



Robotu sacensības

Mākslīgā intelekta pētnieki no Segedas Universitātes organizē robotu programmēšanas sacensības. Jūsu draugs Hanga izlēma piedalīties šajās sacensībās. Mērķis ir uzprogrammēt ideālu *Pulibotu*, godinot slavenas ungāru ganu suņu šķirnes Puli izcilās prāta spējas.

Pulibota darbība tiks testēta labirintā, kas sastāv no rūtiņu režģa ar izmēriem $(H + 2) \times (W + 2)$. Režģa rindas sanumurētas no -1 līdz H no ziemeļiem uz dienvidiem un režģa kolonnas ir sanumurētas no -1 līdz W no rietumiem līdz austrumiem. Rūtiņu, kas atrodas režģa rindā r un kolonnā c ($-1 \leq r \leq H$, $-1 \leq c \leq W$), saucim par rūtiņu (r, c) .

Aplūkosim rūtiņu (r, c) , kur $0 \leq r < H$ un $0 \leq c < W$. Tai ir 4 **blakusrūtiņas**:

- rūtiņa $(r, c - 1)$ tiek saukta par rūtiņas (r, c) **rietumu** rūtiņu;
- rūtiņa $(r + 1, c)$ tiek saukta par rūtiņas (r, c) **dienvidu** rūtiņu;
- rūtiņa $(r, c + 1)$ tiek saukta par rūtiņas (r, c) **austrumu** rūtiņu;
- rūtiņa $(r - 1, c)$ tiek saukta par rūtiņas (r, c) **ziemeļu** rūtiņu.

Teiksim, ka rūtiņa (r, c) ir režģa **robežas** rūtiņa, ja $r = -1$ vai $r = H$ vai $c = -1$ vai $c = W$. Katra rūtiņa, kas nav robežas rūtiņa ir vai nu **šķēršļa** rūtiņa, vai **tukša** rūtiņa. Papildus, katrai tukšai rūtiņai ir **krāsa**, ko apzīmē ar nenegatīvu veselu skaitli starp 0 un Z_{MAX} , ieskaitot. Sākotnēji, katras tukšas rūtiņas krāsa ir 0.

Piemēram, aplūkosim labirintu ar izmēriem $H = 4$ un $W = 5$, kas satur tikai vienu šķēršļa rūtiņu $(1, 3)$:

	-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							

Vienīgā robežas rūtiņa zīmējumā ir krustota. Labirinta šķēršļa rūtiņas ir tumšas. Skaitļi, kas ierakstīti visās tukšās rūtiņās, apzīmē to attiecīgās krāsas.

Par **ceļu** garumā ℓ ($\ell > 0$) no rūtiņas (r_0, c_0) uz rūtiņu (r_ℓ, c_ℓ) saucim virkni ar atšķirīgām *tukšām* rūtiņām $(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$, kur katram i ($0 \leq i < \ell$) rūtiņas (r_i, c_i) un (r_{i+1}, c_{i+1}) ir blakusrūtiņas.

Ievērojiet, ka ceļš garumā ℓ satur tieši $\ell + 1$ rūtiņas.

Sacensībās pētnieki izveido labirintu tādā veidā, ka starp rūtiņām $(0, 0)$ un $(H - 1, W - 1)$ pastāv vismaz viens ceļš. Ievērojiet, ka no šī tiek garantēts, ka rūtiņas $(0, 0)$ un $(H - 1, W - 1)$ ir tukšas.

Hanga nezina, kuras rūtiņas labirintā ir tukšas un kuras ir šķēršļa rūtiņas.

Jūsu uzdevums ir palīdzēt Hanga uzprogrammēt Pulibotu tā, lai tas būtu spējīgs atrast *īsāko ceļu* (tas ir, ceļu ar mazāko iespējamo garumu) no rūtiņas $(0, 0)$ uz rūtiņu $(H - 1, W - 1)$ nezināmā labirintā, ko izveidoja pētnieki. Pulibota apraksts un sacensību noteikumi ir izklāstīti zemāk.

Nemiet vērā, ka pēdējā šo uzdevuma nosacījumu sadaļa apraksta vizualizācijas rīku, ko jūs varat lietot, lai vizualizētu Pulibotu.

