



Úžasná výhra

Dostali jste se do finále populární televizní soutěže Úžasná výhra. Stojíte před řadou n krabic, které jsou zleva doprava očíslované 0 až $n - 1$. V každé krabici je schovaná cena, kterou nemůžete vidět, dokud neotevřete krabici. Existuje $v \geq 2$ různých *typů* cen. Typy jsou očíslované od 1 do v v *klesajícím* pořadí podle hodnoty.

Cena typu 1 má nejvyšší hodnotu: diamant. Diamant je schovaný v právě jedné z krabic. Cena typu v má nejnížší hodnotu: lízátko. Aby byla atmosféra soutěže napjatější, počet cen s menší hodnotou je mnohem větší než počet cen s vyšší hodnotou. Konkrétně, pro každé t splňující $2 \leq t \leq v$ musí platit následující podmínka: jestliže krabice obsahují právě k cen typu $t - 1$, pak musí obsahovat *ostře více* než k^2 cen typu t .

V soutěži máte za úkol získat diamant. Na konci soutěže budete muset otevřít jednu krabici a dostanete cenu, kterou obsahuje. Předtím než si zvolíte krabici, kterou otevřete, se můžete zeptat moderátora soutěže Ramboda na několik dotazů. Při každém dotazu si zvolíte nějakou krabici i . Rambod vám na dotaz odpoví polem a se dvěma celými čísly. Význam těchto čísel je následující:

- Mezi krabicemi nalevo od krabice i je právě $a[0]$ krabic, které obsahují cenu s vyšší hodnotou než krabice i .
- Mezi krabicemi napravo od krabice i je právě $a[1]$ krabic, které obsahují cenu s vyšší hodnotou než krabice i .

Předpokládejme například, že $n = 8$. Při svém dotazu si zvolíte krabici $i = 2$. Rambod vám na dotaz odpoví, že $a = [1, 2]$. Význam této odpovědi je následující:

- Právě jedna z krabic 0 a 1 obsahuje cenu s vyšší hodnotou než krabice 2.
- Právě jedna z krabic 3, 4, \dots , 7 obsahuje cenu s vyšší hodnotou než krabice 2.

Váším úkolem je najít krabici obsahující diamant tím, že položíte moderátorovi malý počet dotazů.

Podrobnosti k implementaci

Implementujte následující proceduru:

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

Vyhodnocovač zavolá tuto proceduru právě jednou.

- n : počet krabic.
- Procedura musí vrátit číslo krabice, která obsahuje diamant, tj. unikátní číslo d (

$0 \leq d \leq n - 1$) takové, že krabice d obsahuje cenu typu 1.

Výše uvedená procedura může volat následující proceduru:

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

- i : číslo krabice, na kterou se chcete zeptat. Hodnota i musí být mezi 0 a $n - 1$ včetně.
- Tato procedura musí vrátit pole a délky 2. Hodnota $a[0]$ je rovna počtu cen s vyšší hodnotou nalevo od krabice i . Hodnota $a[1]$ je rovna počtu cen s vyšší hodnotou napravo od krabice i .

