

Robotų varžybos

Segedo universiteto mokslininkai, tyrinėjantys dirbtinį intelektą, organizuoja robotų programavimo varžybas. Jūsų draugas Hanga nusprendė jose sudalyvauti. Varžybų tikslas – suprogramuoti *Pulibotą*, kuris pavadintas norint pažymėti žinomos Vengriškos šunų veislės – pulių – protingumą.

Pulibotas bus testuojamas labirinte, sudarytame iš $(H + 2) \times (W + 2)$ langelių lentelės. Eilutės numeruojamos nuo -1 iki H iš šiaurės į pietus, o stulpeliai numeruojami nuo -1 iki W iš vakarų į rytus. Labirinto langelį, esantį r eilutėje ir c stulpelyje (kur $-1 \leq r \leq H$, $-1 \leq c \leq W$) žymėsime (r, c) .

Panagrinėkime langelį (r, c) , kur $0 \leq r < H$ ir $0 \leq c < W$. Langeliui (r, c) iš viso yra 4 **gretimi** langeliai:

- langelis $(r, c - 1)$ yra į **vakarų** nuo langelio (r, c) ;
- langelis $(r + 1, c)$ yra į **pietus** nuo langelio (r, c) ;
- langelis $(r, c + 1)$ yra į **rytus** nuo langelio (r, c) ;
- langelis $(r - 1, c)$ yra į **šiaurę** nuo langelio (r, c) .

Langelis (r, c) vadinamas **kraštiniu** labirinto langeliu, jei $r = -1$, arba $r = H$, arba $c = -1$, arba $c = W$. Kiekviename labirinto langelyje, kuris nėra kraštinis, yra arba **kliūtis**, arba langelis yra **tuščias**. Kiekvienas tuščias langelis turi **spalvą**, kuri žymima neneigiamu sveikuoju skaičiumi nuo 0 iki Z_{MAX} imtinai. Pradiniu momentu kiekvieno tuščio langelio spalva yra 0.

Pavyzdžiui, panagrinėkime labirintą, kuriame $H = 4$, $W = 5$ ir jame yra vienintelė kliūtis langelyje $(1, 3)$:

| | -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 | | | | | | | |

Paveikslėlyje vienintelė kliūtis pažymėta kryžiuoku, kraštiniai langeliai patamsinti, o likusiuose langeliuose įrašyti skaičiai reiškia tų langelių spalvas.

ℓ ($\ell > 0$) ilgio **keliu** iš (r_0, c_0) į (r_ℓ, c_ℓ) vadinsime poromis skirtingų *tuščių* langelių seką $(r_0, c_0), (r_1, c_1), \dots, (r_\ell, c_\ell)$, kur kiekvienam i ($0 \leq i < \ell$) langeliai (r_i, c_i) ir (r_{i+1}, c_{i+1}) yra gretimi.

Atkreipkite dėmesį, kad ℓ ilgio kelią sudaro lygiai $\ell + 1$ langelių.

Varžybų organizatoriai suprojektuoja tokį labirintą, kuriame egzistuoja bent vienas kelias iš $(0, 0)$ į $(H - 1, W - 1)$. Tai reiškia, kad langeliai $(0, 0)$ ir $(H - 1, W - 1)$ būtinai yra tušti.

Hanga nežino, kurie labirinto langeliai tušti, o kuriuose yra kliūtys.

Padėkite Hangai suprogramuoti Pulibotą taip, kad jis sugebėtų rasti *trumpiausią kelią* (t. y. kelią su mažiausiu įmanomu ilgiu) iš $(0, 0)$ į $(H - 1, W - 1)$ nežinomame labirinte, kurį sukonstravo mokslininkai. Žemiau pateikia Puliboto specifikacija ir varžybų taisyklės.

Atkreipkite dėmesį, kad paskutiniame skyrelyje aprašytas vizualizavimo įrankis, su kuriuo galite vizualizuoti Puliboto veiksmus.

