



Büyük Ödül (The Big Prize)

Büyük ödül herkesin bildiği bir TV yarışmasıdır. Siz, bu yarışmada finale kalan şanslı yarışmacısınız. Soldan sağa doğru 0'dan $n - 1$ 'e kadar numaralanmış n tane kutunun önündesiniz. Her kutuda bir ödül vardır, ancak bir kutudaki ödülü o kutu açılana kadar göremezsiniz. Toplam $v \geq 2$ farklı ödül türü vardır. Türler 1'den v 'ye kadar *azalan* değerde numaralandırılmıştır.

Tür 1 ödülü en pahalı ödüdür: Elmas. Kutularda tam olarak bir tane elmas vardır. Tür v ödülü ise en ucuz ödüdür: Lolipop. Oyunu daha çekici yapmak için, ucuz ödüllerin sayısı pahalı ödüllerin sayısından oldukça fazla yapılmıştır. Daha açık olarak, $2 \leq t \leq v$ aralığındaki tüm t 'ler için şunu biliyoruz: Eğer $t - 1$ türünde k tane ödül varsa, t türünde k^2 'dan *daha fazla* ödül vardır.

Hedefiniz elması kazanmaktır. Oyunun sonunda bir kutuyu açmak zorundasınız ve o kutudaki ödülü kazanacaksınız. Kutunuzu açtırmadan önce, yarışmanın sunucusu olan Rambod'a bazı soruları sorma hakkınız vardır. Her bir soru için, i kutusunu seçiyorsunuz. Rambod cevap olarak, iki tam sayı içeren a dizisini size veriyor. Bu dizinin anlamı şu şekildedir:

- i kutusunun solundaki kutular içinde, i 'den daha pahalı olan tam olarak $a[0]$ tane kutu vardır.
- i kutusunun sağındaki kutular içinde, i 'den daha pahalı olan tam olarak $a[1]$ tane kutu vardır.

Örnek olarak, $n = 8$ olsun. Sorunuzda $i = 2$ kutusunu seçmiş olun. Rambod cevaben $a = [1, 2]$ 'yi söylüyor. Rambod'un cevabının manası şudur:

- 0 ve 1 kutularından tam olarak bir tanesi 2 kutusundan daha pahalı ödül barındırmaktadır.
- 3, 4, ..., 7 kutularından tam olarak iki tanesi 2 kutusundan daha pahalı ödül barındırmaktadır.

Göreviniz, az sayıda soru sorarak elması içeren kutuyu bulmaktır.

Gerçekleştirim detayları

Aşağıdaki prosedürü kodlamalısınız:

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

- bu prosedür değerlendirici tarafından tam olarak bir kez çağrılır.
- n : kutuların sayısı.
- prosedür, elması içeren kutunun etiketini dönmelidir. Yani, tek (eşsiz) bir tam sayı d ($0 \leq d \leq n - 1$) öyleki d kutusu tür 1 ödülünü içermektedir.

Yukarıdaki prosedür aşağıdaki prosedürü çağırabilir:

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

- i : Soru soracağınız kutunun etiketi. i 'nin değeri 0 ile $n - 1$ arasında olmalıdır (sınırlar dahildir).
- Bu prosedür 2 elemanlı a dizisini döner. Burada, $a[0]$, i kutusunun solunda olan kutulardaki daha pahalı olan ödüllerin sayısıdır. $a[1]$ ise i kutusunun sağında olan kutulardaki daha pahalı olan ödüllerin sayısıdır.

