Súťaž robotov

Odborníci na umelú inteligenciu na Szegedskej univerzite si pripravili súťaž v programovaní robotov. Presnejšie, súťažiaci budú programovať *Pulibota:* robota vyzerajúceho ako veľmi inteligentné maďarské psie plemeno Puli. Tvoja kamarátka Hanka sa rozhodla, že sa do tejto súťaže zapojí.

Pulibot bude testovaný v bludisku, ktoré organizátori postavia na mriežke tvorenej $(H+2) \times (W+2)$ štvorcovými políčkami. Riadky tejto mriežky majú čísla od -1 po H zo severu na juh, stĺpce majú čísla -1 až W zo západu na východ. Políčko v riadku r a stĺpci c budeme označovať (r, c).

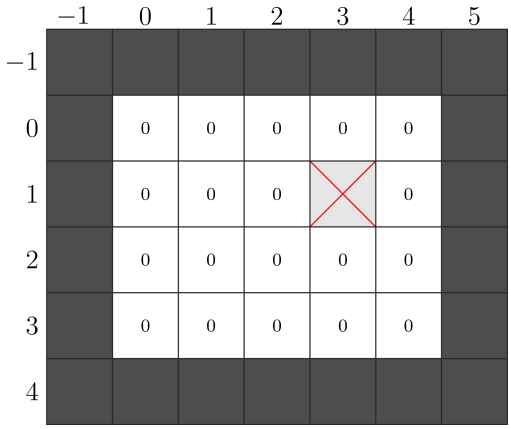
Políčko (r, c) je **obvodové**, ak platí aspoň jedna z podmienok $r = -1$, $r = H$, $c = -1$ a $c = W$. Ostatné políčka, teda tie, pre ktoré platí $0 \leq r < H$ a $0 \leq c < W$, sú **vnútorné**.

Vnútorné políčko (r, c) má práve 4 *susedov:*

- políčko $(r, c-1)$ je susedom *na západ* od políčka (r, c);
- políčko $(r+1, c)$ je susedom *na juh* od políčka (r, c);
- políčko $(r, c+1)$ je susedom *na východ* od políčka (r, c);
- políčko $(r-1, c)$ je susedom *na sever* od políčka (r, c).

Každé vnútorné políčko je buď **prázdne**, alebo obsahuje **prekážku**. Navyše platí, že každé *prázdne* políčko má nejakú **farbu**. Farby sú celé čísla od 0 po Z_{MAX}, vrátane. Na začiatku majú všetky prázdne políčka farbu 0.

Uvažujme napríklad bludisko v mriežke, pre ktorú platí $H = 4$ a $W = 5$. V mriežke máme jednu prekážku: na políčku $(1, 3)$.



Na obrázku je prekážka označená krížikom a obvodové políčka sú vyplnené šedou. Čísla v prázdnych vnútorných políčkach predstavujú ich farby.

Cesta dĺžky ℓ ($\ell > 0$) z bunky (r_0, c_0) do bunky (r_ℓ, c_ℓ) je postupnosť navzájom rôznych **vnútorných prázdnych** políčok $(r_0, c_0), (r_1, c_1), \ldots, (r_\ell, c_\ell)$ v ktorej každé dve po sebe idúce políčka susedia.

Všimni si, že cestu dĺžky ℓ tvorí $\ell + 1$ políčok.

Na súťaž si organizátori pripravili mriežku s bludiskom z prekážok. V tomto bludisku existuje aspoň jedna cesta z políčka $(0, 0)$ na políčko $(H - 1, W - 1)$. Hanka vopred nevie, ktoré políčka sú prázdne a ktoré obsahujú prekážky.

Hankinou úlohou je naprogramovať Pulibota tak, aby zaručene našiel *najkratšiu cestu* z políčka $(0, 0)$ na políčko $(H - 1, W - 1)$ -- bez ohľadu na to, ako neznáme bludisko vyzerá. Pomôž jej s touto neľahkou úlohou!