Puliboto specifikacija

Kiekvienam $-1 \leq r \leq H$ ir $-1 \leq c \leq W$ langelio (r, c) **būsena** yra sveikasis skaičius, kuris apibrėžiamas taip:

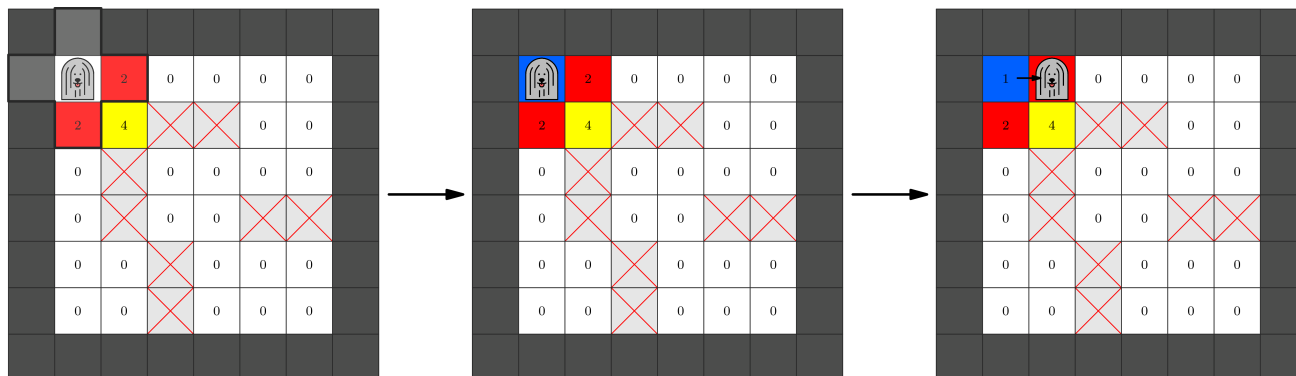
- jei langelis (r, c) yra kraštinis, tuomet jo būseną yra -2 ;
- jei langelis (r, c) yra kliūtis, tuomet jo būseną yra -1 ;
- jei langelis (r, c) yra tuščias, tuomet jo būseną lygi langelio spalvai.

Puliboto programa yra žingsnių seka. Kiekvieno žingsnio metu Pulibotas surenka gretimų langelių būsenas ir įvykdo komandą, kurią nulemia surinktos būsenos. Žemiau pateiktas tikslus aprašymas.

Tarkime, kad pradiniu einamojo žingsnio momentu Pulibotas yra tuščiaame langelyje (r, c) . Žingsnis vykdomas taip:

1. Pirmiausia, Pulibotas surenka dabartinį **būsenų masyvą**, tai yra masyvą $S = [S[0], S[1], S[2], S[3], S[4]]$, sudarytą iš langelio (r, c) ir visų jam gretimų langelių būsenų:
 - $S[0]$ yra langelio (r, c) būseną.
 - $S[1]$ yra į vakarus esančio gretimo langelio būseną.
 - $S[2]$ yra į pietus esančio gretimo langelio būseną.
 - $S[3]$ yra į rytus esančio gretimo langelio būseną.
 - $S[4]$ yra į šiaurę esančio gretimo langelio būseną.
2. Tuomet Pulibotas parenka **komandą** (Z, A) , atitinkančią surinktų būsenų masyvą.
3. Galiausiai Pulibotas įvykdo tą komandą: pakeičia langelio (r, c) spalvą į spalvą Z ir įvykdo veiksmą A , kuris yra vienas iš žemiau pateiktų veiksmų:
 - *pasilieka* langelyje (r, c) ;
 - *pajuda* į vieną iš 4 gretimų langelių;
 - *užbaigia programos vykdymą*.

