырылат жана өзүнүн программасын аткара баштайт.
- *Pulibot*ко бош эмес клеткага өтүүгө уруксат берилбейт.
- *Pulibot*тун программасы эң көп 500 000 кадамдан кийин токтотулушу керек.
- *Pulibot*тун программасы аяктагандан кийин лабиринттеги бош клеткалар төмөнкүдөй түстө болушу керек:
 - $(0, 0)$ дөн $(H - 1, W - 1)$ ге чейин эң кыска жолдун ар бир клеткасынын түсүн 1 кылуу керек.
 - Бардык башка бош клеткалардын түсүн 0 кылуу керек.
- *Pulibot* өзүнүн программасын каалаган бош клеткада токтото алат.

Мисалы, төмөнкү сүрөттө $H = W = 6$ менен мүмкүн болгон лабиринт көрсөтүлгөн. Баштапкы конфигурация сол жакта жана бош клеткалардын бир мүмкүн болгон түсү аяктагандан кийин оң жакта көрсөтүлгөн:



Ишке ашыруу деталдары

Сиз төмөнкү процедураны ишке ашырууңуз керек.

```
void program_pulibot()
```

- Бул процедура *Pulibot*тун программасын түзүшүңүз керек. Бул программа *H* жана *W* бардык маанилери жана тапшырма чектөөлөрүнө жооп берген лабиринт үчүн туура иштеши керек.
- Бул процедура ар бир тест учуру үчүн бир жолу чакырылат.

Бул процедура *Pulibot* программасын жасоо үчүн төмөнкү процедурага чакырууларды жасай алат:

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

- *S*: Абалдардын массивин сүрөттөгөн 5 узундуктагы массив.
- *Z*: түстү билдирген терс эмес бүтүн сан.
- *A*: төмөнкүдөй *Pulibot*тун аракетин билдирген бир символ:
 - H: калуу;
 - W: батышка жылдыруу;
 - S: түштүккө жылдыруу;
 - E: чыгышка жылдыруу;
 - N: түндүккө жылдыруу;
 - T: программаны токтотуу.
- Бул процедураны чакыруу *Pulibot*ко *S* абалдардын массивин тааныгандан кийин ал (*Z*, *A*) көрсөтмөсүн аткарат

Бул процедураны бир эле *S* абалдардын массиви менен бир нече жолу кайталоо Output isn't correct деген жооп берет.

Ар бир мүмкүн болгон абалдардын массиви S менен `set_instruction` деп чакыруу талап кылынбайт. Бирок, кийинчерээк *Pulibot* инструкция коюлбаган абалдардын массивин тааныса, сизге `Output isn't correct` деген жооп берет.

`program_pulibot` аяктагандан кийин, грейдер бир же бир нече лабиринт боюнча *Pulibot*тун программасын чакырат. Бул чакырыктар сиздин чыгарылышыңыздын убактысына эсептелбейт. Грейдер адаптивдүү эмес, башкача айтканда лабиринттердин жыйындысы ар бир тест алдын ала аныкталган.

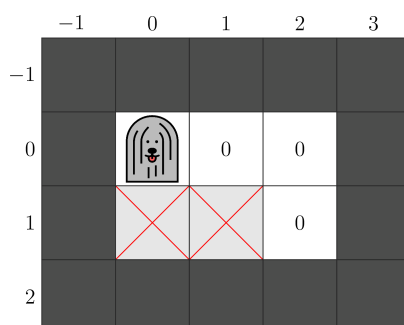
Эгер *Pulibot* өзүнүн программасын токтотконго чейин робот мелдешинин эрежелеринин бирин бузса, сизге `Output isn't correct` деген жооп берет.

Мисал

`program_pulibot` процедурасы `set_instruction`га төмөнкүдөй чакырууларды жасай алат:

