



## The Big Prize

The Big Prize là một trò chơi truyền hình nổi tiếng. Bạn là một người chơi may mắn lọt vào vòng chung kết. Bạn đứng trước một dãy  $n$  hộp, được gán nhãn từ 0 đến  $n - 1$  theo chiều từ trái qua phải. Mỗi hộp chứa một giải thưởng mà không được biết cho đến khi mở hộp. Có  $v \geq 2$  loại giải thưởng khác nhau. Các loại giải thưởng được đánh số từ 1 to  $v$  theo thứ tự *giảm dần* về mặt giá trị.

Giải thưởng loại 1 có giá trị lớn nhất: một viên kim cương. Có đúng một viên kim cương trong các hộp. Giải thưởng loại  $v$  có giá trị nhỏ nhất: một cây kẹo mút. Để trò chơi thêm phần hấp dẫn, giải thưởng giá trị nhỏ hơn có số lượng nhiều hơn hẳn so với giải thưởng giá trị lớn hơn. Cụ thể, với mọi  $t$  mà  $2 \leq t \leq v$  ta biết tính chất sau: nếu có  $k$  giải thưởng loại  $t - 1$ , thì có *lớn hơn hẳn*  $k^2$  giải thưởng loại  $t$ .

Mục tiêu của bạn là giành được viên kim cương. Cuối trò chơi, bạn sẽ phải mở một hộp và bạn sẽ nhận giải thưởng trong hộp đó. Trước khi phải chọn chiếc hộp để mở, bạn được hỏi Rambod, người dẫn chương trình, một vài câu hỏi. Với mỗi câu hỏi, bạn chọn một hộp  $i$  nào đó. Để trả lời bạn, Rambod sẽ đưa bạn một mảng  $a$  gồm hai số nguyên. Ý nghĩa của chúng như sau:

- Trong tất cả các hộp nằm ở bên trái hộp  $i$  có đúng  $a[0]$  hộp chứa giải thưởng có giá trị lớn hơn giá trị giải thưởng nằm trong hộp  $i$ .
- Trong tất cả các hộp nằm ở bên phải hộp  $i$  có đúng  $a[1]$  hộp chứa giải thưởng có giá trị lớn hơn giá trị giải thưởng nằm trong hộp  $i$ .

Ví dụ, giả sử  $n = 8$ . Với câu hỏi của bạn, bạn chọn hộp  $i = 2$ . Đáp lại, Rambod đưa bạn câu trả lời  $a = [1, 2]$ . Ý nghĩa của câu trả lời này là:

- Có đúng một hộp trong số các hộp 0 và 1 chứa giải thưởng có giá trị lớn hơn giá trị giải thưởng trong hộp 2.
- Có đúng hai hộp trong số các hộp 3, 4, ..., 7 chứa giải thưởng có giá trị lớn hơn giá trị giải thưởng trong hộp 2.

Nhiệm vụ của bạn là tìm hộp chứa viên kim cương bằng cách hỏi một số ít các câu hỏi.

## Chi tiết cài đặt

Bạn phải cài đặt thủ tục sau:

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

- Thủ tục này được gọi đúng một lần bởi trình chấm.

- $n$ : số lượng các hộp.
- Thủ tục này phải trả về nhãn của hộp chứa viên kim cương, chính xác là số nguyên duy nhất  $d$  ( $0 \leq d \leq n - 1$ ) mà hộp  $d$  chứa giải thưởng loại 1.

Thủ tục trên có thể gọi thủ tục sau đây nhiều lần:

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

- $i$ : nhãn của hộp mà bạn chọn để hỏi. Giá trị của  $i$  phải nằm giữa 0 và  $n - 1$ , tính cả hai đầu mút.
- Thủ tục này trả về mảng  $a$  chứa 2 phần tử. Ở đây,  $a[0]$  là số lượng các giải thưởng có giá trị lớn hơn nằm trong các hộp ở bên trái hộp  $i$  và  $a[1]$  là số lượng các giải thưởng có giá trị lớn hơn nằm trong các hộp ở bên phải hộp  $i$ .

