



## جایزه بزرگ

«جایزه بزرگ» یک مسابقه تلویزیونی مشهور است. شما یک شرکت‌کننده خوش‌شانس هستید که به دور نهایی راه یافته‌اید. شما در مقابل یک سطر از  $n$  جعبه ایستاده‌اید که از چپ به راست با 0 تا  $n - 1$  برچسب‌گذاری شده‌اند. هر جعبه جایزه‌ای دارد که تا وقتی جعبه باز نشده است نمی‌توان آن را مشاهده کرد.  $v \geq 2$  نوع مختلف جایزه وجود دارد. این نوع جایزه‌ها از 1 تا  $v$  به ترتیب نزولی ارزششان شماره‌گذاری شده‌اند.

جایزه از نوع 1 گران‌ترین جایزه است: الماس. دقیقاً یک الماس در میان جعبه‌ها وجود دارد. جایزه از نوع  $v$  ارزان‌ترین جایزه است: آب‌نبات چوبی. برای اینکه بازی هیجان‌انگیزتر شود، تعداد جایزه‌های ارزان خیلی بیشتر از تعداد جایزه‌های گران‌تر است. به صورت دقیق‌تر، برای هر  $t$  که  $2 \leq t \leq v$  می‌دانیم که شرط روبرو برقرار است: اگر  $k$  جایزه از نوع  $t - 1$  داشته باشیم، تعداد جایزه‌های با نوع  $t$  / کید/ بیشتر از  $k^2$  است.

هدف شما این است که الماس را ببرید. در پایان بازی شما باید یک جعبه را انتخاب کنید و برنده جایزه داخل آن جعبه خواهید شد. قبل از انتخاب این‌که چه جعبه‌ای را باز کنید، می‌توانید از رامبد، مجری این بازی تلویزیونی، چند سوال بپرسید. برای هر سوال، شما یک جعبه مانند  $i$  را انتخاب می‌کنید. در جواب، رامبد به شما آرایه  $a$  را خواهد داد که شامل دو عدد است. معنی آن‌ها به صورت زیر است:

- بین همه جعبه‌های سمت چپ جعبه  $i$ ، دقیقاً  $a[0]$  جعبه جایزه‌ای گران‌تر از جعبه  $i$  دارند.
- بین همه جعبه‌های سمت راست جعبه  $i$ ، دقیقاً  $a[1]$  جعبه جایزه‌ای گران‌تر از جعبه  $i$  دارند.

برای نمونه، فرض کنید  $n = 8$ . شما برای سوال خود جعبه  $i = 2$  را انتخاب می‌کنید. رامبد در پاسخ، به شما می‌گوید که  $a = [1, 2]$ . معنی این پاسخ این است که:

- دقیقاً یک جعبه از جعبه‌های 0 و 1 جایزه‌ای گران‌تر از جعبه شماره 2 دارد.
- دقیقاً دو جعبه از جعبه‌های 3، 4، ...، 7 جایزه‌ای گران‌تر از جعبه 2 دارند.

شما باید جعبه حاوی الماس را با تعداد کمی پرسش بیابید.

## جزئیات پیاده‌سازی

شما باید تابع زیر را پیاده‌سازی کنید:

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

- این تابع دقیقاً یکبار توسط ارزیاب (grader) فراخوانی می‌شود.
- $n$ : تعداد جعبه‌ها.
- این تابع باید برچسب جعبه‌ای که حاوی الماس است را برگرداند. یعنی، عدد یکتای  $d$  ( $0 \leq d \leq n - 1$ ) که جعبه  $d$  حاوی جایزه از نوع 1 است.

تابع بالا می‌تواند تابع زیر را فراخوانی کند:

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

- $i$ : برچسب جعبه‌ای که برای پرسش انتخاب کرده‌اید. مقدار  $i$  باید بین 0 و  $n - 1$  (شامل 0 و  $n - 1$ ) باشد.
- این تابع آرایه  $a$  شامل 2 عضو را بر می‌گرداند. اینجا،  $a[0]$  تعداد جعبه‌های گران‌تر از جعبه  $i$  در جعبه‌های سمت چپ جعبه  $i$  است و  $a[1]$  تعداد جعبه‌های گران‌تر از جعبه  $i$  در جعبه‌های سمت راست جعبه  $i$  است.