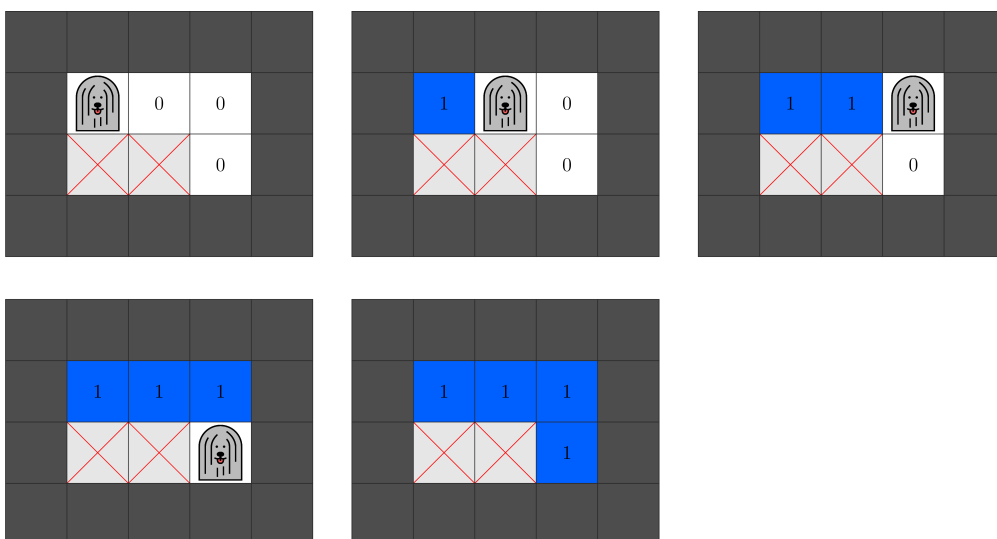
Чакыраа	Абалдардын массиви үчүн инструкция S
<code>set_instruction([0, -2, -1, 0, -2], 1, E)</code>	Клетканын түсүн 1 кылып, чыгышка жылыңыз
<code>set_instruction([0, 1, -1, 0, -2], 1, E)</code>	Клетканын түсүн 1 кылып коюп, чыгышка жылыңыз
<code>set_instruction([0, 1, 0, -2, -2], 1, S)</code>	Клетканын түсүн 1 кылып коюп, түштүккө жылыңыз
<code>set_instruction([0, -1, -2, -2, 1], 1, T)</code>	Клетканын түсүн 1 кылып коюп, программаны токтотуңуз

$H = 2$ жана $W = 3$ болгон сценарийди карап көрөлү жана лабиринт төмөнкү сүрөттө көрсөтүлгөн.



Бул лабиринт үчүн *Pulibot*тун программасы төрт чакыруудан турат. *Pulibot* тааныган абалдардын массиви жана ал аткарган көрсөтмөлөр ирети менен жогоруда жасалган `set_instruction` төрт чалууга дал келет. Бул инструкциянын акыркысы программаны токтотот.

Төмөнкү сүрөттө төрт чакыруунун ар биринин алдындагы лабиринт жана аяктагандан кийинки акыркы түстөр көрсөтүлгөн.



Бирок, 4 инструкциясынан турган бул программа башка жарактуу лабиринттерде эң кыска жолду таппай калышы мүмкүн экенин эске алыңыз. Ошондуктан, эгер тапшырылса, ал `Output isn't correct` деген жооп берет.

Чектөөлөр

$Z_{MAX} = 19$. Демек, *Pulibot* 0дөн 19га чейинки түстөрдү колдоно алат.

*Pulibot*ту тест үчүн колдонулган ар бир лабиринт үчүн:

- $2 \leq H, W \leq 15$
- $(0, 0)$ клеткасынан $(H - 1, W - 1)$ клеткасына чейин жок дегенде бир жол бар.
-

Кошумча тапшырмалар

1. (6 упай) Лабиринтте эч кандай тоскоолдук клеткасы жок.
2. (10 упай) $H = 2$
3. (18 упай) Ар бир жуп бош клетканын ортосунда так бир жол бар.
4. (20 упай) $(0, 0)$ клеткасынан $(H - 1, W - 1)$ клеткасына чейинки ар бир эң кыска жолдун узундугу $H + W - 2$.
5. (46 упай) Кошумча чектөөлөр жок.

Эгерде тест учурларынын кайсы биринде `set_instruction` процедурасына же *Pulibot* программасына анын аткарылышына карата чакыруулар Ишке ашыруунун чоо-жайында сүрөттөлгөн чектөөлөргө туура келбесе, ал кошумча тапшырма үчүн чыгарылышыңыздын упайы 0 болот.

Ар бир кошумча тапшырмада сиз дээрлик туура болгон боё чыгаруу менен жарым-жартылай упай ала аласыз.

Расмий түрдө:

- Эгерде бош уячалардын акыркы боёгу Робот мелдешинин эрежелерине жооп берсе, тесттик иштин жыйынтыгы **толук** болуп саналат.
- Тесттердин чечими **жарым-жартылай** болуп саналат, эгерде акыркы боё төмөнкүдөй болсо:
 - $(0, 0)$ дөн $(H - 1, W - 1)$ ге чейин эң кыска жол бар, ал үчүн жолго киргизилген ар бир клетканын түсү 1.
 - Тордо 1 түсү бар башка бош клетка жок.
 - Тордогу кээ бир бош клетканын түсү 0 жана 1 эмес.

Эгерде сиздин чыгарылышыңыз толук же жарым-жартылай болбосо, тиешелүү тест иши боюнча упайыңыз 0 болот.

1-4-кошумча тапшырмаларда толук упай үчүн 100% , ал эми тесттик иштин жарым-жартылай чыгарылышы үчүн упай анын кошумча тапшырмасы үчүн упайдын 50% түзөт.