Ako pomôcku dostaneš od nás zobrazovátko, ktorým si vieš vizualizovať Pulibotove programy. Viac o tom sa dozvieš v poslednej časti zadania.

Špecifikácia Pulibota

Každé políčko má celočíselný **stav**, a to nasledovný:

- Obvodové políčka majú stav -2.
- Vnútorné políčka s prekážkou majú stav -1.
- Prázdne vnútorné políčka majú stav rovný svojej farbe.

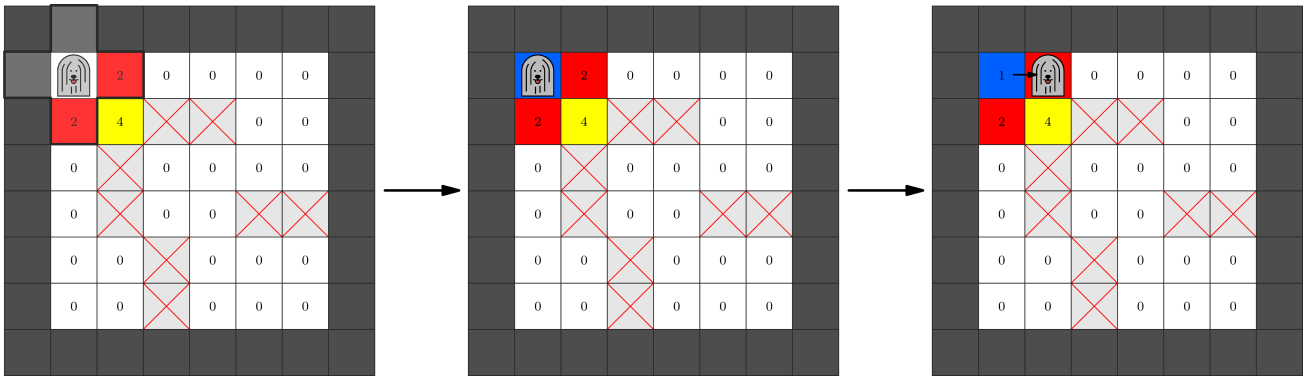
Pulibot vykonáva program v krokoch. V každom kroku sa pozrie na stavy okolitých políčok a podľa nich si vyberie inštrukciu, ktorú vykoná.

Presnejšie to celé vyzerá nasledovne. Na začiatku kroku je Pulibot na prázdnom políčku (r, c).

1. Ako prvé si Pulibot zozbiera aktuálne **pole stavov**: pole $S = [S[0], S[1], S[2], S[3], S[4]]$, v ktorom:
   - $S[0]$ je stav samotného políčka (r, c).
   - $S[1]$ je stav jeho suseda smerom na západ.
   - $S[2]$ je stav jeho suseda smerom na juh.
   - $S[3]$ je stav jeho suseda smerom na východ.
   - $S[4]$ je stav jeho suseda smerom na sever.
2. Obsahom poľa S je jednoznačne určená **inštrukcia** (Z, A), ktorú ide Pulibot v tomto kroku vykonať.
3. Pulibot vykoná dotyčnú inštrukciu, a to tak, že nastaví farbu aktuálneho políčka (r, c) na Z a následne vykoná akciu A. Existujú nasledujúce možné akcie:
   - *zostaň* na políčku (r, c);
   - *pohni sa* na niektoré zo 4 susedných políčok;

- *ukonči program*.