Örnek

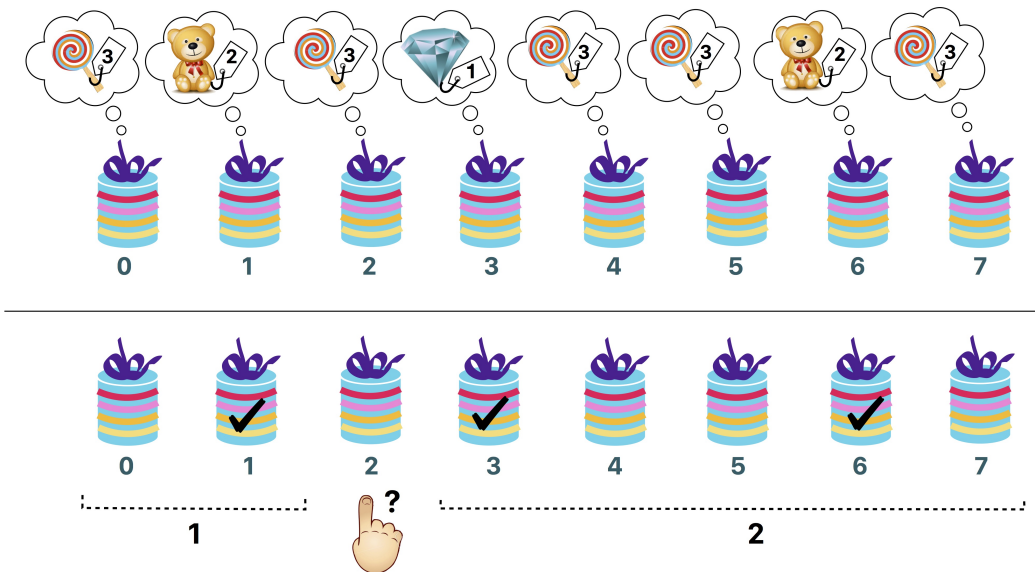
Değerlendirici (grader) aşağıdaki prosedürü çağırır:

```
find_best(8)
```

$n = 8$ tane kutu vardır. Ödül türleri $[3, 2, 3, 1, 3, 3, 2, 3]$ olsun. `ask` prosedürüne olabilecek tüm çağrılar ve bunların dönme (return) değerleri aşağıda listelenmiştir:

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

Bu örnekte, elmas 3 kutusundadır. Yani, `find_best` prosedürü 3 dönmelidir.



Yukarıdaki şekil bu örneği açıklamaktadır. Üst kısım, her bir kutudaki ödüllerin türünü gösterir. Alt

kısıt, $ask(2)$ sorgusunu gösterir. İşaretli kutular 2 kutusundan daha pahalı ödülleri içerir.

Sınırlar

- $3 \leq n \leq 200\,000$.
- Her bir kutudaki ödül türü 1 ile v arasındadır (sınırlar dahildir).
- 1 türünde tam olarak bir tane ödül vardır.
- Tüm $2 \leq t \leq v$ için, $t - 1$ türünde k ödül varsa, t türünde k^2 'den daha fazla ödül vardır.

Altgörevler ve puanlama

Bazı test durumlarında, değerlendiricinin davranışı adaptifdir. Yani, bu test durumlarında değerlendirici sabit ödül sırasına sahip değildir. Onun yerine, değerlendiricinin cevapları çözümünüz tarafından sorulan sorulara bağlı olabilir. Değerlendiricinin şu şekilde cevap vereceği garantidir: Her bir cevap sonrasında, o ana kadar verilen tüm cevaplar ile tutarlı olan en az bir ödül sırası olacaktır.

1. (20 puan) Tam olarak 1 tane elmas ve $n - 1$ tane lolipop vardır (dolayısıyla, $v = 2$). ask prosedürünü en fazla 10 000 kez çağırabilirsiniz.
2. (80 puan) Ek kısıt yoktur.

Alt görev 2'de kısmi puan alabilirsiniz. Bu alt görevdeki tüm test durumları içinde, q ask prosedürüne olan maksimum çağrı sayısı olsun. Bu durumda, bu alt görev için puanınız aşağıdaki tabloya göre hesaplanır:

Sorular	Puan
$10\,000 < q$	0 (CMS 'de 'Wrong Answer' olarak verilir)
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

Örnek değerlendirici

Örnek değerlendirici adaptif değildir. Onun yerine, sadece okur ve ödül türlerini içeren sabit p dizisi kullanır. Tüm $0 \leq b \leq n - 1$ için, b kutusundaki ödül türü $p[b]$ olarak verilir. Örnek değerlendirici, girdiyi aşağıdaki formatta bekler:

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

Örnek değerlendirici, `find_best` 'in dönme (return) değerini ve ask prosedürüne olan çağrı sayısını içeren tek satır yazar.