

## Robot Contest

Ransakendur á gervigreindarsviði við Háskólann í Szeged eru að halda þjarka forritunarkeppni. Vinur þinn Hanga hefur ákveðið að taka þátt í keppninni. Markmiðið er að forrita hin fullkomna *Pulibot*, til heiðurs frábæru gáfna hinna frægu ungversku hundategundar, Puli.

Pulibot verður prófaður á völungarhúsi sem samanstendur af  $(H + 2) \times (W + 2)$  reitum. Raðirnar í völungarhúsinu eru númeraðar frá  $-1$  til  $H$  frá norðri til suðurs og dálkarnir eru númeraðir frá  $-1$  til  $W$  frá vestri til austurs. Við segjum að reitur í röð  $r$  og dálk  $c$  í völungarhúsinu ( $-1 \leq r \leq H$ ,  $-1 \leq c \leq W$ ) sé  $(r, c)$ .

Íhugið reit  $(r, c)$  þar sem  $0 \leq r < H$  og  $0 \leq c < W$ . Sá reitur hefur 4 **aðlæga** reiti:

- reitur  $(r, c - 1)$  er fyrir **vestan** reit  $(r, c)$ ;
- reitur  $(r + 1, c)$  er fyrir **sunnan** reit  $(r, c)$ ;
- reitur  $(r, c + 1)$  er fyrir **austan** reit  $(r, c)$ ;
- reitur  $(r - 1, c)$  er fyrir **norðan** reit  $(r, c)$ ;

Reitur  $(r, c)$  er kallaður **jaðar** reitur völungarhúsins ef  $r = -1$  or  $r = H$  eða  $c = -1$  or  $c = W$ . Hver reitur sem er ekki jaðar reitur, er annaðhvort **hindrunar** reitur eða **tómur** reitur. Þar að auki hefur hver tómur reitur **lit**, sem táknaður er með heiltölu á milli 0 og  $Z_{MAX}$ , báðar þar með taldar. Til að byrja með er liturinn á hverjum tómunum reit 0.

Skoðum völungarhúsið með  $H = 4$  og  $W = 5$ , sem inniheldur einn hindrunar reit  $(1, 3)$ :

	-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							

Eini hindrunar reiturinn er merktur með kross. Jaðar reitirnir eru dökkir. Tölurnar skrifaðar á hvern reit tákna litinn á þeim reit.

**Leið** af lengd  $\ell$  ( $\ell > 0$ ) frá reit  $(r_0, c_0)$  til reits  $(r_\ell, c_\ell)$  er röð af tveimur og tveimur ólíkum \*tómum\* reitum  $(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$  þar sem fyrir sérhvert  $i$  ( $0 \leq i < \ell$ ) þá eru reitirnir  $(r_i, c_i)$  og  $(r_{i+1}, c_{i+1})$  aðlægir hvorum öðrum.

Takið eftir að leið af lengd  $\ell$  inniheldur nákvæmlega  $\ell + 1$  reiti.

Í keppninni setja rannsakendurnir upp völundarhús þar sem til er allavega ein leið frá reit  $(0, 0)$  að reit  $(H - 1, W - 1)$ . Takið eftir að þetta leiðir af sér að reitir  $(0, 0)$  og  $(H - 1, W - 1)$  eru alltaf tómir.

Hanga veit ekki hvaða reitir í völundarhúsinu eru tómir og hvaða reitir eru hindranir.

Verkefnið þitt er að hjálpa Hanga við að forrita Pulibot þannig að hann getur fundið *stystu leið*, það er leið af minnstu lengd frá reit  $(0, 0)$  til reitar  $(H - 1, W - 1)$  í óþekkta völundarhúsinu sem sett var upp af rannsakendunum. Staðallinn sem Pulibot fylgir og reglur keppinnar er lýst hér að neðan.

