



## Hadiah Akbar

Hadiah Akbar adalah sebuah acara permainan televisi terkenal. Anda adalah peserta beruntung yang maju ke babak final. Anda berdiri di depan sebuah baris berisi  $n$  kotak, dinomori dari 0 hingga  $n - 1$  dari kiri ke kanan. Setiap kotak berisi sebuah hadiah yang tidak dapat dilihat sampai kotak tersebut dibuka. Terdapat  $v \geq 2$  tipe hadiah berbeda. Tipe-tipe tersebut dinomori dari 1 hingga  $v$  dengan urutan nilai *menurun*.

Hadiah bertipe 1 adalah hadiah yang paling mahal: sebuah berlian. Terdapat tepat satu berlian pada kotak-kotak tersebut. Hadiah bertipe  $v$  adalah hadiah yang paling murah: sebuah permen *lollipop*. Untuk membuat permainan lebih menarik, banyaknya hadiah yang lebih murah jauh lebih besar daripada banyaknya hadiah yang lebih mahal. Lebih jelasnya, untuk setiap  $t$  sedemikian sehingga  $2 \leq t \leq v$  kita mengetahui hal berikut: jika terdapat  $k$  hadiah bertipe  $t - 1$ , terdapat *lebih* dari  $k^2$  hadiah bertipe  $t$ .

Tujuan Anda adalah mendapatkan berlian tersebut. Pada akhir permainan, Anda harus membuka sebuah kotak dan Anda akan mendapatkan hadiah di dalam kotak tersebut. Sebelum memilih kotak yang harus dibuka, Anda dapat menanyakan beberapa pertanyaan kepada Rambod, pembawa acara dari permainan tersebut. Untuk setiap pertanyaan, Anda memilih sebuah kotak  $i$ . Sebagai jawabannya, Rambod akan memberi Anda sebuah *array*  $a$  berisi dua bilangan bulat. Artinya adalah sebagai berikut:

- Di antara kotak-kotak di sebelah kiri kotak  $i$ , terdapat tepat  $a[0]$  kotak yang berisi sebuah hadiah yang lebih mahal daripada hadiah pada kotak  $i$ .
- Di antara kotak-kotak di sebelah kanan kotak  $i$ , terdapat tepat  $a[1]$  kotak yang berisi sebuah hadiah yang lebih mahal daripada hadiah pada kotak  $i$ .

Sebagai contoh, anggap  $n = 8$ . Sebagai pertanyaan Anda, Anda memilih kotak  $i = 2$ . Sebagai tanggapannya, Rambod memberi tahu Anda bahwa  $a = [1, 2]$ . Arti dari tanggapan ini adalah:

- Tepat satu dari kotak-kotak 0 dan 1 berisi sebuah hadiah yang lebih mahal daripada hadiah pada kotak 2.
- Tepat dua dari kotak-kotak 3, 4,  $\dots$ , 7 berisi sebuah hadiah yang lebih mahal daripada hadiah pada kotak 2.

Tugas Anda adalah mencari kotak yang berisi berlian dengan menanyakan sedikit pertanyaan.

## Detil implementasi

Anda harus mengimplementasikan prosedur berikut ini:

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

- Prosedur ini dipanggil tepat sekali oleh *grader*
- $n$ : banyaknya kotak.
- Prosedur tersebut harus mengembalikan nomor dari kotak yang berisi berlian, yakni, bilangan bulat unik  $d$  ( $0 \leq d \leq n - 1$ ) sedemikian sehingga kotak  $d$  berisi sebuah hadiah bertipe 1.

Prosedur di atas dapat memanggil prosedur berikut:

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

- $i$ : nomor dari kotak yang Anda pilih untuk ditanyakan. Nilai dari  $i$  harus di antara 0 dan  $n - 1$ , inklusif.
- Prosedur ini mengembalikan *array*  $a$  berisi 2 elemen. Di sini,  $a[0]$  adalah banyaknya hadiah yang lebih mahal pada kotak-kotak di sebelah kiri kotak  $i$  dan  $a[1]$  adalah banyak hadiah yang lebih mahal pada kotak-kotak di sebelah kanan kotak  $i$ .

