

Malování podle čísel

Malování podle čísel je dobře známá logická hra. Mějme jednoduchou jednorozměrnou verzi této hry. V této hře dostane hráč řadu n buněk. Buňky jsou očíslovány od 0 do $n - 1$ zleva doprava. Hráč může každou buňku namalovat černě nebo bíle. Pomocí 'X' budeme označovat černé buňky a pomocí '_' bílé buňky.

Hráč dostane posloupnost k kladných čísel $c = [c_0, \dots, c_{k-1}]$ jako návod. Má podle něj malovat buňky takovým způsobem, že černé buňky na řádku vytvářejí přesně k souvislých bloků po sobě jdoucích buněk. Tyto bloky jsou očíslovány od nuly zleva doprava. Počet černých buněk v i -tém bloku musí být roven c_i . Například je-li návodem $c = [3, 4]$, pak řešení musí mít právě dva bloky po sobě následujících černých buněk: jeden o délce 3 a druhý délky 4. Je-li $n = 10$ a $c = [3, 4]$, pak jedno řešení vyhovující tomuto návodu je "_XXX_XXXX". Všimněte si, že řešení "XXXX_XXX_" návodu neodpovídá, protože bloky černých buněk nejsou ve správném pořadí. Také řešení "_____" nevyhovuje, neboť obsahuje jediný blok černých buněk a ne dva oddělené bloky.

Dostanete částečně vyřešenou hru. To znamená, že znáte n a c a navíc víte, že některé buňky musí být černé a některé buňky musí být bílé. Vaším úkolem je odvodit další informace o barvě buněk.

Za *platné řešení* označíme takové, které odpovídá návodu a zároveň také respektuje zadané barvy některých buněk. Váš program musí najít buňky, které jsou černé v každém platném řešení, a buňky, které jsou bílé v každém platném řešení. Můžete předpokládat, že vstup je takový, že existuje alespoň jedno platné řešení.

Implementační detaily

Implementujte následující funkci (metodu):

- `string solve_puzzle(string s, int[] c)`.
 - s : řetězec délky n . Pro každé i ($0 \leq i \leq n - 1$) je i -tý znak:
 - 'X', jestliže buňka i musí být černá,
 - '_', jestliže buňka i musí být bílá,
 - '.', jestliže o buňce i nic nevíme.
 - c : pole délky k obsahující výše popsaný návod,
 - funkce musí vrátit řetězec délky n . Pro každé i ($0 \leq i \leq n - 1$) je i -tý znak vráceného řetězce:
 - 'X', jestliže buňka i je černá v každém platném řešení,
 - '_', jestliže buňka i je bílá v každém platném řešení,
 - '?' v ostatních případech (tj. existují dvě platná řešení taková, že buňka i je černá v jednom a bílá v druhém).

V jazyce C je signatura funkce odlišná:

- `void solve_puzzle(int n, char* s, int k, int* c, char* result)`
 - `n`: délka řetězce `s` (počet buněk),
 - `k`: délka pole `c` (počet prvků návodu),
 - ostatní parametry viz výše,
 - namísto vrácení řetězce délky `n` znaků musí funkce zapsat výsledek do řetězce `result`.

ASCII kódy znaků použitých v této úloze jsou:

- `'X'`: 88,
- `'_'`: 95,
- `'.'`: 46,
- `'?'`: 63.

Použijte poskytnuté ukázkové soubory pro získání implementačních detailů ve vašem programovacím jazyce.

Příklady

Příklad 1

```
solve_puzzle(".....", [3, 4])
```

Všechna možná řešení hry jsou:

- `"XXX_XXXX_"`,
- `"XXX__XXXX_"`,
- `"XXX___XXXX"`,
- `"_XXX_XXXX_"`,
- `"_XXX__XXXX"`,
- `"__XXX_XXXX"`.

Lze si povšimnout, že buňky s indexy 2, 6 a 7 (počítáno od 0) jsou v každém platném řešení černé. Všechny ostatní buňky mohou ale nemusí být černé, tj. správná odpověď je `"??X???XX??"`.

Příklad 2

```
solve_puzzle(".....", [3, 4])
```

V tomto případě je celé řešení jednoznačně určené a správná odpověď je `"XXX_XXXX"`.

Příklad 3

```
solve_puzzle("..._.....", [3])
```

V tomto případě můžeme odvodit, že buňka 4 musí být bílá, jinak by nebylo možné vložit blok tří černých buněk mezi bílé ležící na indexech 3 a 5. Tudíž správná odpověď je `"???_????"`.

Example 4

```
solve_puzzle(".X.....", [3])
```

Existují právě dvě řešení:

- “XXX_____”,
- “_XXX_____”.

Tudíž správná odpověď je “?XX?_____”.

Podúlohy

Ve všech podúlohách platí $1 \leq k \leq n$ a $1 \leq c_i \leq n$ pro každé $0 \leq i \leq k - 1$.

1. (7 bodů) $n \leq 20$, $k = 1$, s obsahuje pouze ‘.’ (prázdná hra),
2. (3 bodů) $n \leq 20$, s obsahuje pouze ‘.’,
3. (22 bodů) $n \leq 100$, s obsahuje pouze ‘.’,
4. (27 bodů) $n \leq 100$, s obsahuje pouze ‘.’ a ‘_’ (informace jen o bílých buňkách),
5. (21 bodů) $n \leq 100$,
6. (10 bodů) $n \leq 5\,000$, $k \leq 100$,
7. (10 bodů) $n \leq 200\,000$, $k \leq 100$.

Vzorový vyhodnocovač

Vzorový vyhodnocovač čte vstup v následujícím formátu:

- řádek 1: řetězec s ,
- řádek 2: celé číslo k následované k celými čísly c_0, \dots, c_{k-1} .