



The Big Prize大獎

大獎是一個家喻戶曉的 TV 遊戲節目。這次你很幸運地進入到最後一輪。已知編號從 0 到 $n - 1$ 的 n 個盒子從左到右排成一行，你站在這排盒子的前面。每個盒子裡面放有一個獎品，必須打開盒子才能看到是什麼獎品。已知有 $v \geq 2$ 種不同類型的獎品。這 v 種類型按照獎品價值的降冪從 1 到 v 排列。

類型 1 的獎品是一塊鑽石，價值最貴。所有盒子中恰好只有一塊鑽石。類型 v 的獎品是一塊棒棒糖，價值最便宜。為使得遊戲更加激動人心，相對便宜的獎品數量遠比價值昂貴的獎品數量更多。更為形式化地可描述為，對於滿足 $2 \leq t \leq v$ 的所有 t ，我們已知如下事實：如果類型 $t - 1$ 的獎品有 k 個，那麼類型 t 的獎品將嚴格多於 k^2 個。

你的目標是贏得那塊鑽石。在遊戲結束時，你必須打開一個盒子並獲取盒子內的獎品。在選擇要打開的盒子之前，你可以向節目主持人 Rambod 提一些問題。對於每一個問題，你要選擇某個 i 號盒子。Rambod 將給你一個包含兩個整數的陣列 a 作為問題的回答。這個陣列的意義如下：

- 在 i 號盒子左面的盒子中，剛好有 $a[0]$ 個盒子，每個盒子裡面包含一個獎品，其價值比 i 號盒子中的獎品價值更貴。
- 在 i 號盒子右面的盒子中，剛好有 $a[1]$ 個盒子，每個盒子裡面包含一個獎品，其價值比 i 號盒子中的獎品價值更貴。

例如，假設 $n = 8$ 。對於你的問題，你選擇 $i = 2$ 號盒子。Rambod 用 $a = [1, 2]$ 來回答你。這一回答的意義是：

- 0 號盒子和 1 號盒子中恰好有一個盒子包含的獎品比 2 號盒子中的獎品更貴。
- 在 $3, 4, \dots, 7$ 號盒子中恰好有 2 個盒子包含的獎品比 2 號盒子中的獎品更貴。

你的任務就是通過詢問少量的問題以找出包含那塊鑽石的那個盒子。

實現細節

你應當實現下列過程(函數段)：

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

- 此函數只被評測系統呼叫僅一次
- n ：盒子的數目。
- 你實現的這個過程應該返回包含鑽石的盒子編號，即，那個唯一的整數 d ($0 \leq d \leq n - 1$) 滿足 d 號盒子中放有類型 1 的獎品。

上述過程可以調用下列過程：

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

- i : 你在詢問時選擇的盒子編號。 i 的數值必須介於 0 和 $n - 1$ 之間（含）。
- 這個過程返回包含 2 個元素的陣列 a 。其中， $a[0]$ 是在 i 號盒子左面的盒子中，比 i 號盒子的獎品價值更貴的獎品數目。而 $a[1]$ 則是在 i 號盒子右面的盒子中，比 i 號盒子的獎品價值更貴的獎品數目。

例子

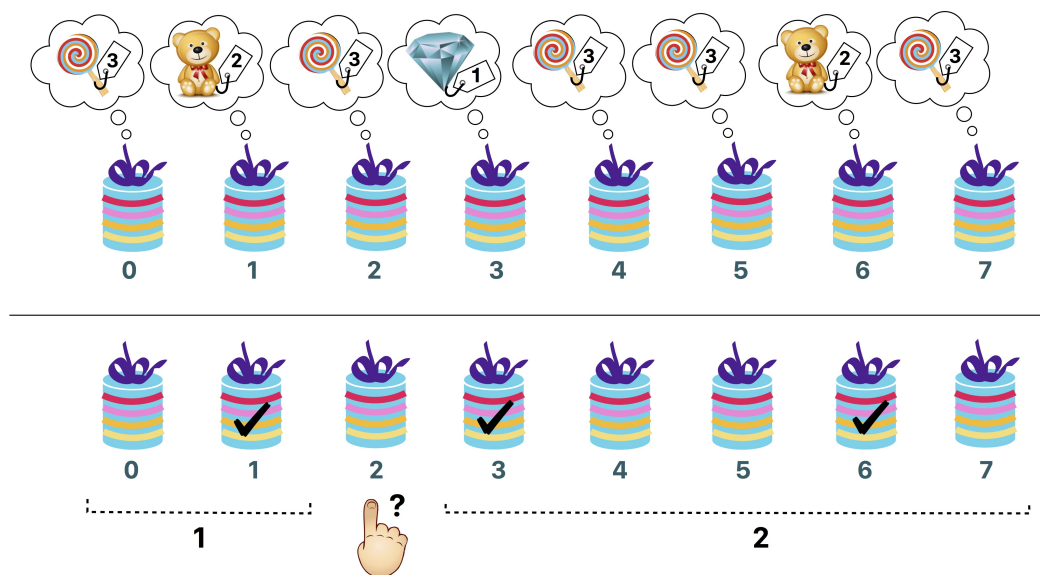
評測工具將做下列程序呼叫：

```
find_best(8)
```

有 $n = 8$ 個盒子。假定獎品類型為 $[3, 2, 3, 1, 3, 3, 2, 3]$ 。對過程 `ask` 的所有可能的調用以及相應的返回值列出如下：

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

在這個例子中，鑽石放在 3 號盒子裡。所以過程 `find_best` 應該返回 3 。



上圖展示了這個例子。圖中最上面一行給出了每個盒子中獎品的類型。圖中的下半部分展示了詢問 `ask(2)`。做了標記的盒子中包含有比 2 號盒子的獎品價值更貴的獎品。

限制

- $3 \leq n \leq 200\,000$.
- 每個盒子中獎品的類型介於 1 和 v 之間（含）。
- 類型 1 的獎品恰有一個。
- 對於所有 $2 \leq t \leq v$ ，如果類型 $t - 1$ 的獎品有 k 個，那麼類型 t 的獎品將嚴格多於 k^2 個。

子任務與評分

在某些測試資料中，評測工具的行為是自我調整的。這意味著在這些測試資料中評測工具並沒有一個固定的獎品序列。取而代之的是，由評測工具給出的答案可能依賴於你的求解過程詢問的問題。評測工具的回答可以保證，在每次回答之後，至少有一個獎品序列與到目前為止給出的所有答案是一致的。

1. (20 分) 恰好有 1 個鑽石和 $n - 1$ 個棒棒糖（所以， $v = 2$ ）。你可以調用過程 `ask` 最多 10 000 次。
2. (80 分) 沒有附加限制。

在子任務 2 中你可以獲得部分分。令 q 是在這個子任務中過程 `ask` 對所有測試資料的最大調用次數，那麼你在這個子任務的得分將按照下表計算：

問題	得分
$10\,000 < q$	0（在CMS中報告為 'Wrong Answer'）
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

評測工具示例

評測工具的這個示例不是自我調整的。取而代之的是，它只是讀取並使用一個固定的獎品類型的陣列 p 。對於所有的 $0 \leq b \leq n - 1$ ， b 號盒子中的獎品類型將由 $p[b]$ 給出。評測工具示例期待按下列格式輸入

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

評測工具示例輸出單獨一行，上面包含 `find_best` 的返回值以及調用過程 `ask` 的次數。