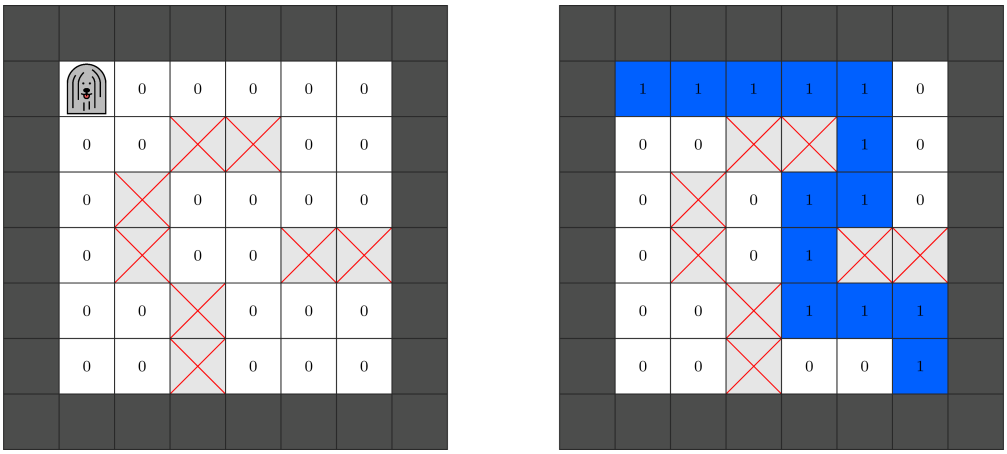
Uvažujme napríklad situáciu, ktorá je na ľavej strane nasledujúceho obrázku. Pulibot si v nej zozbiera pole stavov $S = [0, -2, 2, 2, -2]$. Ak by mal(a) program, ktorý tomuto poľu stavov priradí inštrukciu ($Z = 1$, pohni sa na východ), prefarbením by vznikla situácia v strede a následným pohybom situácia na pravej strane obrázku.



Pravidlá súťaže robotov

- Na začiatku umiestnia Pulibota na políčko $(0, 0)$. Tam začne vykonávať svoj program.
- Pulibot sa smie hýbať len po voľných políčkach. Nikdy ho teda nesmie tvoj program poslať ani na obvod, ani do prekážky.
- Pulibot musí ukončiť vykonávanie svojho programu po nanajvýš $500\,000$ krokoch.
- Keď sa tak stane, musia byť prázdne políčka ofarbené nasledovne:
  - Pre nejakú jednu konkrétnu najkratšiu cestu z $(0, 0)$ do $(H - 1, W - 1)$ musí platiť, že všetky jej políčka majú farbu 1.
  - Všetky ostatné prázdne políčka majú farbu 0.
- Po skončení programu smie Pulibot zostať stáť na ľubovoľnom (prázdnom) políčku.

Na nasledujúcom obrázku je jedno možné bludisko s $H = W = 6$. Vľavo je začiatočná konfigurácia, vpravo je jedno možné správne ofarbenie políčok po skončení programu.



Detaily implementácie

Implementuj nasledujúcu funkciu:

```
void program_pulibot()
```

- Táto funkcia vygeneruje a testovaču odovzdá tvoj program pre Pulibota. Tento program musí fungovať pre všetky platné bludiská všetkých možných rozmerov.
- Pri každom spustení tvojho programu túto funkciu testovač zavolá práve raz.

Tvoja funkcia odovzdáva vygenerovaný program tak, že postupne veľakrát zavolá nasledovnú funkciu testovača:

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

- S: pole dĺžky 5 predstavujúce pole stavov.
- Z: nezáporné celé číslo predstavujúce novú farbu.
- A: jedno písmeno predstavujúce akciu:
  - `H`: ostaň na mieste;
  - `W`: sprav krok na západ (west);
  - `S`: sprav krok na juh (south);
  - `E`: sprav krok na východ (east);
  - `N`: sprav krok na sever (north);
  - `T`: ukonči program (terminate).
- Zavolanie tejto funkcie naučí Pulibota, že keď vidí pole stavov S, má vykonať inštrukciu (Z, A).
- Túto funkciu nesmieš zavolať dvakrát s tým istým S.

Nie je nutné zavolať `set_instruction` pre úplne všetky teoreticky možné polia stavov S. Samozrejme, ty zodpovedáš za to, že vynecháš len také, ktoré tvoj Pulibot nevie nikdy uvidieť. Ak nastane počas vykonávania programu situácia, v ktorej si Pulibot zozbiera pole stavov, pre ktoré nemá definovanú inštrukciu, dostaneš verdikt `Output isn't correct`.