Pulibota apraksts

Definēsim rūtiņas (r, c) **stāvokli** visiem $-1 \leq r \leq H$ un $-1 \leq c \leq W$ kā veselu skaitli tādu, ka:

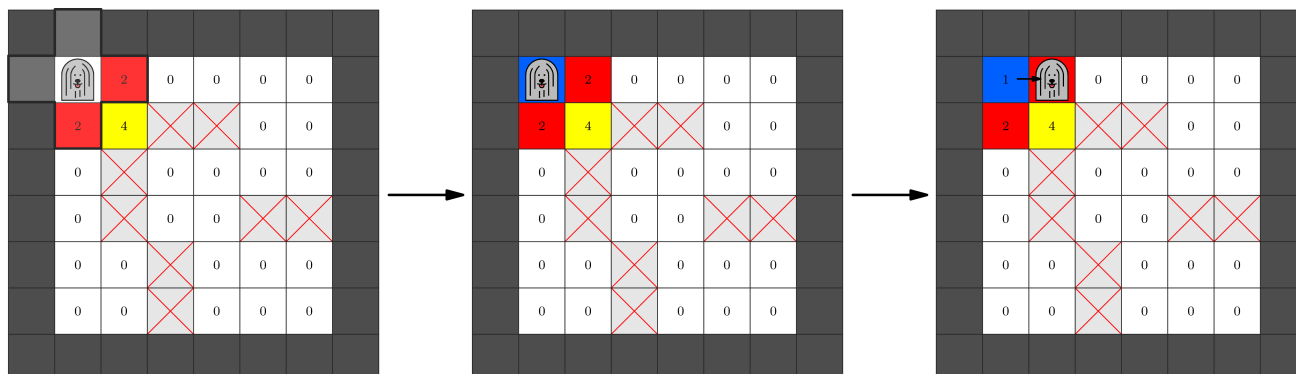
- ja rūtiņa (r, c) ir robežas rūtiņa, tad tās stāvoklis ir -2 ;
- ja rūtiņa (r, c) ir šķēršļa rūtiņa, tad tās stāvoklis ir -1 ;
- ja rūtiņa (r, c) ir tukša, tad tās stāvoklis ir šīs rūtiņas krāsa.

Pulibota programma tiek izpildīta kā soļu virkne. Katrā solī, Pulibots atpazīst blakusrūtiņu stāvokļus un tad izpilda vienu instrukciju. Šo instrukciju nosaka atpazītie stāvokļi. Precīzāks apraksts seko.

Pieņemsim, ka pašreizējā soļa sākumā Pulibots atrodas rūtiņā (r, c) , kas ir tukša. Solis tiek izpildīts šādi:

1. Sākumā, Pulibots atpazīst pašreizējo **stāvokļu masīvu**, kas ir masīvs $S = [S[0], S[1], S[2], S[3], S[4]]$, kas sastāv no rūtiņas (r, c) un blakusrūtiņu stāvokļiem:
 - $S[0]$ ir rūtiņas (r, c) stāvoklis.
 - $S[1]$ ir rietumu rūtiņas stāvoklis.
 - $S[2]$ ir dienvidu rūtiņas stāvoklis.
 - $S[3]$ ir austrumu rūtiņas stāvoklis.
 - $S[4]$ ir ziemeļu rūtiņas stāvoklis.
2. Tad Pulibots nosaka **instrukciju** (Z, A) , kas atbilst atpazītām stāvokļu masīvam.
3. Tad Pulibots izpilda šo instrukciju: tas uzstāda rūtiņas (r, c) krāsu uz Z un izpilda darbību A , kas ir viena no šādām darbībām:
 - *palikt* rūtiņā (r, c) ;
 - *pārvietoties* vienā no 4 blakusrūtiņām;
 - *izbeigt programmas darbību*.

