



## Kam sa pozrieť?

Pošťastilo sa ti dostať sa do finále známej televíznej súťaže *Komu sa pošťastí?*, v ktorom sa hrá o riadne prachy.

V rade pred tebou je  $n$  nepriehľadných krabíc očíslovaných zľava do prava 0 až  $n - 1$ . V každej krabici sa skrýva nejaká cena. Dokopy sa v krabiciach nachádza  $v \geq 2$  rôznych *druhov* cien. Tieto druhy sú očíslované od 1 po  $v$  v *klesajúcom* poradí podľa ich hodnoty.

Najdrahší je druh číslo 1 -- diamanty. Diamant je iba jeden a nachádza sa v niektorej z krabíc. Najlacnejší je druh  $v$  -- lízanky. Aby bola hra zaujímavšia, lacnejších cien je výrazne viac ako tých drahších. Presnejšie, ak máme  $k$  cien druhu  $t - 1$ , potom existuje *ostro viac* než  $k^2$  cien druhu  $t$ , pre všetky  $2 \leq t \leq v$ .

Podľa pravidiel súťaže si môžeš vybrať jednu krabicu, pozrieť sa do nej a jej obsah si nechať ako svoju výhru. Samozrejme, chceš vyhrať to najdrahšie, teda diamant. Predtým sa ešte môžeš moderátorky Moniky postupne povypytovať na niekoľko krabíc.

Ak sa spýtaš na krabicu  $i$ , Monika ti povie dve čísla (ktoré budú v praxi vyzerieť ako jedno dvojprvkové pole  $a$ ):

- $a[0]$  je počet krabíc, ktoré sú *naľavo* od  $i$  a obsahujú ceny *drahšie* ako je cena v krabici  $i$ .
- $a[1]$  je počet krabíc, ktoré sú *napravo* od  $i$  a obsahujú ceny *drahšie* ako je cena v krabici  $i$ .

Napríklad pre  $n = 8$  a otázku  $i = 2$  odpovie Monika poľom  $a = [1, 2]$ , ktoré znamená že:

- Práve jedna z krabíc 0 a 1 obsahuje cenu, ktorá je drahšia ako je cena v krabici 2.
- Práve dve krabice spomedzi krabíc 3, 4, ..., 7 obsahujú cenu, ktorá je drahšia ako je cena v krabici 2.
- Tvojou úlohou je pomocou malého počtu otázok zistiť, v ktorej krabici sa nachádza diamant.

## Implementačné detaily

Implementuj nasledujúcu funkciu:

```
int find_best(int n)
```

- Táto funkcia bude testovačom zavolaná práve raz.
- $n$ : počet krabíc.
- Funkcia má vrátiť jediné číslo  $d$  ( $0 \leq d \leq n - 1$ ): číslo krabice, v ktorej sa nachádza cena druhu 1 (diamant).

Funkcia `find_best` môže volať nasledovnú funkciu:

```
int[] ask(int i)
```

- $i$ : číslo krabice, na ktorú sa pýtaš Moniky. Hodnota  $i$  musí byť medzi 0 a  $n - 1$  vrátane.
- Funkcia vráti pole obsahujúce 2 prvky (nazvime ho  $a$ ). Hodnota  $a[0]$  je počet krabíc, ktoré sú naľavo od  $i$  a obsahujú drahšie ceny ako obsahuje krabica  $i$ . Hodnota  $a[1]$  je počet krabíc, ktoré sú napravo od  $i$  a obsahujú drahšie ceny ako obsahuje krabica  $i$ .

