

Find the Box | Surasti dėžę

Užduoties pavadinimas	Surasti dėžę
Laiko apribojimas	1 sekundė
Atminties apribojimas	1 gigabaitas

Maja yra robotikos tyrėja, dirbanti Lundo universitete. Ji sužinojo apie vertingą lobį universiteto rūsyje. Lobis yra dėžėje, esančioje tuščiam kambaryje giliai po žeme. Deja, Maja negali tiesiog eiti ir ieškoti dėžės. Rūsyje labai tamsu ir ėjimas su šviesa sukeltų įtarimų. Vienintelis jos būdas rasti lobį – nuotoliniu būdu valdyti rūsyje esantį dulkių siurblių-robotą.

Rūsys vaizduojamas kaip $H \times W$ dydžio tinklelis, kurio eilutės sunumeruotos nuo 0 iki $H - 1$ (iš viršaus į apačią), o stulpeliai sunumeruoti nuo 0 iki $W - 1$ (iš kairės į dešinę), t. y. viršutinis kairysis langelis yra $(0, 0)$ o apatinis dešinysis langelis yra $(H - 1, W - 1)$. Dėžė su lobiu yra nežinomame langelyje, bet ne $(0, 0)$. Kiekvieną naktį robotas dulkių siurblys pradeda darbą viršutiniame kairiajame kampe ir juda po rūšį.

Kiekvieną naktį Maja gali pateikti robotui instrukcijų seką apie tai, kaip jis turėtų judėti, eilutės (string), sudarytos iš simbolių "<", ">", "^" ir "v", pavidalu. Formaliai, jei robotas stovi (r, c) pozicijoje, kuri iš visų pusių neužblokuota, "<" perkelia robotą į kairę, į langelį $(r, c - 1)$, ">" perkelia robotą į dešinę, į langelį $(r, c + 1)$, "^" perkelia robotą aukštyn, į langelį $(r - 1, c)$, o "v" perkelia robotą žemyn, į langelį $(r + 1, c)$.

Rūsio sienos yra tvirtos, todėl, jei robotas bandys judėti už tinklelio ribų, nieko neatsitiks. Dėžė taip pat yra kieta (solid), todėl ji negali būti pastumta. Kiekvienos nakties pabaigoje robotas praneš savo buvimo vietą ir grįš į viršutinį kairįjį kampą.

Laikas labai svarbus, todėl Maja nusprendžia surasti dėžę per kuo mažesnį naktų skaičių.

Interakcija

Tai yra interaktyvi užduotis.

- Programa turėtų prasidėti nuo eilutės su dviem sveikaisiais skaičiais H ir W (tinklelio aukštis ir plotis) perskaitymo.
- Tada programa turėtų sąveikauti su vertinimo programa (grader). Kiekviename sąveikos raunde turėtumėte spausdinti klaustuką "?", o po jo netuščią eilutę (string) s , sudarytą iš

simbolių "<", ">", "^", "v". Ši eilutė turi būti ne ilgesnė kaip 20 000. Tada jūsų programa turėtų nuskaityti du sveikuosius skaičius r, c ($0 \leq r \leq H - 1$, $0 \leq c \leq W - 1$), tai yra roboto buvimo vietą po instrukcijų vykdymo. Svarbu pastebėti, kad po kiekvienos užklaustos robotas visada grįžta į $(0, 0)$.

- Kai žinote dėžės vietą, išspausdinkite "!", po kurio išspausdinkite du sveikuosius skaičius r_b, c_b , dėžės eilutę ir stulpelį ($0 \leq r_b \leq H - 1$, $0 \leq c_b \leq W - 1$). Tai padarius programa turi baigtis, neatlikdama jokių kitų užklausių. Nustatant jūsų įvertinimą šis galutinis išvesties rezultatas nėra laikomas užklausa.

Po užklaustos pateikimo standartinei išvesčiai būtinai atlikite procedūrą flush, kitaip jūsų programa gali būti įvertinta kaip "Klaida - viršytas laiko limitas" (Time Limit Exceeded). Python kalboje `print()` atlieka flush procedūrą automatiškai. C++ kalboje `cout << endl;` taip pat atlieka flush procedūrą (ir, žinoma, išspausdina naują eilutę); jei naudojate `printf`, naudokite `fflush(stdout)`.

Vertinimo programa (grader) yra neadaptivi, t. y. dėžės padėtis nustatoma prieš pradėdant sąveikavimą.

Apribojimai ir vertinimas

- $1 \leq H, W \leq 50$.
- Dėžė niekada nebus laukelyje $(0, 0)$. Tai reiškia, kad $H + W \geq 3$.
- Kiekvieną užklausą gali sudaryti ne daugiau kaip 20 000 instrukcijų.
- Galite pateikti ne daugiau kaip 2 500 užklausių (galutinio atsakymo spausdinimas neskaičiuojamas kaip užklausa).

Jūsų programa bus testuojama keliais testavimo variantais. Jei jūsų sprendimas bus atmestas *bet kuriame* iš šių testų (pvz., programa pateiks neteisingą dėžės padėtį kaip atsakymą (wrong box position - Wrong Answer), programa neveiks (crashing - Runtime Error), viršys laiko limitą (Time Limit Exceeded) ir t. t.), gausite 0 taškų ir atitinkamą įvertinimą.