## Staðallinn á Pulibot

Við skilgreinum **stöðu** reitar  $(r, c)$  fyrir sérhvert  $-1 \leq r \leq H$  og  $-1 \leq c \leq W$  sem heiltölu þannig að:

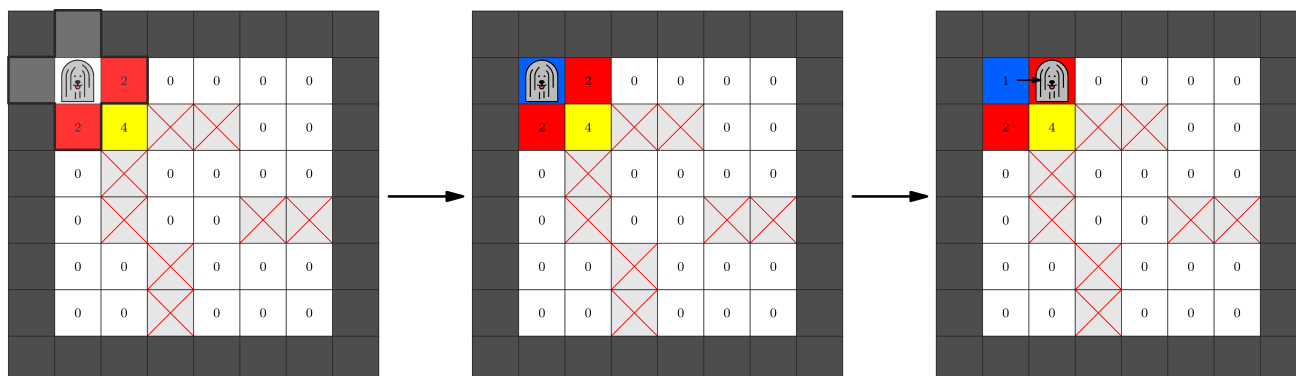
- Ef reitur  $(r, c)$  er jaðar reitur, þá er staða hans  $-2$ ;
- Ef reitur  $(r, c)$  er hindrunar reitur, þá er staða hans  $-1$ ;
- Ef reitur  $(r, c)$  er tómur reitur, þá er staða hans liturinn í þeim reit.

Forritið hans Pulibot er keyrt sem röð skrefa. Í hverju skrefi þá skoðar Pulibot stöðurnar á aðlægum reitum og framkvæmir síðan skipun. Skipunin sem hann framkvæmir er ákvörðuð út frá þessum stöðum. Nákvæmari skilgreining fylgir.

Gerum ráð fyrir að í byrjun á núverandi skrefi, þá er Pulibot í reit  $(r, c)$ , sem er tómur. Skrefið er framkvæmt á eftirfarandi hátt:

1. Fyrst skoðar Pulibot núverandi **stöðufylki**, það er fylkið  $S = [S[0], S[1], S[2], S[3], S[4]]$ , sem samanstendur af reitunum  $(r, c)$  og af öllum aðlægum reitum.
  - $S[0]$  er staðan í reit  $(r, c)$ .
  - $S[1]$  er staðan í reitnum til vesturs.
  - $S[2]$  er staðan í reitnum til suðurs.
  - $S[3]$  er staðan í reitnum til austurs.
  - $S[4]$  er staðan í reitnum til norðurs.
2. Síðan ákveður Pulibot **skrefið**  $(Z, A)$  sem samsvarar skoðuðum stöðum í stöðufylkinu.
3. Í lokin framkvæmir Pulibot skrefið: Hann breytir litnum í reit  $(r, c)$  yfir í  $Z$  og síðan framkvæmir aðgerð  $A$ , sem er eitt af eftirfarandi:
  - *halda sér* í reit  $(r, c)$ ;
  - *færa sig* í einn af 4 aðlægum reitunum;
  - *hætta keyrslu*.

