Hledání pokladu

Název úlohy	Find the box
Časový limit	1 sekunda
Paměťový limit	1 gigabyte

Mája je výzkumnice v oblasti robotiky pracující na LTH. Mája zaslechla, že se v univerzitním sklepení nachází drahocený poklad. Poklad je uložen v truhle v jinak prázdné místnosti hluboko pod zemí. Bohužel, Mája nemůže jen tak jít do sklepa a hledat truhlu. Je tam veliká tma a jít tam se světlem by mohlo vyvolat podezření. Jediný způsob, jak Mája může poklad najít, je za pomoci robotického vysavače na dálkové ovládání, který se ve sklepě nachází.

Sklep můžeme reprezentovat jako tabulku $H \times W$, jejíž řádky jsou očíslovány od 0 do $H-1$ (shora dolů) a jejíž sloupce jsou očíslovány od 0 do $W-1$ (zleva doprava). V levém horním rohu se tedy nachází políčko $(0,0)$ a v pravém dolním políčko $(H-1, W-1)$. Truhla s pokladem se nachází na neznámém políčku. Robotický vysavač začíná každý večer v levém horním rohu a odtamtud se vydává na cestu po sklepě.

Mája může každý večer robotovi předat posloupnost instrukcí, jak se má pohybovat, v podobě řetězce skládajícího se ze znaků "<", ">", "^" a "v". Jestliže se robot nachází na políčku (r, c), které není blokováno z žádné strany, "<" pohne robota doleva na políčko $(r, c-1)$, ">" pohne robota doprava na políčko $(r, c+1)$, "^" pohne robota nahoru na políčko $(r-1, c)$, a "v" pohne robota dolů na políčko $(r+1, c)$.

Stěny místnosti jsou neprůchozí, takže pokud se robot pokusí pohnout mimo tabulku, nic se nestane. Truhla s pokladem je také neprůchozí a nelze ji posunout. Na konci každé noci robot zahlásí svoji pozici a automaticky se vrátí do levého horního rohu místnosti.

Čas je rozhodující, takže se Mája snaží najít truhlu s pokladem za co nejméně nocí.

Interakce

Toto je interaktivní úloha.

- Váš program by měl začít načtením řádku s dvěma celými čísly H a W: výška a šířka tabulky.
- Následně by váš program měl komunikovat s graderem (testovacím programem). V každém kole interakce byste měli vypsat "?" následovaný neprázdným řetězcem s skládajícím se ze

znaků “<”, “>”, “^”, “v”. Délka řetězce může být nejvýše $20\,000$. Poté by váš program měl načíst dvě celá čísla r, c ($0 \le r \le H-1,\ 0 \le c \le W-1$), polohu robota po splnění instrukcí. Nezapomeňte, že se mezi každými dvěma dotazy robot vrací zpátky na $(0, 0)$.
- Jakmile odhalíte polohu pokladu, vypište “`!`” následovaný dvěma celými čísly r_b, c_b, řádek a sloupec, ve kterém se truhla s pokladem nachází. ($0 \le r_b \le H-1,\ 0 \le c_b \le W-1$). Následně váš program musí skončit, aniž by položil další dotazy. Ve výpočtu skóre se vaše finální odpověď nepočítá do počtu dotazů.

Ujistěte se, že flushujete standardní výstup po každém dotazu, jinak byste mohli dostat verdikt Time Limit Exceeded. V Pythonu `print()` flushuje automaticky. V C++ `cout << endl;` také flushuje automaticky a navíc vypíše konec řádku. Pokud používáte `printf`, flushujte výstup pomocí `fflush(stdout)`.

Grader není adaptivní. To znamená, že poloha truhly je pevně určena ještě předtím, než začne interakce s vaším programem.

Omezení a bodování

- $1 \le H, W \le 50$.
- Truhla nikdy nebude na políčku $(0, 0)$. To znamená, že $H + W \ge 3$.
- Každý dotaz (řetězec instrukcí) smí obsahovat nejvýše $20\,000$ znaků.
- Můžete použít nejvýše $2\,500$ dotazů (finální odpověď se nepočítá do počtu dotazů).

Vaše řešení bude testováno na určitém počtu testů. Pokud vaše řešení selže na *libovolném* testu (např. odpoví špatně pozici pokladu (Wrong Answer), spadne (Runtime Error), překročí časový limit (Time Limit Exceeded) atd.), získáte 0 bodů a obdržíte příslušný verdikt.

Pokud váš program najde pozici pokladu správně ve *všech* testech, obdržíte verdikt Accepted a vaše skóre bude spočítáno následujícím způsobem:

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

kde Q je maximální počet položených dotazů přes všechny testy. Vypisování odpovědi se nepočítá jako dotaz. Skóre bude zaokrouhleno na nejbližší celé číslo.