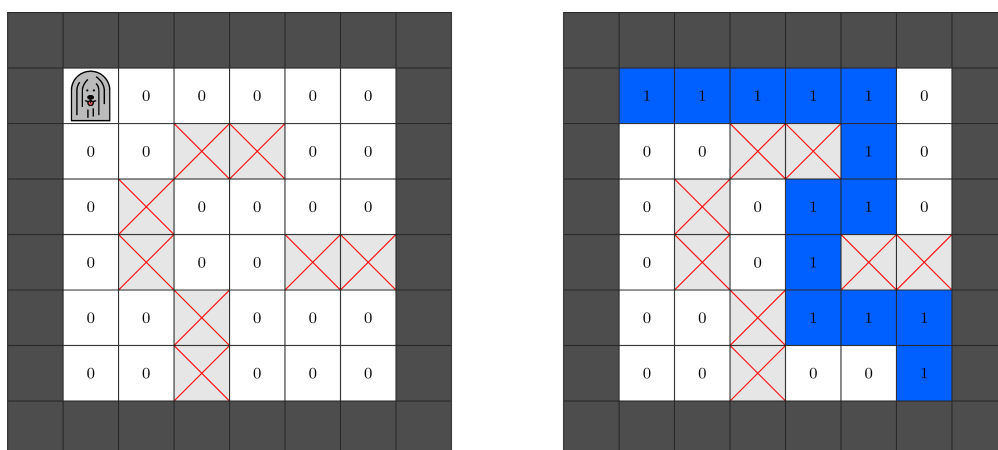
Piemēram, aplūkosim scenāriju, attēlotu pa kreisi zīmējumā zemāk. Pulibots atrodas rūtiņā $(0,0)$ ar krāsu 0. Pulibots atpazīst stāvokļu masīvu $S = [0, -2, 2, 2, -2]$. Pulibotam var būt programma, kas pēc masīva atpazīšanas uzstāda pašreizējās rūtiņas stāvokli uz $Z = 1$ un pārvietojas uz austrumiem, kā parādīts vidū un pa labi zīmējumā:



Robotu sacensību noteikumi

- Sākumā Pulibots tiek ievietots rūtiņā $(0,0)$ un sāk izpildīt savu programmu.
- Pulibotam nav atļauts pārvietoties uz rūtiņu, kas nav tukša.
- Pulibota programmai jāizbeidz darbība pēc ne vairāk kā 500 000 soļiem.
- Pēc Pulibota programmas pabeigšanas, tukšām rūtiņām labirintā jābūt nokrāsotām tā, ka:
 - Pastāv īsākais ceļš no $(0,0)$ uz $(H-1, W-1)$, kur katras ceļa rūtiņas krāsa ir 1.
 - Visām pārējām tukšām rūtiņām krāsa ir 0.
- Pulibots var izbeigt programmu, atrodoties jebkurā tukšā rūtiņā.

Piemēram, zīmējums zemāk attēlo vienu iespējamu labirintu ar izmēriem $H = W = 6$. Sākotnējā konfigurācija ir attēlota pa kreisi un viens derīgs tukšo rūtiņu krāsojums pēc programmas izbeigšanas ir attēlots pa labi:



Implementācijas detaļas

Jums ir jāimplementē šāda procedūra.

```
void program_pulibot()
```

- Šai procedūrai jāizveido Pulibota programmu. Šai programmai jāstrādā pareizi visām H un W vērtībām un jebkuram labirintam, kas atbilst uzdevuma nosacījumiem.
- Šī procedūra tiks izsaukta tieši vienu reizi katram testam.

Šī procedūra var veikt šādas procedūras izsaukumus, lai izveidotu Pulibota programmu:

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

- S : masīvs garumā 5, kas apraksta stāvokļu masīvu.
- Z : nenegatīvs vesels skaitlis, kas apzīmē krāsu.
- A : viens simbols, kas apraksta Pulibota veiktu darbību šādi:
 - H: palikt;
 - W: pārvietoties uz rietumiem;
 - S: pārvietoties uz dienvidiem;
 - E: pārvietoties uz austrumiem;
 - N: pārvietoties uz ziemeļiem;
 - T: izbeigt programmas darbību.
- Šīs procedūras izsaukšana nosaka, ka Pulibotam jāizpilda instrukcija (Z, A) pēc stāvokļa masīva S atpazīšanas.