Příklad

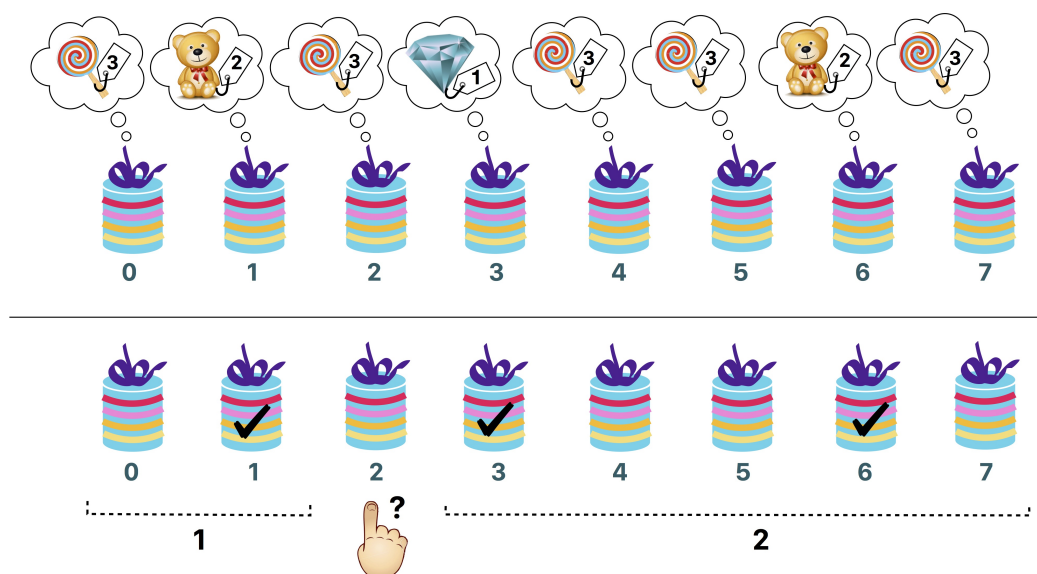
Vyhodnocovač provede následující volání procedury:

```
find_best(8)
```

Soutěží se tedy s $n = 8$ krabicemi. Předpokládejme, že typy cen jsou $[3, 2, 3, 1, 3, 3, 2, 3]$. Všechna možná volání procedury `ask` a jejich návratové hodnoty jsou uvedena níže.

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

U tohoto příkladu je diamant v krabici 3. Procedura `find_best` tedy musí vrátit 3.



Tento příklad je znázorněn na obrázku výše. V horní části obrázku jsou uvedeny typy cen v jednotlivých krabicích. Spodní část obrázku zobrazuje dotaz `ask(2)`. Označené krabice obsahují ceny s vyšší hodnotou než krabice 2.

Omezení

- $3 \leq n \leq 200\,000$.
- Typ ceny v každé krabici je mezi 1 a v včetně.
- Existuje právě jedna krabice s cenou typu 1.
- Pro všechna $2 \leq t \leq v$ platí: jestliže krabice obsahují právě k cen typu $t - 1$, pak obsahují *ostře* více než k^2 cen typu t .

Podúlohy a bodování

V některých testovacích případech se chování vyhodnocovače přizpůsobuje položeným dotazům. To znamená, že v těchto testovacích případech nemá vyhodnocovač neměnnou posloupnost cen. Naopak platí, že odpovědi vrácené vyhodnocovačem mohou záviset na dotazech položených vaším řešením. Máte zaručeno, že vyhodnocovač vrátí odpovědi takovým způsobem, že po každém zodpovězení dotazu existuje alespoň jedna posloupnost cen konzistentní se všemi dosud vrácenými odpověďmi.

1. (20 bodů) Existuje právě 1 diamant a $n - 1$ lízátek (tedy, $v = 2$). Proceduru `ask` můžete zavolat nejvýše 10 000krát.
2. (80 points) Žádná další omezení.

V podúloze 2 můžete získat částečný počet bodů. Označme q maximální počet volání procedury `ask` přes všechny testovací případy v této podúloze. Váš počet bodů za tuto podúlohu bude spočítán podle následující tabulky:

Dotazy	Body
$10\,000 < q$	0 (v CMS zobrazeno jako 'Wrong Answer')
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač nepřizpůsobuje posloupnost pokládaným dotazům. Namísto toho čte a používá fixní pole p s typy cen. Pro každé $0 \leq b \leq n - 1$ je typ ceny v krabici b roven $p[b]$. Ukázkový vyhodnocovač očekává vstup v následujícím formátu:

- řádek 1: n

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

Ukázkový vyhodnocovač vypíše jeden řádek obsahující návratovou hodnotu procedury `find_best` a počet volání procedury `ask`.