Pro zisk 100 bodů, musí váš program vyřešit všechny testy na nejvýše $Q = 2$ dotazy. Tabulka níže ukazuje skóre pro některé hodnoty Q.

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

Testovací nástroj

Pro usnadnění lokálního testování řešení je vám poskytnut jednoduchý testovací nástroj, který si můžete stáhnout. Najdete jej v "attachments" dole na Kattis stránce této úlohy. Použití testovacího nástroje je nepovinné a můžete jej libovolně modifikovat. Oficiální grader používaný v Kattisu se od testovacího nástroje může lišit.

Ukázkové použití (s parametry $H = 4$, $W = 5$ a s truhlou na políčku $r = 2$, $c = 3$):

Pro program v Pythonu, například `solution.py` (který byste mohli spustit jako `pypy3 solution.py`):

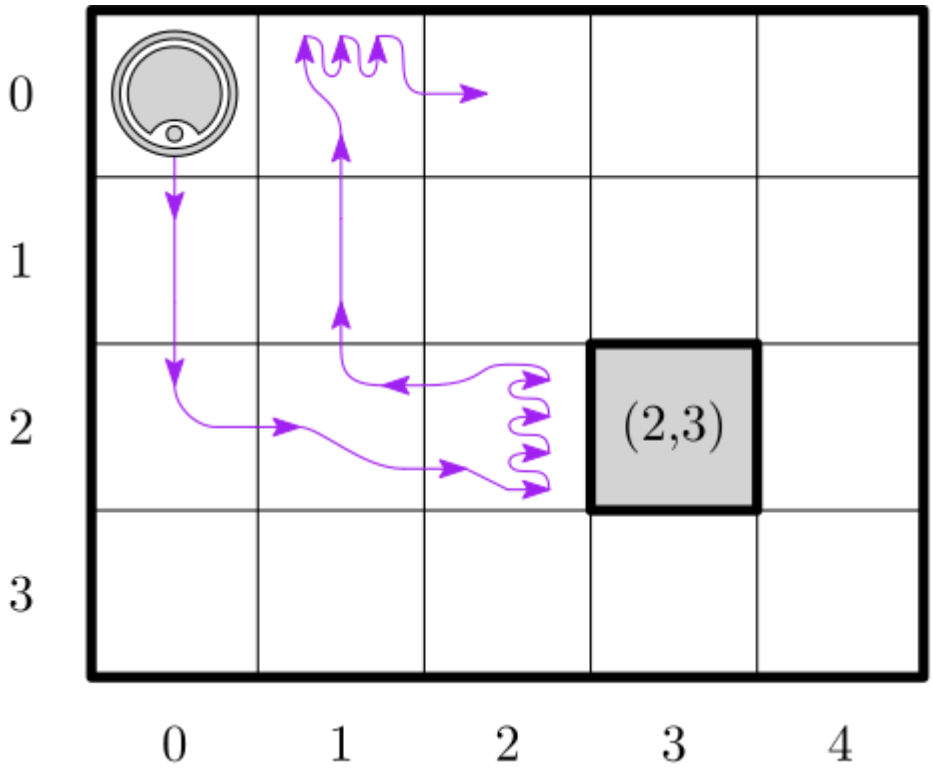
```
python3 testing_tool.py pypy3 solution.py <<<"4 5 2 3"
```

Program v C++, například `solution.cpp`, nejprve zkompilujte (např. pomocí `g++ -g -O2 -std=gnu++17 -static solution.cpp -o solution.out`) a následně spusťte:

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

Příklad

Uvažme ukázkový test s mřížkou výšky $H = 4$, šířky $W = 5$ a s truhlou na políčku $(r, c) = (2, 3)$. Obrázek níže zachycuje pohyb robota pro první dotaz s instrukcemi "`? vv>>>>>><^^^^^>`", podle kterých robot skončí na políčku $(r, c) = (0, 2)$. Před druhým dotazem se robot vrátí zpět do levého horního rohu na políčko $(0, 0)$. Následně řešení položí další dotaz s instrukcemi "`? >>>>>>>>vvvvvvvvvv`" podle nichž se robot dostane do pravého dolního rohu na políčko $(r, c) = (3, 4)$. Následně se řešení rozhodne odpovědět a výpíše "`! 2 3`", což je správná poloha truhly s pokladem.



výstup graderu	váš výstup
4 5	
	? vv>>>>>><^^^^^>
0 2	
	? >>>>>>>>vvvvvvvvv
3 4	
	! 2 3