## Contoh

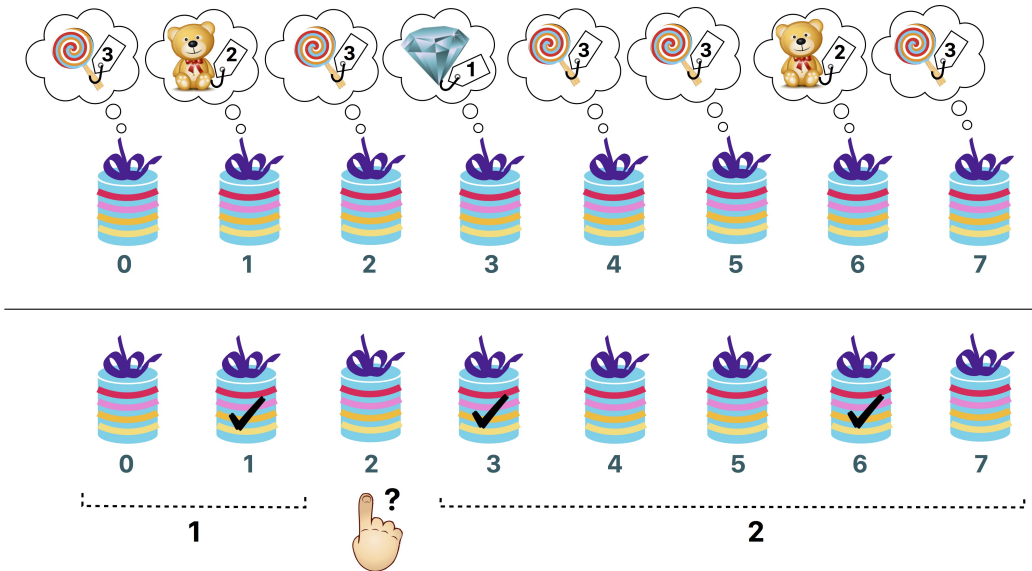
*Grader* melakukan pemanggilan prosedur berikut:

```
find_best(8)
```

Terdapat  $n = 8$  kotak. Anggap tipe-tipe hadiah adalah  $[3, 2, 3, 1, 3, 3, 2, 3]$ . Seluruh pemanggilan prosedur `ask` yang mungkin dan nilai kembalian yang sesuai terdaftar di bawah ini.

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

Pada contoh ini, berlian tersebut berada pada kotak 3. Maka prosedur `find_best` harus mengembalikan 3.



Gambar di atas mengilustrasikan contoh ini. Bagian atas menunjukkan tipe dari hadiah pada setiap kotak. Bagian bawah mengilustrasikan pertanyaan  $\text{ask}(2)$ . Kotak yang ditandai berisi hadiah yang lebih mahal daripada hadiah pada kotak 2.

## Batasan

- $3 \leq n \leq 200\,000$ .
- Tipe dari hadiah pada setiap kotak di antara 1 dan  $v$ , inklusif.
- Terdapat tepat satu hadiah bertipe 1.
- Untuk setiap  $2 \leq t \leq v$ , jika terdapat  $k$  hadiah bertipe  $t - 1$ , terdapat lebih dari  $k^2$  hadiah bertipe  $t$ .

## Subsoal dan penilaian

Pada beberapa kasus uji, perilaku *grader* adalah adaptif. Ini berarti dalam kasus-kasus uji ini, *grader* tidak memiliki sebuah barisan hadiah yang tetap. Akan tetapi, jawaban-jawaban yang diberikan oleh *grader* dapat bergantung pada pertanyaan-pertanyaan yang ditanyakan oleh solusi Anda. Dijamin bahwa *grader* menjawab sedemikian sehingga setelah setiap jawaban, terdapat setidaknya satu barisan hadiah yang konsisten dengan seluruh jawaban yang diberikan sejauh ini.

1. (20 poin) Terdapat tepat 1 berlian dan  $n - 1$  permen *lollipop* (sehingga,  $v = 2$ ). Anda dapat memanggil prosedur  $\text{ask}$  paling banyak 10 000 kali.
2. (80 poin) Tidak ada batasan tambahan.

Pada subsoal 2 Anda dapat mendapatkan sebuah nilai parsial. Anggap  $q$  adalah banyaknya pemanggilan prosedur  $\text{ask}$  maksimum di antara seluruh kasus uji pada subsoal ini. Maka, nilai Anda untuk subsoal ini dihitung berdasarkan tabel berikut ini:

Pertanyaan	Nilai
$10\,000 < q$	0 (dilaporkan di CMS sebagai 'Wrong Answer')
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

## Grader contoh

*Grader* contoh tidak adaptif. Melainkan, *grader* contoh hanya membaca dan menggunakan sebuah *array* tetap  $p$  berisi tipe-tipe hadiah. Untuk setiap  $0 \leq b \leq n - 1$ , tipe dari hadiah pada kotak  $b$  diberikan sebagai  $p[b]$ . *Grader* contoh mengharapkan masukan dengan format berikut:

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

*Grader* contoh mencetak sebuah baris berisi nilai kembalian dari `find_best` dan banyaknya pemanggilan prosedur `ask`.