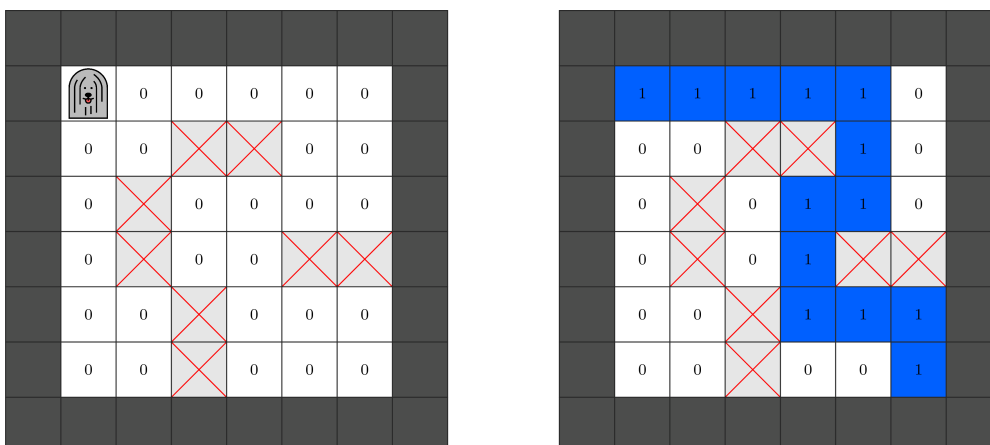
Panagrinėkime scenarijų, pateiktą kairiajame paveikslėlyje žemiau. Šiuo metu Pulibotas yra langelyje $(0,0)$, kurio spalva yra 0. Pulibotas surenka būsenas į būsenų masyvą $S = [0, -2, 2, 2, -2]$. Pulibotas gali vykdyti programą, kuri surinkusi būsenas pakeičia dabartinio langelio spalvą į $Z = 1$ ir pajuda į rytus, kaip pavaizduota viduriniame ir dešiniajame paveikslėliuose:



Robotų varžybų taisyklės

- Pradiniu momentu Pulibotas padedamas į $(0,0)$ langelį ir pradeda vykdyti savo programą.
- Pulibotui neleidžiama judėti į netuščią langelį.
- Puliboto programa privalo baigti vykdymą po ne daugiau kaip 500 000 žingsnių.
- Puliboto programai baigus darbą, tušti langeliai labirinte turi būti nuspalvinti taip, kad:
 - Egzistuoja trumpiausias kelias iš $(0,0)$ į $(H-1, W-1)$, kurio kiekvieno langelio spalva yra 1.
 - Visų kitų tuščių langelių spalva yra 0.
- Pulibotas gali baigti programos vykdymą bet kuriame tuščiame langelyje.

Pavyzdžiui, žemiau pateiktame paveikslėlyje yra labirintas, kur $H = W = 6$. Pradinė labirinto konfigūracija pateikta kairėje, o tinkamas tuščių langelių nuspalvinimas užbaigus programos vykdymą yra dešinėje:



Realizacija

Parašykite šią funkciją:

```
void program_pulibot()
```

- Ši funkcija turi sugeneruoti Poliboto programą. Programa turi teisingai veikti su visomis H ir W reikšmėmis ir bet kuriuo labirintu, atitinkančiu užduoties ribojimus.
- Ši funkcija kiekvienam testui išskviečiama lygiai vieną kartą.

Kad sugeneruotų Puliboto programą, ši funkcija gali kviesti funkciją:

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

- S : būsenų masyvas, kurio ilgis 5.
- Z : neneigiamas sveikasis skaičius, reiškiantis spalvą.
- A : vienas iš žemiau pateiktų simbolių, nusakantis Poliboto veiksmą
 - H: likti langelyje;
 - W: pajudėti į vakarus;
 - S: pajudėti į pietus;
 - E: pajudėti į rytus;
 - N: pajudėti į šiaurę;
 - T: užbaigti programos vykdymą.
- Šios funkcijos iškvietimas reiškia, kad Polibotas atpažinęs būseną S turi įvykdyti komandą (Z, A) .

Jei funkcija bus iškviesta keletą kartų su tuo pačiu būsenų masyvu S , bus pateiktas klaidos pranešimas `Output isn't correct`.

