



Великий приз

Великий приз – це відоме ігрове телешоу, де вам поталанило дійти до фінального раунду. Ви стоїте перед рядом з n ящиків, що пронумеровано від 0 до $n - 1$ зліва направо. Кожен ящик містить приз, який не видно доки ящик не відкрито. Є $v \geq 2$ різних *типів* призів. Ці типи пронумеровано від 1 до v у порядку *спадання* їх вартості.

Приз типу 1 є найбільш коштовним – це діамант. Серед усіх ящиків є тільки один, у якому знаходиться діамант. Приз типу v є найбільш дешевим – це льодяник. Щоб зробити гру більш захоплюючою, кількість дешевих призів значно більша, ніж кількість більш коштовних. Більш точно, для всіх t таких що $2 \leq t \leq v$ ми знаємо наступне: якщо є k призів типу $t - 1$, то є *точно більше* ніж k^2 призів типу t .

Ваша мета – виграти діамант. У кінці гри вам потрібно буде відкрити ящик і ви отримаєте приз, що в ньому знаходиться. Перед тим, як обрати ящик, щоб відкрити його, ви можете задати Рамбоду, ведучому ігрового шоу, декілька запитань. Для кожного запитання ви обираєте деякий ящик i . Як відповідь, Рамбод надасть вам масив a з двох цілих чисел. Їх зміст такий:

- Серед ящиків ліворуч від ящика i є точно $a[0]$ ящиків, що містять більш коштовний приз, ніж ящик i .
- Серед ящиків праворуч від ящика i є точно $a[1]$ ящиків, що містять більш коштовний приз, ніж ящик i .

Наприклад, нехай $n = 8$. Для вашого запитання ви обираєте ящик $i = 2$. У відповідь Рамбод каже вам, що $a = [1, 2]$. Зміст цієї відповіді такий:

- Рівно один з ящиків 0 та 1 містить приз більш коштовний ніж приз у ящику 2.
 - Рівно два ящики з ящиків 3, 4, ..., 7 містять приз більш коштовний, ніж приз у ящику 2.
- Ваше завдання – знайти ящик з діамантом задавши малу кількість запитань.

Деталі реалізації

Ви маєте реалізувати наступну процедуру:

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

- Ця процедура викликається моделюванням перевірки один раз.
- n : кількість ящиків.
- Ця процедура має повертати номер ящика, що містить діамант, тобто одне ціле число d ($0 \leq d \leq n - 1$) таке, що ящик d містить приз типу 1.

Згадана процедура може викликати наступну процедуру:

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

- i : номер ящика, про який ви запитуєте. Значення i має бути між 0 та $n - 1$ включно.
- Процедура повертає масив a з 2 елементами. Тут $a[0]$ є кількістю більш коштовних призів ліворуч від ящика i та $a[1]$ є кількістю більш коштовних призів праворуч від ящика i .

Приклад

Модуль перевірки робить наступний виклик:

```
find_best(8)
```

Є $n = 8$ ящиків. Нехай типи призів будуть $[3, 2, 3, 1, 3, 3, 2, 3]$. Усі можливі виклики до процедури `ask` і відповідні результати перелічено нижче.

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

У цьому прикладі діамант знаходиться у ящику 3. Отже процедура `find_best` має повернути 3.

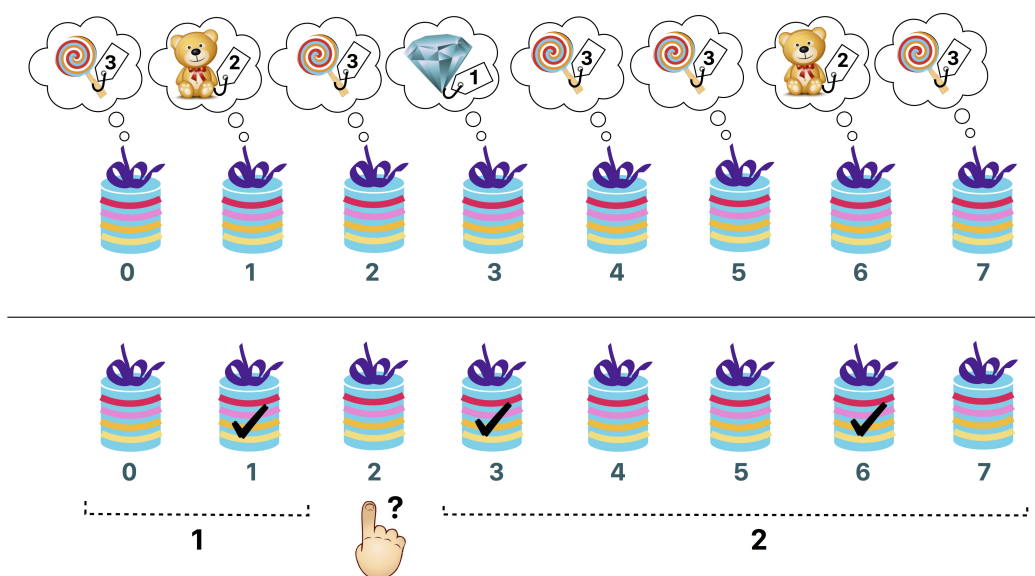


Рисунок вище ілюструє цей приклад. Верхня частина показує вартість призів у кожному з

ящиків. Нижня частина ілюструє запит `ask(2)`. Відмічені ящики містять більш коштовні призи, ніж приз у ящику 2.

Обмеження

- $3 \leq n \leq 200\,000$.
- Тип призу у кожному з ящиків знаходиться між 1 та v включно.
- Є рівно один приз типу 1.
- Для всіх $2 \leq t \leq v$, якщо є k призів типу $t - 1$, є *точно* більше ніж k^2 призів типу t .

Підзадачі та оцінювання

У деяких тестах поведінка модуля перевірки є адаптивною. Це означає, що у деяких тестах модуль перевірки не має фіксованої послдовності призів. Натомість, відповіді, які дає модуль перевірки, залежать від запитань, що задає ваш розв'язок. Гарантується, що модуль перевірки відповідає таким чином, що після кожної відповіді є принаймні одна послідовність призів, яка підходить під відповіді, що вже дав модуль перевірки.

1. (20 балів) Є точно один 1 діамант та $n - 1$ льодяників (отже, $v = 2$). Ви можете викликати процедуру `ask` не більше 10 000 разів.
2. (80 балів) Немає додаткових обмежень.

У підзадачі 2 ви можете отримати частину балів. Нехай q буде максимальною кількістю викликів процедури `ask` серед усіх тестів у цій підзадачі. Тоді кількість балів обчислюється за таблицею:

Запитання	Бали
$10\,000 < q$	0 (позначається в CMS як 'Wrong Answer')
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

Приклад модуля перевірки

Модуль перевірки з прикладу не є адаптивним. Натомість, він просто читає та використовує фіксований масив p типів призів. Для всіх $0 \leq b \leq n - 1$ тип призу у ящику b задається як $p[b]$. Цей модуль перевірки очікує вхідні дані у наступному форматі:

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

Модуль перевірки з прикладу друкує один рядок, що містить значення, повернуте `find_best`

та кількість викликів процедури `ask`.