Keď dobehne tvoja funkcia `program_pulibot`, testovač postupne spustí tvoj program pre Pulibota na niekoľkých bludiskách. Testovač **nie je adaptívny** -- sada bludísk, na ktorých bude testovať je určená vopred.

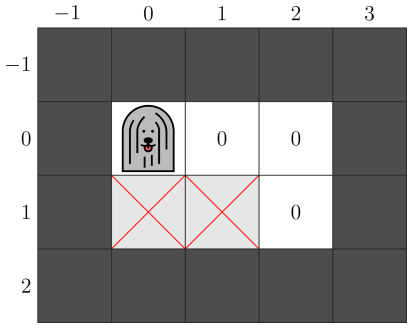
Ak Pulibot poruší hociktoré z pravidiel súťaže robotov, dostaneš verdikt `Output isn't correct`.

Príklad

Uvažujme scenár, v ktorom funkcia `program_pulibot` postupne spravila nasledovné volania `set_instruction`:

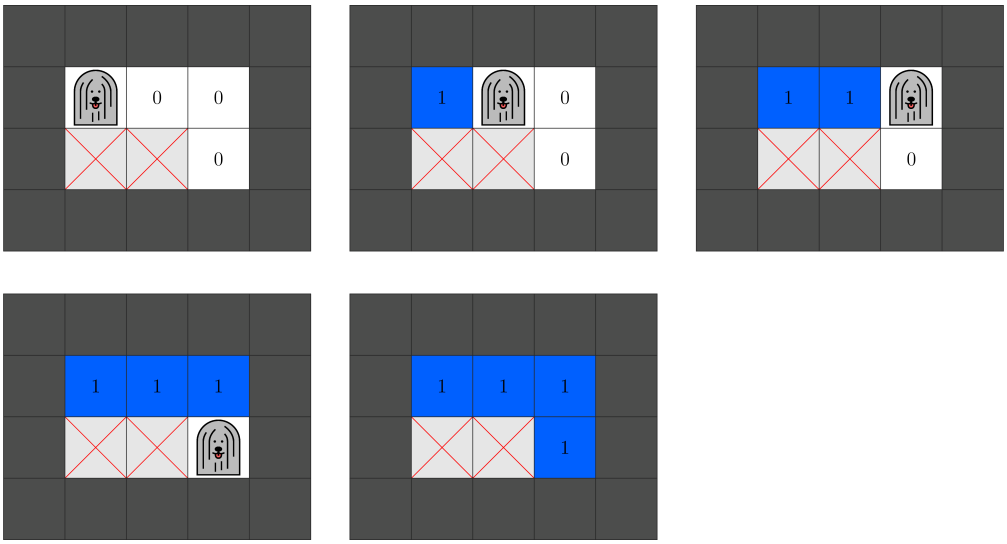
Volanie	Inštrukcia rozpísaná slovne
`set_instruction([0, -2, -1, 0, -2], 1, E)`	Nastav farbu na 1 a choď na východ
`set_instruction([0, 1, -1, 0, -2], 1, E)`	Nastav farbu na 1 a choď na východ
`set_instruction([0, 1, 0, -2, -2], 1, S)`	Nastav farbu na 1 a choď na juh
`set_instruction([0, -1, -2, -2, 1], 1, T)`	Nastav farbu na 1 a ukonči program

Ďalej uvažujme bludisko, v ktorom $H = 2$, $W = 3$ a prekážky sú rozmiestnené nasledovne:



Pre bludisko z obrázku by Pulibot vykonal postupne štyri kroky. Tie zodpovedajú presne vyššie uvedeným štyrom volaniam `set_instruction`, v tom poradí, v ktorom sú uvedené.

Na ďalšom obrázku je postupnosť zmien, ktorými prejde bludisko pri vykonávaní Pulibotovho programu.