## مثال

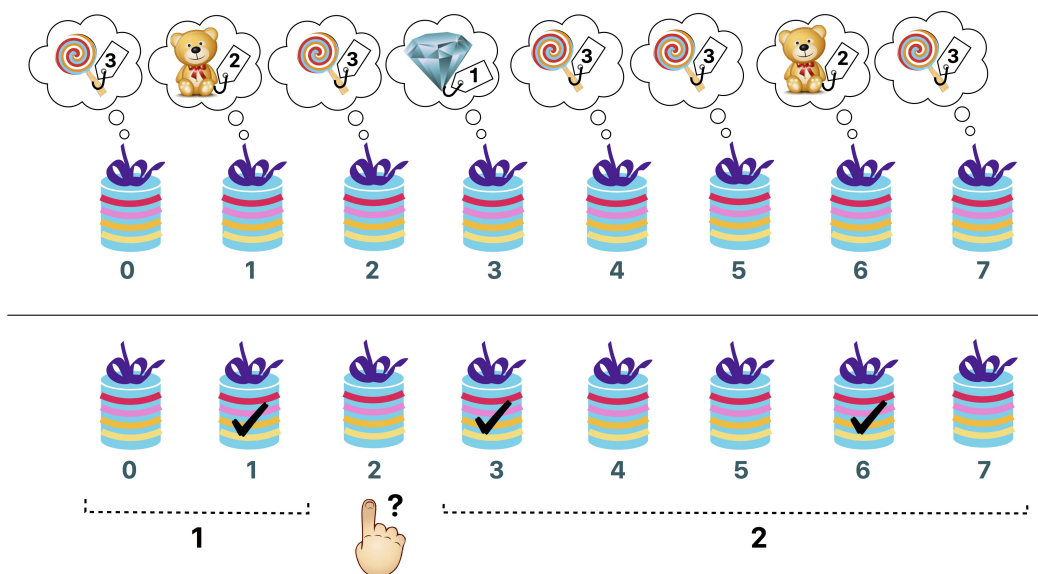
ارزیاب، تابع زیر را فراخوانی می‌کند.

```
find_best(8)
```

تعداد جعبه‌ها  $n = 8$  است. فرض کنید که نوع جایزه‌ها  $[3, 2, 3, 1, 3, 3, 2, 3]$  باشد. همه فراخوانی‌های ممکن تابع  $ask$  و مقدار برگشتی آن‌ها در زیر لیست شده است:

- $ask(0)$  آرایه  $[0, 3]$  را بر می‌گرداند.
- $ask(1)$  آرایه  $[0, 1]$  را بر می‌گرداند.
- $ask(2)$  آرایه  $[1, 2]$  را بر می‌گرداند.
- $ask(3)$  آرایه  $[0, 0]$  را بر می‌گرداند.
- $ask(4)$  آرایه  $[2, 1]$  را بر می‌گرداند.
- $ask(5)$  آرایه  $[2, 1]$  را بر می‌گرداند.
- $ask(6)$  آرایه  $[1, 0]$  را بر می‌گرداند.
- $ask(7)$  آرایه  $[3, 0]$  را بر می‌گرداند.

در این مثال، الماس در جعبه 3 است. بنابراین تابع  $find\_best$  باید عدد 3 را برگرداند.



شکل بالا همین مثال را نمایش می‌دهد. بخش بالا ارزش جایزه‌های هر جعبه را نمایش می‌دهد. بخش پایین

پرسش (2) ask را نمایش می‌دهد. جعبه‌های علامت‌خورده حاوی جایزه‌های گران‌تر از جایزه درون جعبه 2 هستند.

## محدودیت‌ها

- $3 \leq n \leq 200\,000$
- نوع جایزه در هر جعبه عددی بین 1 تا  $v$  (شامل 1 و  $v$ ) است.
- دقیقاً یک جایزه نوع 1 وجود دارد.
- برای هر  $2 \leq t \leq v$ ، اگر  $k$  جایزه با نوع  $t - 1$  وجود داشته‌باشد، تعداد جایزه‌های با نوع  $t$  / کید/ بیشتر از  $k^2$  است.

## زیرمسئله‌ها و امتیازدهی

در بعضی از آزمایش‌ها (test cases) رفتار ارزیاب تطبیقی (adaptive) است. یعنی در این آزمایش‌ها ارزیاب دنباله معینی از جایزه‌ها ندارد. در عوض، پاسخ‌های ارزیاب ممکن است به سوالات پرسیده شده توسط راه حل شما بستگی داشته باشد. تضمین می‌شود که ارزیاب به گونه‌ای پاسخ می‌دهد که پس از هر پاسخ، حداقل یک دنباله از جایزه‌ها وجود داشته باشد که با پاسخ‌های تاکنون داده شده مطابقت داشته باشد.

1. (۲۰ امتیاز) دقیقاً 1 الماس و  $n - 1$  آب‌نبات چوبی داریم (پس،  $v = 2$ ). شما می‌توانید تابع ask را حداکثر 10 000 بار فراخوانی کنید.
2. (۸۰ نمره) بدون محدودیت اضافی.

در زیرمسئله ۲ شما می‌توانید نمره جزئی (partial score) دریافت کنید.  $q$  را بیشترین تعداد فراخوانی تابع ask در میان همه آزمایش‌های (test cases) این زیرمسئله در نظر بگیرید. سپس، امتیاز شما برای این زیرمسئله مطابق جدول زیر محاسبه می‌شود:

تعداد سوالات	امتیاز
$10\,000 < q$	0 (در CMS با عبارت 'Wrong Answer' گزارش می‌شود)
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

## ارزیاب نمونه

ارزیاب نمونه تطبیقی نیست. در عوض، آرایه معین  $p$  از نوع جایزه‌ها را از ورودی می‌خواند و استفاده می‌کند. برای هر  $0 \leq b \leq n - 1$ ، نوع جایزه در جعبه  $b$  برابر با  $p[b]$  است. ارزیاب نمونه ورودی را در قالب زیر می‌خواند:

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

ارزیاب نمونه یک سطر شامل مقدار برگشتی find\_best و تعداد فراخوانی تابع ask را چاپ می‌کند.