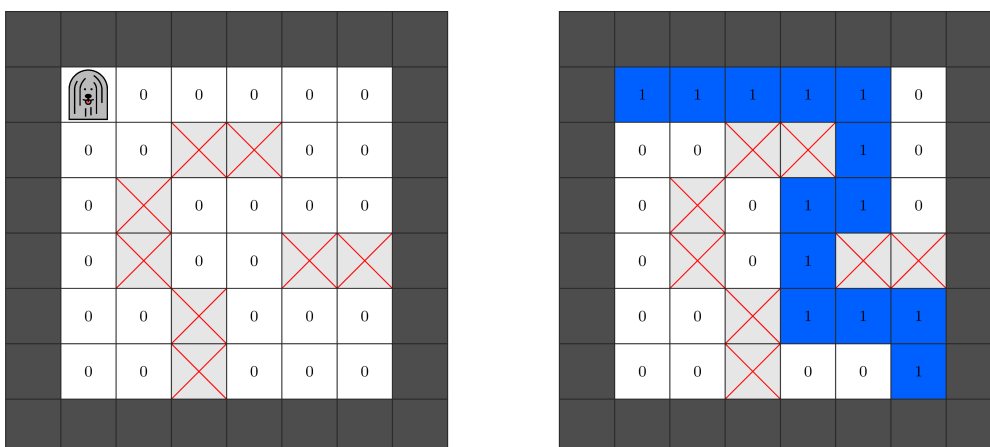
Til dæmis, íhugið atburðarrásina á eftirfarandi mynd sem byrjar á vinstri myndinni. Pulibot er fyrst staðsettur í reit  $(0,0)$  með lit 0. Pulibot skoðar stöðufylkið  $S = [0, -2, 2, 2, -2]$ . Pulibot gæti verið með forrit sem ákveður að miðað við fylkið sem var gefið, þá breytir hann litinum á reitnum í  $Z = 1$  og færir sig síðan í austur, eins og sýnt er á miðju og hægri myndinni.



## Keppnisreglur

- Í byrjun er Pulibot settur í reit  $(0,0)$  og byrjar keyrslu á forritinu sínu.
- Pulibot má ekki hreyfa sig á reit sem er ekki tómur.
- Forritið hans Pulibot þarf að hætta keyrslu eftir í mesta lagi 500 000 skref.
- Eftir að forritið hættir keyrslu, ættu reitirnir í völundarhúsinu að vera litaðir þannig að:
  - Það er til stysta leið frá  $(0,0)$  yfir í  $(H-1, W-1)$  þannig að hver reitur í leiðinni hefur litinn 1.
  - Liturinn á öllum öðrum tómum reitum er 0.
- Pulibot má hætta keyrslu á hvaða tómum reit sem er.

Eftirfarandi mynd sýnir mögulegt völundarhús með  $H = W = 6$ . Byrjunarstaðan er sýnd á vinstri og ein af samþykktu litununum af reitum eftir að forritið hefur hætt keyrslu er sýnd hægra megin á myndinni.



## Útfærsluatriði

Þú skalt útfæra eftirfarandi fall.

```
void program_pulibot()
```

- Þetta fall á að skapa forritið hans Pulibot. Þetta forrit ætti að virka fyrir öll gildi á  $H$  og  $W$ , ásamt öllum völungarhúsum sem fylgja skorðum verkefnisins.
- Það er kallað á þetta fall nákvæmlega einu sinni fyrir sérhvert prufutilvik.

Þetta fall getur kallað á eftirfarandi fall til að hanna forritið hans Pulibot.

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

- $S$ : fylki af lengd 5 sem lýsir stöðufylkinu.
- $Z$ : heiltala, sem er ekki neikvæð, sem táknar lit.
- $A$ : einn stafur sem táknar aðgerðina sem Pulibot framkvæmir:
  - H: halda fyrir kyrru;
  - W: færa sig til vesturs;
  - S: færa sig til suðurs;
  - E: færa sig til austurs;
  - N: færa sig til norðurs;
  - T: hætta keyrslunni á forritinu.
- Að kalla á þetta fall, þýðir að þegar Pulibot fær stöðufylki  $S$ , þá á hann að framkvæma skipunina  $(Z, A)$ .

Ef kallað er á þetta fall oftari en einu sinni með sama stöðufylki  $S$ , þá mun lausnin þín fá Output `isn't correct`.