Všimni si ale, že pre mnohé iné bludiská tento kratučký program nenájde správnu najkratšiu cestu. Ak by teda tvoja funkcia odovzdala tento konkrétny program, dostaneš verdikt `Output isn't correct`.

Obmedzenia

$Z_{MAX} = 19$. To znamená, že ako farby môžeš používať čísla od 0 po 19, vrátane.

V každom bludisku, na ktorom bude Pulibot testovaný, platí:

- $2 \leq H, W \leq 15$
- Existuje cesta prázdnymi políčkami z $(0, 0)$ do $(H - 1, W - 1)$.

Podúlohy

1. (6 bodov) V bludisku nie sú žiadne prekážky.
2. (10 bodov) $H = 2$
3. (18 bodov) Medzi každými dvoma prázdnymi políčkami existuje práve jedna cesta.
4. (20 bodov) Najkratšia cesta z $(0, 0)$ na $(H - 1, W - 1)$ má dĺžku $H + W - 2$.
5. (46 bodov) Žiadne ďalšie obmedzenia.

Ak v ľubovoľnom teste zavoláš funkciu `set_instruction` s neplatnými parametrami, dostaneš za príslušnú podúlohu 0 bodov. To isté sa stane aj vtedy, ak bude pri vykonávaní tvojho programu pre Pulibota porušené ľubovoľné z obmedzení uvedených v časti Detaily implementácie.

Za každú podúlohu sa dajú získať čiastočné body. Tie vieš dostať za program, ktorý nájde najkratšiu cestu, ale na ostatných políčkach nebude mať len farbu 0.

Presnejšie:

Tvoje riešenie testu je **úplné**, ak spĺňa Pravidlá súťaže robotov.

Tvoje riešenie testu je **čiastočné**, ak nie je úplné, ale platí:

- Existuje nejaká konkrétna najkratšia cesta z políčka $(0, 0)$ na políčko $(H - 1, W - 1)$, ktorej všetky políčka majú farbu 1.
- Žiadne iné voľné políčko nemá farbu 1.

V čiastočnom riešení teda existuje niekde mimo nájdenej cesty voľné políčko, ktoré má farbu inú od 0 aj 1.

Ak tvoje riešenie testu nie je ani úplné, ani čiastočné, dostaneš za ten test 0 bodov.

V podúlohách 1-4 dostaneš za úplne vyriešený test plný počet bodov a za čiastočne vyriešený test 50% bodov.

V podúlohe 5 tvoj výsledný počet bodov záleží nielen od toho, či je tvoje riešenie úplné, ale aj od toho, ako veľa farieb tvoj program pre Pulibota potrebuje použiť. Presnejšie, nech $Z^\star$ označuje najväčšiu zo všetkých hodnôt Z vo všetkých tvojích volaniach funkcie `set_instruction`. Tvoje body potom určíme podľa nasledovnej tabuľky:

Počet farieb	Body za úplné rieš.	Body za čiastočné rieš.
$11 \leq Z^\star \leq 19$	$20 + (19 - Z^\star)$	$12 + (19 - Z^\star)$
$Z^\star = 10$	31	23
$Z^\star = 9$	34	26
$Z^\star = 8$	38	29
$Z^\star = 7$	42	32
$Z^\star \leq 6$	46	36

Samozrejme, pre každú podúlohu sa tvoj výsledný počet bodov určí ako minimum bodov za jednotlivé testy v nej.

Ukážkový testovač

Ukážkový testovač očakáva na štandardnom vstupe nasledovné údaje:

- riadok 1: H W
- riadok $2 + r$ (pre $0 \leq r < H$): $s[r][0]$ $s[r][1]$ $\ldots$ $s[r][W-1]$

Pole s popisuje začiatočné stavy všetkých vnútorných políčok. Presnejšie, toto pole má H riadkov, W stĺpcov, a platí, že pre prázdne políčko (r, c) je $s[r][c] = 0$ a pre políčko s prekážkou je $s[r][c] = 1$.