5-кошумча тапшырмада сиздин упайыңыз *Pulibot* программасында колдонулган түстөрдүн санына жараша болот. Тагыраак айтканда, Z^* менен `set_instruction` үчүн жасалган бардык чакыруулардагы Z максималдуу маанисин белгилеңиз. Тесттин упайы төмөнкү таблица боюнча эсептелет:

Шарт	Упай (толук)	Упай (жарым-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

Ар бир кошумча тапшырманын упайы – бул кошумча тапшырмадагы тесттик иштердин минималдуу упайы.

Үлгү Грейдер

Үлгү грейдер киргизүүнү төмөнкү форматта окуйт:

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

Бул жерде, m лабиринттин чек арадагы эмес клеткаларын сүрөттөгөн W бүтүн сандардын H массивдеринин массиви. $m[r][c] = 0$, эгерде (r, c) клеткасы бош клетка болсо, $m[r][c] = 1$, (r, c) клеткасы тоскоолдук клеткасы болсо.

Үлгү грейдер биринчи `program_pulibot()` чакырат. Эгерде үлгүдөгү грейдер протоколдун бузулушун аныктаса `Protocol Violation: <MSG>` деп чыгарат жана токтотулат, мында `<MSG>` төмөнкү каталардын бири болуп саналат:

- `Invalid array`: $-2 \leq S[i] \leq Z_{MAX}$ кээ бир i үчүн туура келбейт же S узундугу 5 эмес.
- `Invalid color`: $0 \leq Z \leq Z_{MAX}$ туура келбейт.
- `Invalid action`: A белгиси H, W, S, E, N же T символдорунун бири эмес.
- `Same state array`: `set_instruction` бир эле S массиви менен кеминде эки жолу чакырылган.

Болбосо, `program_pulibot` аяктаганда, үлгү грейдер *Pulibot*тун программасын киргизүү менен сүрөттөлгөн лабиринтте аткарат.

Үлгү грейдер эки жыйынтык чыгарат.

Биринчиси, үлгү грейдер жумушчу каталогдогу `robot.bin` файлына *Pulibot*тун аракеттеринин журналын жазат. Бул файл кийинки бөлүмдө сүрөттөлгөн визуалдаштыруу куралынын кириши катары кызмат кылат.

Экинчиси, *Pulibot* программасы ийгиликтүү аяктабаса, үлгү грейдер төмөнкү ката билдирүүлөрүнүн бирин басып чыгарат:

- `Unexpected state`: *Pulibot* `set_instruction` чакырылбаган абалдардын массивин тааныды.
- `Invalid move`: аракеттин натыйжасында *Pulibot* бош эмес клеткага жылды.
- `Too many steps`: *Pulibot* программасын токтотпостон 500 000 кадамдарды жасады.

Болбосо, *Pulibot* программасы аяктагандан кийин $e[r][c]$ (r, c) уячасынын абалы болсун. Үлгү грейдер H сызыктарын төмөнкү форматта чыгарат:

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

Дисплей куралы

Бул тапшырма үчүн тиркеме топтому `display.py` деп аталган файлды камтыйт. Чакырылганда, бул Python скрипти үлгү грейдердин киргизүүсү менен сүрөттөлгөн лабиринтте *Pulibot*тун аракеттерин көрсөтөт. Бул үчүн `robot.bin` бинардык файлы жумушчу каталогдо болушу керек.

Скрипти чакыруу үчүн, төмөнкү буйрукту аткарыңыз.


```
python3 display.py
```

Жөнөкөй графикалык интерфейс пайда болот. Негизги өзгөчөлүктөрү төмөнкүлөр болуп саналат:

- Толук лабиринттин абалын байкай аласыз. Пулиботтун учурдагы жайгашкан жери тик бурчтук менен белгиленген.
- Сиз жебе баскычтарын басуу же алардын `hotkeys` баскычтарын басып *Pulibot* кадамдарын карап чыга аласыз. Ошондой эле белгилүү бир кадамга секире аласыз.
- *Pulibot*тун программасында алдыдагы кадам ылдыйда көрсөтүлгөн. Ал учурдагы абалдардын массиви жана ал аткара турган инструкцияны көрсөтөт. Акыркы кадамдан кийин, ал грейдердин ката билдирүүлөрүнүн бирин көрсөтөт, же программа ийгиликтүү аяктаса `Terminated`.
- Түстү билдирген ар бир санга визуалдык фон түсүн, ошондой эле дисплей текстин дайындай аласыз. Дисплей тексти бул түстөгү ар бир уячада пайда боло турган кыска сап. Фондук түстөрдү дайындап, тексттерди төмөнкү жолдор менен көрсөтсөңүз болот:
 - `Colors` баскычын баскандан кийин аларды диалог терезесинде орнотуңуз.
 - `colors.txt` файлынын мазмунун түзөтүңүз.
- `robot.bin` кайра жүктөө үчүн, `Reload` баскычын колдонуңуз. Бул `robot.bin` мазмуну өзгөргөн болсо, пайдалуу.