## Ví dụ

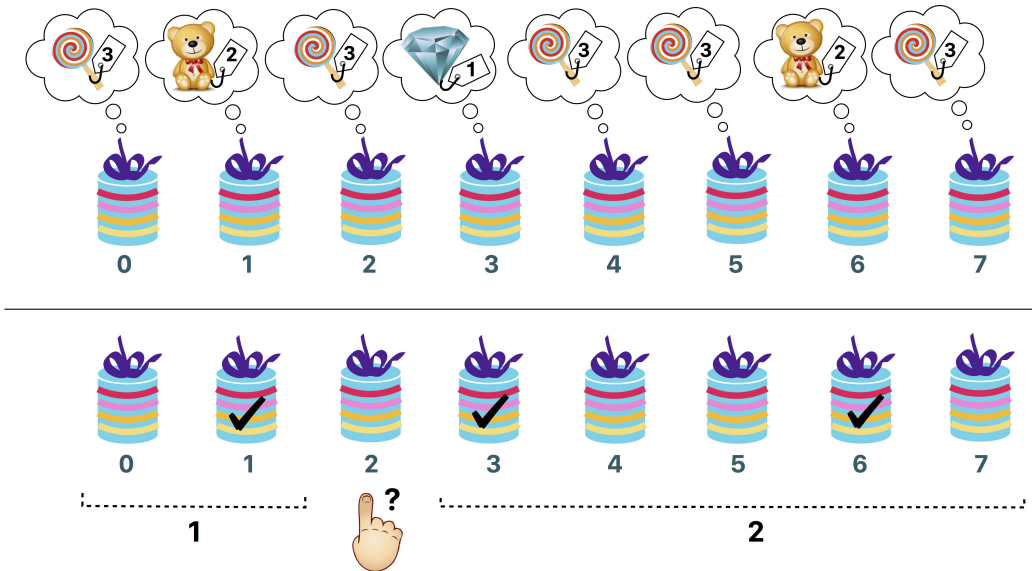
Trình chấm gọi thủ tục sau:

```
find_best(8)
```

Có  $n = 8$  hộp. Giả sử các loại giải thưởng là  $[3, 2, 3, 1, 3, 3, 2, 3]$ . Tất cả các cách có thể gọi thủ tục `ask` và các giá trị trả về tương ứng được liệt kê dưới đây.

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

Trong ví dụ này, viên kim cương nằm trong hộp 3. Cho nên thủ tục `find_best` phải trả về 3.



Hình vẽ trên mô tả ví dụ này. Phần trên của hình vẽ hiển thị loại của giải thưởng trong từng hộp. Phần dưới của hình vẽ mô tả câu hỏi `ask(2)`. Các hộp được đánh dấu chứa giải thưởng có giá trị lớn hơn giá trị giải thưởng trong hộp 2.

## Ràng buộc

- $3 \leq n \leq 200\,000$ .
- Loại của giải thưởng trong mỗi hộp nằm giữa 1 và  $v$ , tính cả hai đầu mút.
- Có đúng một giải thưởng loại 1.
- Với mọi  $2 \leq t \leq v$ , nếu có  $k$  giải thưởng loại  $t - 1$ , thì có lớn hơn *hẳn*  $k^2$  giải thưởng loại  $t$ .

## Subtasks and scoring

Trong một vài bộ dữ liệu, hành vi của trình chấm là thích nghi. Có nghĩa là, đối với các bộ dữ liệu này, trình chấm không có một chuỗi cố định các giải thưởng. Thay vào đó, các câu trả lời của trình chấm có thể phụ thuộc vào các câu hỏi mà lời giải của bạn hỏi. Bạn được đảm bảo rằng, trình chấm sẽ trả lời theo cách mà sau mỗi câu trả lời, có ít nhất một dãy các giải thưởng phù hợp với tất cả các câu trả lời đã được đưa ra.

1. (20 điểm) Có đúng 1 viên kim cương và  $n - 1$  cây kẹo mút (nên  $v = 2$ ). Bạn có thể gọi thủ tục `ask` nhiều nhất 10 000 lần.
2. (80 điểm) Không có thêm ràng buộc nào.

Trong subtask 2 bạn có thể nhận điểm thành phần. Gọi  $q$  là số lần gọi thủ tục `ask` lớn nhất trong tất cả các bộ dữ liệu của subtask này. Khi đó, điểm của bạn cho subtask này được tính theo bảng sau:

| Các câu hỏi             | Điểm   |
|-------------------------|--|
| $10\,000 < q$           | 0 (được thông báo trong CMS là 'Wrong Answer') |
| $6000 < q \leq 10\,000$ | 70   |
| $5000 < q \leq 6000$    | $80 - (q - 5000)/100$                          |
| $q \leq 5000$           | 80   |

## Trình chấm mẫu

Trình chấm mẫu là không thích nghi. Thay vào đó, nó chỉ đọc và sử dụng một mảng cố định  $p$  chứa các loại giải thưởng. Với mọi  $0 \leq b \leq n - 1$ , loại của giải thưởng trong hộp  $b$  được cho là  $p[b]$ . Trình chấm mẫu nhận dữ liệu đầu vào với với khuôn dạng sau:

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

Trình chấm mẫu in ra trên một dòng chứa giá trị trả về của `find_best` và số lần gọi thủ tục `ask`.