Það þarf ekki að kalla á `set_instruction` með öllum mögulegu stöðufylkjum  $S$ . Hinsvegar ef Pulibot fær inn stöðufylki sem ekki er til skipun fyrir, þá mun lausnin þín fá Output `isn't correct`.

Eftir að `program_pulibot` klárast, þá mun dómariinn keyra forritið hans Pulibot á eitt eða fleiri völungarhús. Þessar keyrslur teljast *ekki* í átt að tímamörkum lausnar þinnar. Dómariinn aðlagir sig *ekki*, það er, mengi völungarhúsa er skilgreint fyrirfram fyrir hvert prufutilvik.

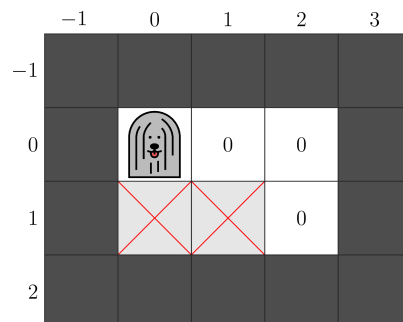
Ef Pulibot brýtur einhverja keppnisreglu, áður en hann hætir keyrslu, þá mun lausnin þín fá Output `isn't correct`.

## Sýnidæmi

Fallið `program_pulibot` má kalla á `set_instruction` á eftirfarandi máta:

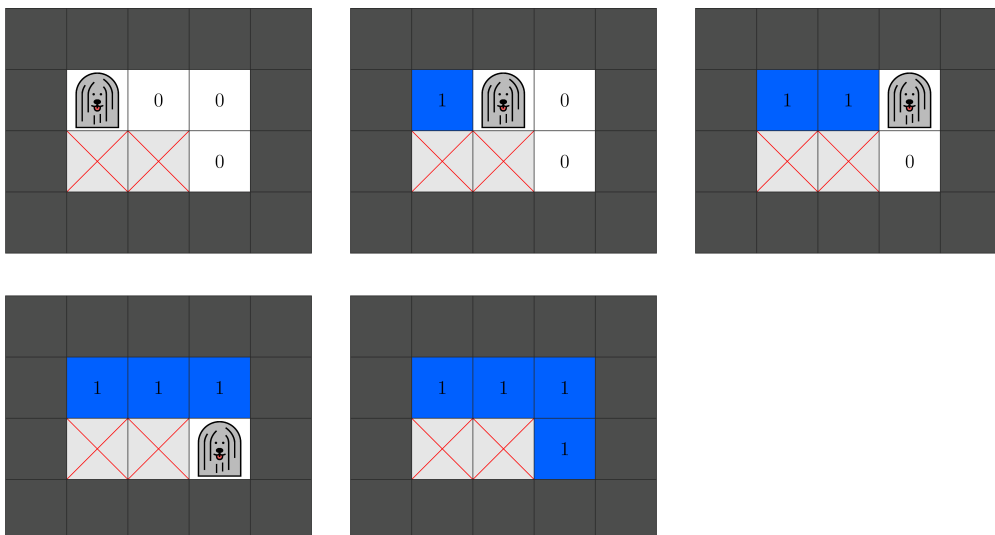
Kall	Skipun fyrir stöðufylkið $S$
<code>set_instruction([0, -2, -1, 0, -2], 1, E)</code>	Setja litinn í 1 og færa sig til austurs.
<code>set_instruction([0, 1, -1, 0, -2], 1, E)</code>	Setja litinn í 1 og færa sig til austurs
<code>set_instruction([0, 1, 0, -2, -2], 1, S)</code>	Setja litinn í 1 og færa sig til suðurs
<code>set_instruction([0, -1, -2, -2, 1], 1, T)</code>	Setja litinn í 1 og hætta keyrslu

Íhugið eftirfarandi atburðarrás þar sem  $H = 2$  og  $W = 3$ , þar sem völundarhúsið er sýnt á eftirfarandi mynd.



