



Големата Награда

Големата Награда е познато телевизиско шоу. Вие сте среќниот натпреварувач кој стигнал до последниот круг на шоуто. Пред вас е поставена низа од n кутии, означени од 0 до $n - 1$ од лево кон десно. Секоја кутија содржи награда која не може да биде видена додека не се отвори кутијата. Постојат $v \geq 2$ различни *типови* на награди. Типовите на награди се нумерирани од 1 до v во *опаѓачки* редослед според вредноста на наградата.

Наградата од тип 1 е најскапата: дијамант. Во низата од кутии се наоѓа точно еден дијамант. Наградата од тип v е најевтината: лажавче. За да шоуто биде поинтересно, бројот на поефтини награди е многу поголем од бројот на поскапи награди. Поточно, за секое t такво што $2 \leq t \leq v$ го знаеме следното: ако во кутиите има k награди од тип $t - 1$, тогаш има *строго повеќе* од k^2 награди од тип t .

Ваша цел е да го освоите дијамантот. На крајот на играта ќе мора да отворите една кутија и ќе ја добиете наградата содржана во таа кутија. Пред да морате да одберете кутија можете да поставите прашања до Даре, водителот на шоуто. За секое прашање изберате некоја кутија i . Како одговор Даре ќе ви даде низа a составена од два цели броеви. Нивното значење е следното:

- Во кутиите што се наоѓаат лево од кутијата i постојат точно $a[0]$ кутии што содржат поскапа награда од онаа во кутијата i .
- Во кутиите што се наоѓаат десно од кутијата i постојат точно $a[1]$ кутии што содржат поскапа награда од онаа во кутијата i .

На пример, нека претпоставите дека $n = 8$. За ваше прашање ја изберате кутијата $i = 2$. Како одговор Даре ви кажува дека $a = [1, 2]$. Значењето на овој одговор е:

- Точно една од кутиите 0 и 1 содржи поскапа награда од наградата во кутијата 2
- Точно две од кутиите 3, 4, ..., 7 содржат поскапа награда од наградата во кутијата 2.

Ваша задача е да ја најдете кутијата што го содржи дијамантот поставувајќи мал број на прашања.

Детали за имплементација

Треба да ја имплементирате следната процедура:

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

- Оваа процедура се повикува точно еднаш од страна на оценувачот.
- n : бројот на кутии.
- Процедурата треба да ја врати ознаката на кутијата што го содржи дијамантот, т.е. единствениот цел број d ($0 \leq d \leq n - 1$) таков што кутијата d содржи награда од тип 1.

Горната процедура може да прави повици до следната процедура:

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

- i : ознака на кутијата за која прашувате. Вредноста на i мора да биде помеѓу 0 и $n - 1$, вклучително.
- Оваа процедура ја враќа низата a со 2 елементи. Овде, $a[0]$ е бројот на поскапи награди во кутиите лево од кутијата i и $a[1]$ е бројот на поскапи награди во кутиите десно од кутијата i .

Пример

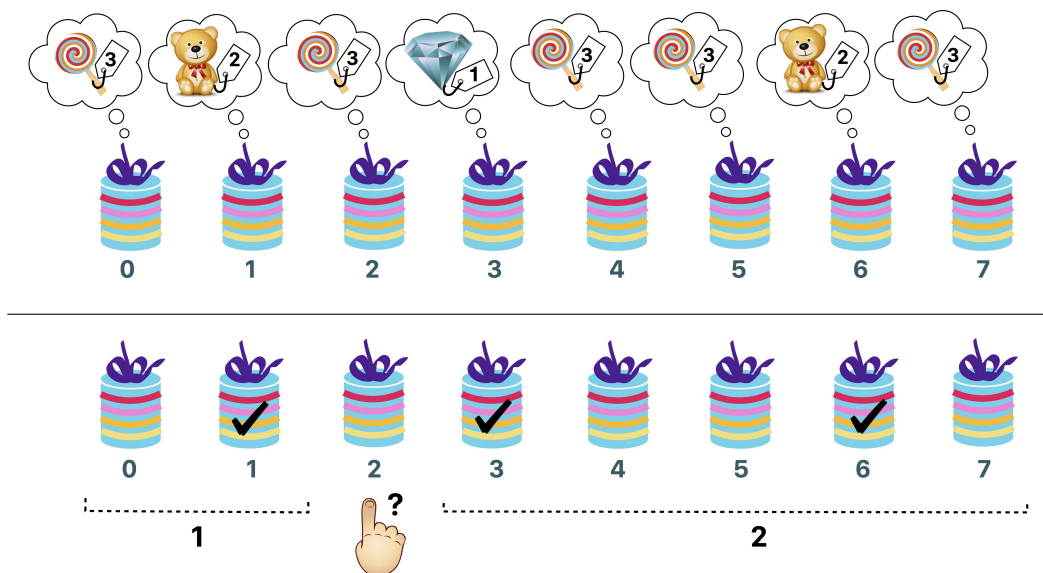
Оценувачот го прави следниот повик на процедура:

```
find_best(8)
```

Има вкупно $n = 8$ кутии. Нека претпоставите дека типовите на наградите се $[3, 2, 3, 1, 3, 3, 2, 3]$. Сите можни повици на процедурата `ask` и соодветните одговори се наведени подолу.

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

Во овој пример дијамантот се наоѓа во кутијата 3. Според тоа, процедурата `find_best` треба да врати 3.



Горната слика го илустрира овој пример. Горниот дел од сликата ги прикажува типовите на наградите во секоја од кутиите. Долниот дел од сликата е илустрација на прашањето `ask(2)`. Означените кутии содржат поскапи награди од наградата во кутија 2.

Ограничувања

- $3 \leq n \leq 200\,000$.
- Типот на награда во секоја кутија е помеѓу, 1 и v , вклучително.
- Има точно една награда од тип 1.
- За секое $2 \leq t \leq v$, ако има k награди од тип $t - 1$, тогаш има *строго повеќе* од k^2 награди од тип t .

Подзадачи и бодување

Во некои од тест случаите однесувањето на оценувачот е прилагодливо. Тоа значи дека во овие тест случаи оценувачот нема фиксна секвенца од награди. Наместо тоа, одговорите кои ги дава оценувачот може да зависат од прашањата што ги поставува вашето решение. Можете да бидете сигурни дека оценувачот одговара така што после секој одговор постои барем една низа од награди конзистентна со сите одговори дадени до тој момент.

1. (20 бодови) Има точно 1 дијамант и $n - 1$ лижавчиња (следствено, $v = 2$). Можете да ја повикате процедурата `ask` најмногу 10 000 пати.
2. (80 бодови) Не постојат дополнителни ограничувања.

Во подзадачата 2 можете да добиете парцијални бодови. Нека q е максималниот број на повици на процедурата `ask` за сите тест случаи во ова подзадача. Потоа, вашите бодови за оваа подзадача се пресметуваат според следната табела:

Број на прашања	Бодови
$10\,000 < q$	0 (во CMS се прикажува како "Wrong Answer")
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

Пример оценувач

Пример оценувачот не е прилагодлив. Овој оценувач само чита и користи фиксна низа p од типови на награди. За секое $0 \leq b \leq n - 1$, типот на награда во кутијата b е даден како $p[b]$. Пример оценувачот очекува влез во следниот формат:

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

Пример оценувачот печати една линија која ја содржи повратната вредност на `find_best` и бројот на повици на процедурата `ask`.