Izsaucot šo procedūru vairākas reizes ar vienu un to pašu stāvokli S , jūsu risinājums iegūs verdiktu `Output isn't correct`.

Nav nepieciešams izsaukt `set_instruction` katram iespējamam stāvokļu masīvam S . Tomēr, ja Pulibots kādā brīdī atpazīst stāvokļu masīvu, kuram instrukcija netika uzstādīta, jūsu risinājums iegūs verdiktu `Output isn't correct`.

Pēc `program_pulibot` pabeigšanās, vērtētājprogramma darbina Pulibota programmu uz viena vai vairākiem labirintiem. Šie izsaukumi *netiek* pieskaitīti jūsu risinājuma darbības laikam. Vērtētājprogramma *nav* adaptīva, kas nozīmē, ka labirintu kopa ir fiksēta katram testam pirms risinājuma izpildes.

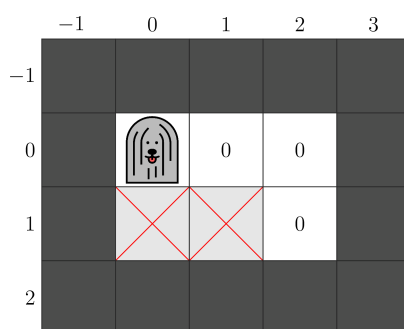
Ja Pulibots pārkāpj jebkuru no Robotu sacensību noteikumiem pirms tā programmas izbeigšanās, jūsu risinājums iegūs `Output isn't correct` verdiktu.

Piemērs

Procedūra `program_pulibot` var veikt `set_instruction` izsaukumus šādi:

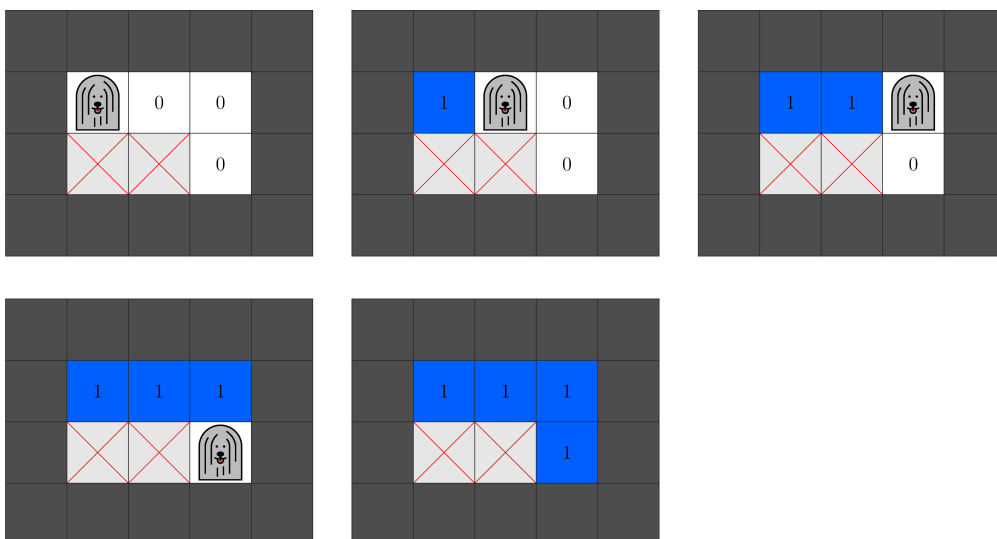
Izsaukums	Instrukcija stāvokļu masīvam S
<code>set_instruction([0, -2, -1, 0, -2], 1, E)</code>	Uzstādīt krāsu uz 1 un pārvietoties uz austrumiem
<code>set_instruction([0, 1, -1, 0, -2], 1, E)</code>	Uzstādīt krāsu uz 1 un pārvietoties uz austrumiem
<code>set_instruction([0, 1, 0, -2, -2], 1, S)</code>	Uzstādīt krāsu uz 1 un pārvietoties uz dienvidiem
<code>set_instruction([0, -1, -2, -2, 1], 1, T)</code>	Uzstādīt krāsu uz 1 un izbeigt programmu

Aplūkosim scenāriju, kur $H = 2$ un $W = 3$, un labirints ir attēlots šajā zīmējumā.