Fyrir þetta tiltæka völundarhús, þá mun forritið hans Pulibot keyra fjögur skref. Stöðufylkið sem Pulibot skoðar og skipanirnar sem hann framkvæmir samsvara nákvæmlega fjóru köllunum í `set_instruction` sem skrifuð eru að ofan í réttri röð. Síðasta skipunin hættir síðan keyrslu.

Eftirfarandi mynd sýnir stöðurnar í völundarhúsinu fyrir öll köll í fjögur skrefin og lokaniðurstöðuna eftir að keyrslu hefur verið hætt.



Hins vegar, takið eftir að þetta forrit af 4 skipunum mun ekki finna stystu leið í öðrum völundarhúsum. Þannig ef þessarri lausn er skilað inn mun hún fá `Output isn't correct`.

## Skorður

$Z_{MAX} = 19$ . Þannig að Pulibot getur notað liti 0 til 19, með báðum meðtöldum.

Fyrir sérhvert völungarhús sem notað er til að prófa Pulibot gildir:

- $2 \leq H, W \leq 15$
- Það er til leið frá reitnum  $(0, 0)$  yfir í reitinn  $(H - 1, W - 1)$ .

## Hlutverkefni

1. (6 stig) Það er engin hindrun í völungarhúsinu.
2. (10 stig)  $H = 2$
3. (18 stig) Það er nákvæmlega ein leið á milli sérhverra para af tómunum reitum.
4. (20 stig) Stysta leiðin frá reitnum  $(0, 0)$  í reitinn  $(H - 1, W - 1)$  hefur lengd  $H + W - 2$
5. (46 points) Engar frekari skorður.

Ef í einhverju prufutilviki munu köllin í `set_instruction` eða forritið hans Pulibot, ekki fylgja skorðunum sem lýst er í útfærsluatriðunum, þá mun lausnin þín fá 0 stig.

Í hverju prufutilviki getur þú fengið hlutstig með því að búa til litun sem er næstum því rétt.

Formlega:

- Lausnin á prufutilviki er **heil** ef lokalitunin af tómu reitunum fylgja keppnisreglunum.
- Lausnin á prufutilviki er **hlutleyst** ef lokalitunin er á eftirfarandi hátt:
  - Til er stysta leið frá  $(0, 0)$  yfir í  $(H - 1, W - 1)$  þannig að liturinn af hverjum reit í leiðinni er 1.
  - Það er engin annar tómur reitur með lit 1.
  - Einhver tómur reitur hefur lit annan en 0 eða 1.

Þú færð 0 stig ef lausnin þín við prufutilviki er hvorki heil né hlutleyst.

Í hlutverkefnum 1-4 færðu þú 100% stiga fyrir heila lausn eða 50% stiga fyrir hlutleysta lausn af samtals stigum hlutverkefnisins.

Í hlutverkefni 5, fara stigin þín eftir fjölda notaðra lita sem forritið hans Pulibot notar. Nánar tiltekið, skrifum  $Z^*$  sem hæsta gildi  $Z$  yfir öll köll í `set_instruction`. Stigin fyrir prufutilvikið eru reiknuð út frá eftirfarandi töflu:

Skilyrði	Stig (heil)	Stig (hlutleyst)
$11 \leq Z^* \leq 19$	$20 + (19 - Z^*)$	$12 + (19 - Z^*)$
$Z^* = 10$	31	23
$Z^* = 9$	34	26
$Z^* = 8$	38	29
$Z^* = 7$	42	32
$Z^* \leq 6$	46	36

Stigin fyrir hvert hlutverkefni er lægsta gildi stiga fyrir prufutilvikin í hlutverkefninu.

## Sýnisfirferðarforrit

Sýnisfirferðarforritið les inntakið á eftirfarandi sniði:

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

Hérna er  $m$  fylki af  $H$  fylkjum af  $W$  heiltölum, sem lýsa þeim reitum í vöndarhúsinu sem eru ekki jaðar reitir. Ef  $m[r][c] = 0$  þá er reitur  $(r, c)$  tómur og ef  $m[r][c] = 1$  þá er reitur  $(r, c)$  hindrunar reitur.