Jei jūsų programa sėkmingai ras dėžės padėtį *visais* bandymų atvejais, gausite verdiktą Accepted ir taškų skaičių, apskaičiuotą taip:

$$\text{vertinimas} = \min\left(\frac{100\sqrt{2}}{\sqrt{Q}}, 100\right) \text{ balai,}$$

kur Q yra didžiausias užklausių, naudojamų bet kuriam testo atvejui, skaičius. Galutinio atsakymo spausdinimas nelaikomas užklausa. Vertinimo rezultatas bus suapvalinamas iki artimiausio sveikojo skaičiaus.

Kad gautumėte 100 taškų, jūsų programa turi išspręsti kiekvieną testo atvejį naudodama ne daugiau kaip $Q = 2$ užklausių. Toliau pateiktoje lentelėje parodytos kai kurios Q reikšmės ir su jomis susiję taškai.

Q	2	3	4	5	...	20	...	50	...	2500
Taškai	100	82	71	63	...	32	...	20	...	3

Testavimo įrankis

Kad būtų lengviau testuoti savo parašytą programą, pateikiame paprastą įrankį, kurį galite atsisiųsti. Žr. "Priedai" ("Attachments") Kattis užduoties puslapyje apačioje. Įrankiu naudotis neprivaloma, ir jums leidžiama jį keisti. Atkreipkite dėmesį, kad oficiali Kattis vertinimo programa (grader) skiriasi nuo testavimo įrankio.

Pavyzdinis naudojimas (kai $H = 4$, $W = 5$ ir paslėpta dėžė pozicijoje $r = 2$, $c = 3$):

Python programoms, tarkime, `solution.py` (kuri paprastai paleidžiama kaip `pypy3 solution.py`):

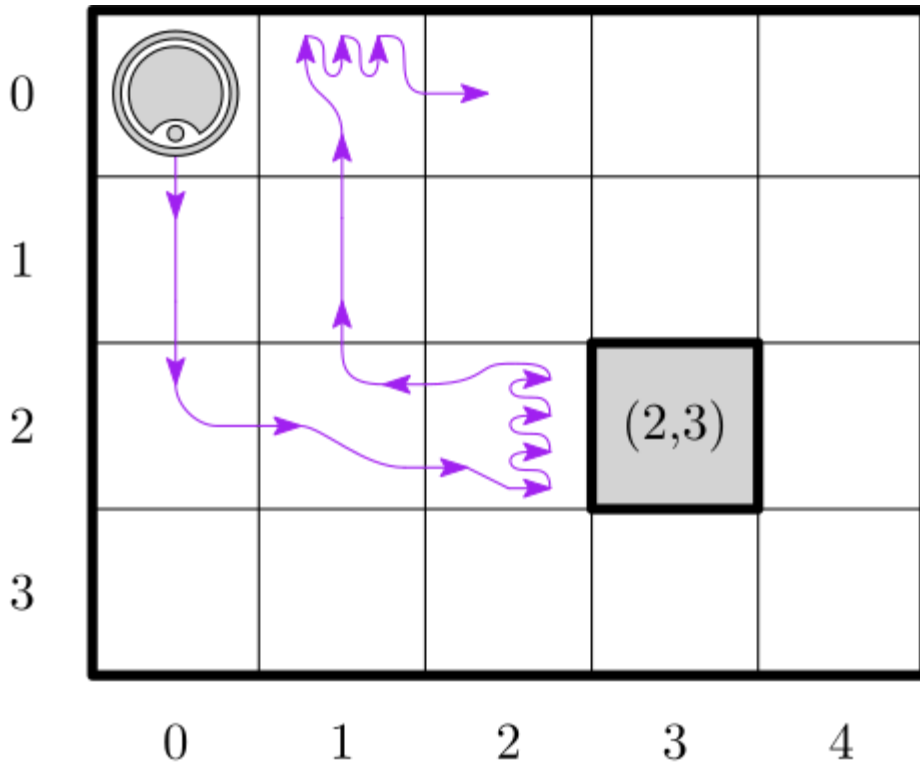
```
python3 testing_tool.py pypy3 solution.py <<<"4 5 2 3"
```

C++ programoms pirmiausia programą sukompiliuokite: (pavyzdžiui, su `g++ -std=gnu++17 solution.cpp -o solution.out`) ir tada programą paleiskite:

```
python3 testing_tool.py ./solution.out <<<"4 5 2 3"
```

Pavyzdys

Panagrinėkime pavyzdinį testavimo atvejį. Tinklelio aukštis $H = 4$, plotis $W = 5$, o dėžės padėtis $(r, c) = (2, 3)$. Toliau pateiktame paveikslėlyje parodytas roboto kelias, kai robotas vykdo pirmosios užklauskos "`? vv>>>>>><^<^<^<^<^<`" nurodymus – robotas atsiduria padėtyje $(r, c) = (0, 2)$. Prieš antrąją užklauską robotas vėl grįš į viršutinį kairįjį kampą $(0, 0)$. Tada programa pateikia dar vieną užklauską "`? >>>>>>>vvvvvvvvvvvvvvvv`", kurią atlikęs robotas atsiduria dešiniajame apatiniame kampe $(r, c) = (3, 4)$. Dabar programa nusprendžia spėti atsakymą, todėl atspausdinama "`! 2 3`", o tai yra teisinga dėžės padėtis.



Vertinimo programos atsakymas	Jūsų programos atsakymas
4 5	
	? v>>>>>>>>><^>>>>>>>
0 2	
	? >>>>>>>>>v<<<<<<<<<<
3 4	
	! 2 3