Šajā konkrētajā labirintā Pulibota programma kopumā veic četrus soļus. Stāvokļu masīvi, ko Pulibots atpazīst un instrukcijas, ko tas attiecīgi izpilda, precīzi atbilst četriem `set_instruction` izsaukumiem, kas veikti augšā, šādā secībā. Pēdējā no šīm instrukcijām izbeidz programmas darbību.

Bilde zemāk attēlo labirintu pirms katra no četriem soļiem un beigu krāsojumu pēc programmas izbeigšanas.



Nemiet tomēr vērā, ka šī programma no 4 instrukcijām var neatrast īsāko ceļu citos derīgos labirintos. Tāpēc šāds risinājums pie iesūtīšanas iegūtu Output `isn't correct` verdiktu.

Ierobežojumi

$Z_{MAX} = 19$. Līdz ar to Pulibots var izmantot krāsas no 0 līdz 19, ieskaitot.

Katram labirintam, ar ko pārbauda Pulibota darbību, izpildās:

- $2 \leq H, W \leq 15$
- Pastāv vismaz viens ceļš no rūtiņas $(0, 0)$ uz rūtiņu $(H - 1, W - 1)$.

Apakšuzdevumi

1. (6 punkti) Labirintā nav šķēršļa rūtiņu.
2. (10 punkti) $H = 2$.
3. (18 punkti) Pastāv tieši viens ceļš starp katru tukšo rūtiņu pāri.
4. (20 punkti) Katra īsākā ceļa no rūtiņas $(0, 0)$ uz rūtiņu $(H - 1, W - 1)$ garums ir vienāds ar $H + W - 2$.
5. (46 punkti) Bez papildus ierobežojumiem.

Ja kādā no testiem procedūras `set_instruction` izsaukumi vai Pulibota programma tās darbībā neatbilst nosacījumiem, kas aprakstīti Implementācijas detaļās, jūsu risinājuma vērtējums šajā apakšuzdevumā būs vienāds ar 0.

Katrā apakšuzdevumā jūs varat iegūt daļēju vērtējumu, ja izveidotais krāsojums ir gandrīz pareizs.

Formāli:

- Testa risinājums ir **pareizs**, ja beigu krāsojums atbilst Robotu sacensību noteikumiem.
- Testa risinājums ir **daļēji pareizs**, ja beigu krāsojums izskatās šādi:
 - Pastāv īsākais ceļš no $(0, 0)$ uz $(H - 1, W - 1)$, kur katras ceļa rūtiņas krāsa ir 1.
 - Nav citas tukšas rūtiņas režģī ar krāsu 1.
 - Kādas citas tukšas režģa rūtiņas krāsa atšķiras no 0 un 1.

Ja jūsu testa risinājums nav ne pareizs, ne daļēji pareizs, vērtējums šim testam būs 0.

1.-4. apakšuzdevumos pilnā risinājuma vērtējums ir 100% un daļējā risinājuma vērtējums ir 50% no attiecīgā apakšuzdevuma punktu skaita.

5. apakšuzdevumā jūsu vērtējums ir atkarīgs no krāsu skaita, kas ir izmantots Pulibota programmā. Precīzāk, apzīmēsim ar Z^* lielāko Z vērtību pāri visiem `set_instruction` izsaukumiem. Testa vērtējums tiek aprēķināts atbilstoši tabulai zemāk:

Nosacījums	Vērtējums (pilns)	Vērtējums (daļējs)
$11 \leq Z^* \leq 19$	$20 + (19 - Z^*)$	$12 + (19 - Z^*)$
$Z^* = 10$	31	23
$Z^* = 9$	34	26
$Z^* = 8$	38	29
$Z^* = 7$	42	32
$Z^* \leq 6$	46	36

Vērtējums katram apakšuzdevumam ir mazākais no punktu skaitiem starp apakšuzdevuma testiem.