Nereikalaujama iškviesti `set_instruction` su kiekviena galima būsenų masyvo S reikšme. Tačiau jei Pulibotas vėliau surinks būsenų masyvą, kuriam nebus nurodyta komanda, bus pateiktas klaidos pranešimas `Output isn't correct`.

Funkcijai `program_pulibot` baigus darbą vertinimo programa kviečia Puliboto programą su vienu ar keliais labirintais. Šių iškvietimų vykdymo laikas *nėra* įskaičiuojamas į jūsų sprendimo vykdymo laiką. Vertinimo programa *nėra* adaptuvi, tai yra labirintų aibė kiekvienam testui yra sukurta iš anksto.

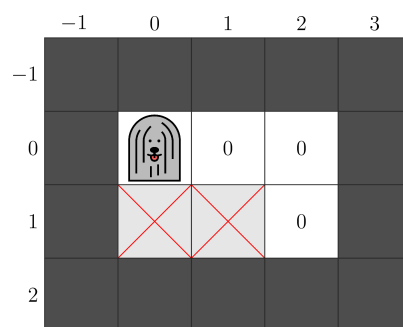
Jei Pulibotas pažeidžia kurią nors Robotų varžybų taisyklę prieš baigdamas programą, pateikiamas pranešimas `Output isn't correct`.

Pavyzdys

Funkcija `program_pulibot` gali atlikti šiuos `set_instruction` iškvietimus:

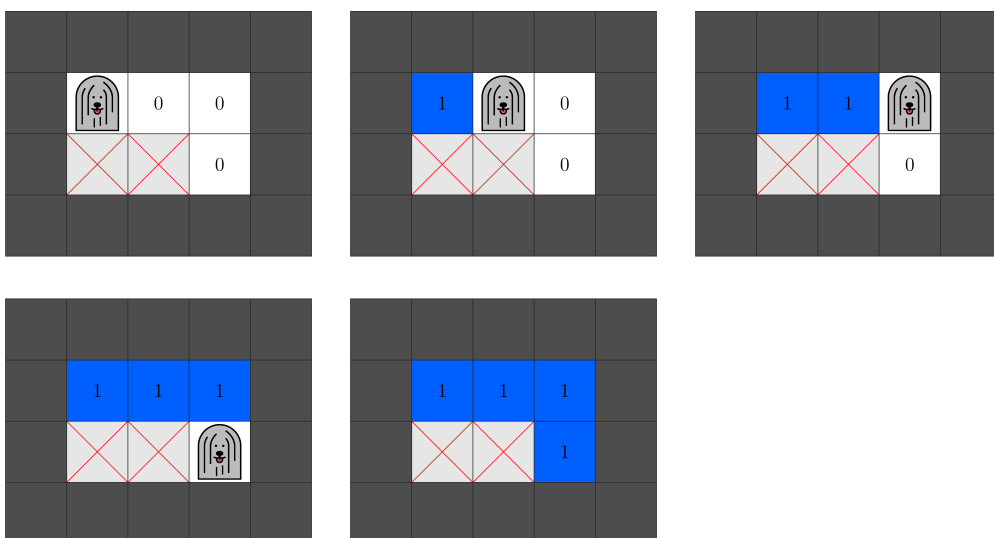
| Iškvietimas | Komanda būsenų masyvui S |
|--|---|
| <code>set_instruction([0, -2, -1, 0, -2], 1, E)</code> | Pakeisti spalvą į 1 ir pajudėti į rytus |
| <code>set_instruction([0, 1, -1, 0, -2], 1, E)</code> | Pakeisti spalvą į 1 ir pajudėti į rytus |
| <code>set_instruction([0, 1, 0, -2, -2], 1, S)</code> | Pakeisti spalvą į 1 ir pajudėti į pietus |
| <code>set_instruction([0, -1, -2, -2, 1], 1, T)</code> | Pakeisti spalvą į 1 ir sustabdyti programos vykdymą |

Panagrinėkime scenarijų, kur $H = 2$, $W = 3$, o labirintas pavaizduotas paveikslėlyje žemiau.