## Príklad

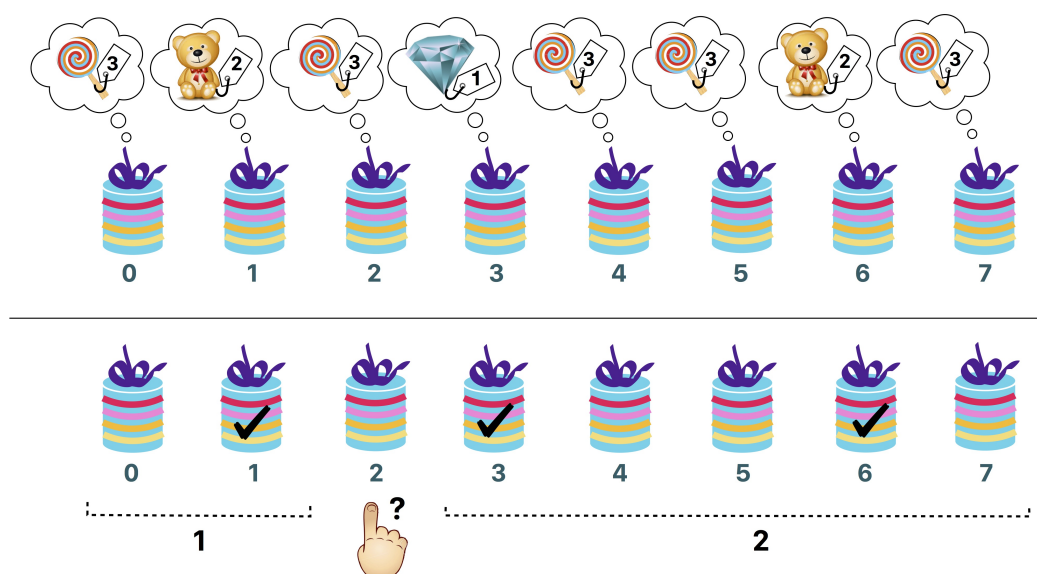
Testovač urobí nasledujúce volanie:

```
find_best(8)
```

Máme teda  $n = 8$  krabíc. Predpokladajme, že druhy cien v týchto krabiciach sú  $[3, 2, 3, 1, 3, 3, 2, 3]$ . Všetky možné volania funkcie `ask` a príslušné odpovede sú nasledovné:

- `ask(0)` vráti  $[0, 3]$
- `ask(1)` vráti  $[0, 1]$
- `ask(2)` vráti  $[1, 2]$
- `ask(3)` vráti  $[0, 0]$
- `ask(4)` vráti  $[2, 1]$
- `ask(5)` vráti  $[2, 1]$
- `ask(6)` vráti  $[1, 0]$
- `ask(7)` vráti  $[3, 0]$

V tomto príklade je diamant v krabici číslo 3. Funkcia `find_best` má teda vrátiť 3.



V hornej časti obrázka sú zobrazené druhy cien v jednotlivých krabiciach. Dolná časť ilustruje otázku `ask(2)`. Označené krabice obsahujú drahšie ceny než krabica 2.

## Obmedzenia

- $3 \leq n \leq 200\,000$ .
- V každej krabici je cena jedného z druhov 1 až  $v$  vrátane.
- V práve jednej krabici je cena druhu 1.
- Pre všetky  $2 \leq t \leq v$  platí, že ak počet cien druhu  $t - 1$  je  $k$ , počet cien druhu  $t$  je *ostro* viac než  $k^2$ .

## Podúlohy a hodnotenie

V niektorých testovacích vstupoch sa testovač správa adaptívne. To znamená, že nemá fixnú postupnosť cien, ale vymýšľa si ju za behu, v závislosti od vašich otázok. Testovač odpovedá tak, aby vždy existovala aspoň jedna možná postupnosť cien konzistentná so všetkými doterajšími otázkami.

1. (20 bodov) Existuje práve 1 diamant a  $n - 1$  lízaniek (teda  $v = 2$ ). Funkciu `ask` môžete volať najviac 10 000 ráz.
2. (80 bodov) Žiadne dodatočné obmedzenia.

V podúlohe 2 sa dajú získať čiastkové body. Nech  $q$  je maximálny počet volaní funkcie `ask`, ktoré váš program urobí, spomedzi všetkých vstupov v tejto podúlohe. Vaše skóre sa potom vypočíta podľa nasledovnej tabuľky:

Otázky	Skóre
$10\,000 < q$	0 (v CMS sa zobrazí ako 'Wrong Answer')
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

## Ukázkový testovač

Ukázkový testovač nie je adaptívny. Namiesto toho iba načíta fixné pole  $p$ , v ktorom sú druhy cien v jednotlivých krabiciach. Pre každé  $0 \leq b \leq n - 1$  je v krabici  $b$  cena druhu  $p[b]$ .

Ukázkový testovač očakáva vstup v nasledovnom formáte:

- riadok 1:  $n$
- riadok 2:  $p[0] \ p[1] \ \dots \ p[n - 1]$

Testovač vypíše jediný riadok s výsledkom funkcie `find_best` a počtom volaní funkcie `ask`.