Po načítaní vstupu ukážkový testovač zavolá funkciu `program_pulibot()`.

Ak ukážkový testovač uvidí ľubovoľné porušenie protokolu, skončí s chybovou hláškou `Protocol Violation: <MSG>`, kde `<MSG>` je jedna z nasledujúcich možných správ:

- `Invalid array`: pole S nemá `5` prvkov, alebo pre niektoré i neplatí podmienka $-2 \leq S[i] \leq Z_{MAX}$.
- `Invalid color`: hodnota Z nespĺňa $0 \leq Z \leq Z_{MAX}$.
- `Invalid action`: znak A sa líši od všetkých povolených (`H`, `W`, `S`, `E`, `N` a `T`).
- `Same state array`: to isté pole S už bolo použité v skoršom volaní `set_instruction`.

Ak funkcia `program_pulibot` korektne skončí, ukážkový testovač vykoná tvoj program pre Pulibota na bludisku, ktoré dostal na vstupe. Pri tom vyrobí dva výstupy:

Do súboru `robot.bin` v aktuálnom adresári ukážkový testovač zapíše log akcií, ktoré Pulibot vykonal. Tento súbor sa následne dá použiť ako vstup pre vizualizátor (o ktorom sa dočítaš nižšie).

Ak pri vykonávaní Pulibotovho programu nastala nejaká chyba, ukážkový testovač na výstup vypíše príslušnú chybovú hlášku:

- `Unexpected state`: Pulibot narazil pri vykonávaní na pole stavov, pre ktoré nebola zavolaná funkcia `set_instruction`.
- `Invalid move`: Pulibot sa pokúsil pohnúť na políčko, ktoré nie je prázdne.
- `Too many steps`: Pulibot vykonal $500\,000$ krokov a ešte nezastal.

Ak Pulibotov program dobehne korektne, ukážkový testovač dá iný výstup. Nech $e[r][c]$ je stav políčka (r, c) na konci behu Pulibotovho programu. Testovač vypíše H riadkov v nasledovnom tvare:

- riadok $1 + r$ (pre $0 \leq r < H$): $e[r][0]$ $e[r][1]$ $\ldots$ $e[r][W-1]$

Zobrazovátko

V prílohe k tejto úlohe nájdete program `display.py`. Spustiť ho vieš nasledovne:

```
python3 display.py
```

Keď ho spustíš, tento program ti graficky zobrazí priebeh Pulibotovho programu pri poslednom spustení ukážkového testovača. Používa na to súbor `robot.bin`, ktorý testovač vyrobí. (Tento súbor sa musí nachádzať v aktuálnom adresári, aby ho `display.py` našiel.)

Hlavné fičúrie zobrazovátka sú nasledovné:

- Môžeš si pozrieť celé bludisko. Políčko Pulibota je zvýraznené obdĺžnikom.
- Môžeš si krokovať Pulibotov program: buď klikaním na šípky alebo stláčaním kláves, ktoré im zodpovedajú.
- Môžeš skočiť na konkrétny krok vo vykonávaní programu.
- V spodnom riadku je vždy uvedený nasledujúci krok Pulibotovho programu: pole stavov, ktoré vidí, a inštrukciu, ktorú podľa neho vykoná. Ak si na konci programu, v spodnom riadku je buď chyba, ktorá nastala, alebo reťazec `Terminated`, ak program skončil korektne.
- Každému číslu farby vieš priradiť aj skutočnú farbu, ktorou má byť zobrazená, aj text, ktorý sa má v jej políčkach zjaviť. Toto vieš spraviť dvoma spôsobmi:
  - Kliknutím na tlačidlo `Colors` v aplikácii.
  - Zeditovaním textového súboru `colors.txt`.
- Tlačidlo `Reload` znovu načíta z disku obsah súboru `robot.bin`. Toto chceš spraviť, ak počas behu zobrazovátka spustíš ukážkový testovač a tým prepíšeš starý obsah súboru `robot.bin` novým.