Šiam konkrečiam labirintui Poliboto programa įvykdys keturis žingsnius. Poliboto surenkami būsenų masyvai ir vykdomos komandos atitinka keturis aukščiau pateiktus `set_instruction` iškvietimus lentelėje nurodyta tvarka. Paskutinioji iš komandų sustabdo programą.

Toliau pateiktame pasveikslėlyje parodytas labirintas prieš vykdant kiekvieną iš keturių žingsnių ir langelių spalvos sustabdžius programą.



Tačiau atkreipkite dėmesį, kad ši 4 komandų programa nebūtinai ras trumpiausią kelią kituose galimuose labirintuose. Taigi ją pateikus bus gaunamas Output `isn't correct` pranešimas.

Ribojimai

$Z_{MAX} = 19$. Taigi, Pulibotas gali naudoti spalvas nuo 0 iki 19 imtinai.

Kiekvienam labirintui, su kuriuo testuojamas Pulibotas:

- $2 \leq H, W \leq 15$
- Egzistuoja bent vienas kelias iš $(0, 0)$ į $(H - 1, W - 1)$.

Dalinės užduotys

1. (6 taškai) Labirinte kliūčių nėra.
2. (10 taškų) $H = 2$
3. (18 taškų) Yra lygiai vienas kelias tarp bet kurios tuščių langelių poros.
4. (20 taškų) Kiekvieno trumpiausio kelio iš $(0, 0)$ į $(H - 1, W - 1)$ ilgis yra $H + W - 2$.
5. (46 taškai) Papildomų ribojimų nėra.

Jei bet kurio testo funkcijos `set_instruction` iškvietimai arba Puliboto programa vykdymo metu neatitinka Realizacijos skyrelyje pateiktų reikalavimų, šios dalinės užduoties įvertinimas bus 0 taškų.

Už kiekvieną dalinę užduotį galima gauti nepilną įvertinimą, jei labirintas nuspalvinamas beveik teisingai.

Tai yra:

- Testo atsakymas yra **pilnas**, jei tušti labirinto langeliai nuspalvinti pagal Robotų varžybų taisykles.
- Testo atsakymas yra **nepilnas**, jei labirinto langeliai nuspalvinti taip:
 - Trumpiausio kelio iš $(0, 0)$ į $(H - 1, W - 1)$ langelių spalva yra 1.
 - Daugiau nei vieno tuščio langelio spalva nelygi 1.
 - Kai kurių tuščių langelių spalva nelygi nei 0, nei 1.

Jei jūsų sprendimo pateiktas testo atsakymas nėra nei pilnas, nei nepilnas, šio testo įvertinimas bus 0 taškų.

Už pilnus dalinių užduočių nr. 1-4 atsakymus bus skiriama 100% taškų, o už nepilnus – 50% taškų.

5-oje dalinėje užduotyje skiriamų taškų skaičius priklausys nuo Puliboto programos naudojamų spalvų kiekio. Tai yra, pažymėkime Z^* didžiausią Z reikšmę per visus `set_instruction` iškvietimus. Už testą skiriami taškai apskaičiuojami pagal žemiau pateiktą lentelę:

| Z^* reikšmė | Taškai (pilnas atsakymas) | Taškai (nepilnas atsakymas) |
|-----------------------|---------------------------|-----------------------------|
| $11 \leq Z^* \leq 19$ | $20 + (19 - Z^*)$ | $12 + (19 - Z^*)$ |
| $Z^* = 10$ | 31 | 23 |
| $Z^* = 9$ | 34 | 26 |
| $Z^* = 8$ | 38 | 29 |
| $Z^* = 7$ | 42 | 32 |
| $Z^* \leq 6$ | 46 | 36 |

Už kiekvieną dalinę užduotį skiriamų taškų skaičius lygus mažiausiam gautam įvertinimui už šios dalinės užduoties testus.