Sýnisfirferðarforritið kallar fyrst á `program_pulibot ( )`. Ef sýnisfirferðarforritið tekur eftir broti á reglum, þá skrifar það `Protocol Violation: <MSG>` og hættir keyrslu, þar sem `<MSG>` er eitt af eftirfarandi villuskilaboðum:

- `Invalid array`:  $-2 \leq S[i] \leq Z_{MAX}$  er ekki uppfyllt fyrir eitthvað  $i$  eða lengdin á  $S$  er ekki 5.
- `Invalid color`:  $0 \leq Z \leq Z_{MAX}$  er ekki uppfyllt.
- `Invalid action`: stafur  $A$  er ekki einn af H, W, S, E, N eða T.
- `Same state array`: `set_instruction` var kallað með sama stöðufylkinu  $S$  í minnsta lagi tvisvar.

Annars, þegar `program_pulibot` klárar keyrslu, mun sýnisfirferðarforritið keyra forritið hans `Pulibot` á vöndarhúsið sem lýst er í inntakinu.

Sýnisfirferðarforritið skilar frá sér tveimur útgildum.

Fyrst skráir sýnisfirferðarforritið aðgerðirnar sem `Pulibot` gerir í skjalið `robot . bin` í sömu möppu og forritið var keyrt í. Þessi skrá er inntakið fyrir myndræna hjálpartólið sem lýst er í næsta hluta.

Næst, ef forritið hans `Pulibot` hættir ekki keyrslu á leyfilegan hátt, þá mun sýnisfirferðarforritið skrifa eitt af eftirfarandi skilaboðum:

- `Unexpected state`: Pulibot skoðaði stöðufylki sem `set_instruction` hefur ekki verið kallað með.
- `Invalid move`: Hreyfing var framkvæmd, þannig að Pulibot endaði á ógildum reit.
- `Too many steps`: Pulibot framkvæmdi 500 000 skref, án þess að hætta keyrslu.

Annars, látum við  $e[r][c]$  vera stöðuna af reit  $(r, c)$  eftir að forritið hans Pulibot hættir keyrslu. Sýnisyfirferðarforritið skrifar  $H$  línur á eftirfarandi máta:

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

## Myndrænt hjálpartól

Í viðhengjum fyrir þetta verkefni er skjal sem heitir `display.py`. Þegar kallað er á Python forritið mun það sýna aðgerðinar sem Pulibot framkvæmdi í völungarhúsinu sem lýst var í inntakinu á sýnisyfirferðarforritinu. Til að þetta virki þarf skráin `robot.bin` að vera staðsett í núverandi möppu.

Til að kalla á forritið skal keyra.

```
python3 display.py
```

Einfalt myndrænt viðmót mun koma upp. Aðal atriðin eru eftirfarandi:

- Þú getur séð núverandi stöðu völungarhússins og reitinn sem Pulibot er á, merkt með fering.
- Þú getur flett á milli skrefa sem Pulibot tekur, með því að ýta á örvarnar á skjánum eða með því að nota lyklaborðið. Þú getur einnig hoppað á nákvæman stað.
- Næsta skref sem Pulibot mun taka er sýnt neðst. Það sýnir núverandi stöðufylki og skipunina sem mun vera framkvæmd. Eftir síðasta skrefið mun annaðhvort standa eitt af villuskilaboðunum eða `Terminated` ef forritið hætti keyrslu.
- Fyrir hverja tölu sem merkir lit getur þú gefið litrænan bakgrunn, ásamt texta. Textinn er stuttur strengur sem mun vera sýndur í hverjum reit af sama lit. Þú getur gefið litrænan bakgrunn eða texta á eftirfarandi hátt:
  - Skrifað hann í textagluggann sem kemur eftir að ýtt er á `Colors` takkan.
  - Eða breyta innihaldinu í `colors.txt` skránni
- Til að endurhlaða `robot.bin`, skal ýta á `Reload` takkann. Það er hentugt skyldi innihald `robot.bin` hafa breyst.