Piemēra vērtētājprogramma

Piemēra vērtētājprogramma nolasa ievadu šādā formātā:

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

Šeit, m ir masīvs ar H masīviem ar W veseliem skaitļiem, kas apraksta visas labirinta rūtiņas, kas nav uz robežas. $m[r][c] = 0$, ja rūtiņa (r, c) ir tukša un $m[r][c] = 1$, ja rūtiņa (r, c) ir šķēršļa rūtiņa.

Piemēra vērtētājprogramma sākumā izsauc `program_pulibot()`. Ja piemēra vērtētājprogramma nosaka protokola pārkāpumu, tā izdrukā `Protocol Violation: <MSG>` un izbeidz darbību, kur `<MSG>` ir viens no šādiem kļūdas ziņojumiem:

- Invalid array: $-2 \leq S[i] \leq Z_{MAX}$ neizpildās kādam i vai S garums nav 5.
- Invalid color: $0 \leq Z \leq Z_{MAX}$ neizpildās.
- Invalid action: simbols A nav viens no H, W, S, E, N vai T.
- Same state array: `set_instruction` tika izsaukta ar vienu un to pašu masīvu S vismaz divreiz.

Citādi, kad `program_pulibot` izbeidz darbību, piemēra vērtētājprogramma izpilda Pulibota programmu labirintā, ko apraksta ievads.

Piemēra vērtētājprogrammai ir divi izvadi.

Pirmkārt, piemēra vērtētājprogramma ieraksta žurnālu ar Pulibota darbībam darba katalogā failā `robot.bin`. Šis fails kalpo kā ievads vizualizācijas rīkam, kas aprakstīts nākamajā nodaļā.

Otrkārt, ja Pulibota programma neizbeidz darbību veiksmīgi, piemēra vērtētājprogramma izdrukā vienu no šādiem kļūdas ziņojumiem:

- Unexpected state: Pulibots atpazīna stāvokļu masīvu, ar kuru netika izsaukta `set_instruction`.
- Invalid move: veicot darbību, Pulibots pārvietojās netukšā rūtiņā.
- Too many steps: Pulibots veica 500 000 soļus bez savas programmas izbeigšanas.

Citādi, apzīmēsim ar $e[r][c]$ rūtiņas (r, c) stāvokli pēc Pulibota programmas izbeigšanās. Piemēra vērtētājprogramma izdrukā H rindas šādā formātā:

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

Vizualizācijas rīks

Šī uzdevuma failu arhīvs satur failu ar nosaukumu `display.py`. Pie izsaukšanas šis Python skripts attēlo Pulibota darbības labirintā, ko apraksta piemēra vērtētājprogrammas ievads. Priekš tā darba katalogā jābūt bināram failam `robot.bin`.

Lai palaistu skriptu, izpildiet šādu komandu.

```
python3 display.py
```

Parādās vienkārša grafiska saskarne. Tās pamatīpašības ir šādas:

- Jūs varat aplūkot pilna labirinta stāvokli. Pašreizējā Pulibota atrašanās vieta ir izcelta ar taisnstūri.
- Jūs varat pārlūkot Pulibota soļus, klikšķinot bultiņu pogas vai spiežot to īsinājumtaustiņus. Jūs varat arī lekt uz specifisko soli.
- Nākamais solis Pulibota programmā ir attēlots apakšā. Tur ir parādīts pašreizējais stāvokļu masīvs un instrukcija, ko tas veiks. Pēc pēdējā soļa tur tiek parādīts vai nu piemēra vērtētājprogrammas kļūdas ziņojums, vai nu Terminated, ja programma pabeidz savu darbību veiksmīgi.
- Katram skaitlim, kas attēlo krāsu, jūs varat piekārtot vizuālu fona krāsu, kā arī parādīto tekstu. Parādītais teksts ir īsa virkne, kas tiks ierakstīta katrā rūtiņā ar šādu krāsu. Jūs varat piešķirt fona krāsas un parādītos tekstus vienā no šādiem veidiem:
 - Iestātīt tos dialoglodziņā pēc pogas `Colors` noklikšķināšanas.
 - Izmainīt faila `colors.txt` saturu.
- Lai pārlādētu `robot.bin`, lietojiet pogu Reload. Tas ir noderīgs, ja `robot.bin` saturs izmainījās.