Pavyzdinė vertinimo programa

Pavyzdinė vertinimo programa skaito duomenis tokiu formatu:

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

Čia m yra H dydžio masyvas, kurį sudaro W dydžio sveikųjų skaičių masyvai, aprašantys ne kraštinius labirinto langelius. $m[r][c] = 0$, jei (r, c) yra tuščias, ir $m[r][c] = 1$, jei langelyje (r, c) yra kliūtis.

Pavyzdinė vertinimo programa pirmiausia iškviečia `program_pulibot()`. Jei pavyzdinė vertinimo programa nustato, kad nesilaikoma protokolo, ji išveda pranešimą `Protocol Violation: <MSG>` ir baigia darbą. Čia `<MSG>` yra vienas šių klaidos pranešimų:

- `Invalid array`: Nesilaikoma ribojimo $-2 \leq S[i] \leq Z_{MAX}$ kuriam nors i arba S ilgis nelygus 5.
- `Invalid color`: nesilaikoma ribojimo $0 \leq Z \leq Z_{MAX}$.
- `Invalid action`: simbolis A nėra vienas šių simbolių H, W, S, E, N ar T.
- `Same state array`: `set_instruction` bent du kartus iškviesta su tuo pačiu masyvu S .

Kitu atveju, kai `program_pulibot` baigia darbą, pavyzdinė vertinimo programa įvykdo Puliboto programą labirinte, kurį nusako pradiniai duomenys.

Pavyzdinė vertinimo programa pateikia du rezultatus.

Pirma, ji į failą `robot.bin` (einamajame kataloge) įrašo Puliboto veiksmų sąrašą. Šis failas naudojamas vizualizacijai, aprašytai tolimesniame skyrelyje.

Antra, jei Puliboto programa nebaigia darbo sėkmingai, pavyzdinė vertinimo programa išveda vieną iš šių klaidos pranešimų:

- Unexpected state: Pulibotas surinko būsenų masyvą, su kuriuo nebuvo iškviesta `set_instruction`.
- Invalid move: vykdydamas komandą Pulibotas pajudėjo į netuščią langelį.
- Too many steps: Pulibotas įvykdė 500 000 žingsnių nebaigdamas programos.

Priešingu atveju, tegul $e[r][c]$ yra langelio (r, c) būsena Poliboto programai baigus vykdymą. Pavyzdinė vertinimo programa išveda H eilučių tokiu formatu:

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

Vizualizavimo įrankis

Šio uždavinio pakete yra failas `display.py`. Paleistas šis Pitono skriptas pavaizduoja Puliboto veiksmus labirinte, aprašytame pradinioose pavyzdinės vertinimo programos duomenyse. Tam reikia, kad dvejetainis failas `robot.bin` būtų pasiekiamas einamajame kataloge.

Norėdami paleisti skriptą įvykdysite šią komandą:

```
python3 display.py
```

Tuomet atsiras paprasta grafinė sąsaja:

- Galėsite matyti viso labirinto būseną. Stačiakampiu pažymėta dabartinė Puliboto vieta.
- Galite simuliuoti Puliboto žingsnius su rodyklių klavišais ar juos atitinkančiais sparčiais klavišais. Taip pat galite nušukti į konkretų žingsnį.
- Apačioje rodomas tolimesnis Puliboto programos žingsnis. Rodomas dabartinis būsenų masyvas ir komanda, kuri bus vykdoma. Atlikus paskutinį žingsnį rodomas arba vienas iš pavyzdinės vertinimo programos klaidų pranešimų, arba pranešimas Terminated, jei programa sėkmingai baigia darbą.
- Kiekvienam spalvą žyminčiam skaičiui galite priskirti spalvą (foną) bei rodomą tekstą. Šis tekstas yra trumpa eilutė, kuri įrašoma visuose tos pačios spalvos langeliuose. Fono spalvą ar rodomą tekstą galite nustatyti šiais būdais:
 - Spausdami Colors mygtuką ir nustatydami juos atsidariusiame dialogo lange.
 - Redaguodami failą `colors.txt`.
- Norėdami iš naujo įkelti `robot.bin`, spustelėkite Reload mygtuką. To gali prireikti, jei pasikeitė failas `robot.bin`.