



## Velika nagrada

"Velika nagrada" je poznati TV kviz. Ti si srećni takmičar koji je prošao u finalnu rundu. Pred tebe je postavljen niz od  $n$  kutija, označenih od 0 do  $n - 1$ , s lijeva na desno. Svaka kutija sadrži tačno jednu nagradu koja se ne može vidjeti dok se ta kutija ne otvori. Postoji  $v \geq 2$  tipova nagrada. Tipovi nagrada su numerisani od 1 do  $v$  po *opadajućem* redoslijedu vrijednosti.

Nagrada tipa 1 je najskuplja: dijamant. Postoji tačno jedan dijamant u kutijama. Nagrada tipa  $v$  je najjeftinija: lizalica. Kako bi igra bila uzbudljivija, broj jeftinijih nagrada je mnogo veći od broja skupljih nagrada. Preciznije, za svako  $t$  za koje  $2 \leq t \leq v$  znamo sljedeće: Ako ima  $k$  nagrada tipa  $t - 1$ , onda ima *strogo više* od  $k^2$  nagrada tipa  $t$ .

Tvoj cilj je da osvojiš dijamant. Na kraju igre otvaraš jednu kutiju i osvajaš nagradu koja se nalazi u njoj. Pre nego što odabereš koju ćeš kutiju otvoriti, imaš priliku da Rambodu, domaćinu kviza, postaviš neka pitanja. Za svako pitanje odabiraš neku kutiju  $i$ . Kao svoj odgovor, Rambod će ti dati niz  $a$  koji sadrži dva cijela broja. Njihovo značenje je sljedeće:

- Među svim kutijama lijevo od kutije  $i$  ima tačno  $a[0]$  kutija koje sadrže skuplju nagradu od nagrade u kutiji  $i$ .
- Među svim kutijama desno od kutije  $i$  ima tačno  $a[1]$  kutija koje sadrže skuplju nagradu od nagrade u kutiji  $i$ .

Na primjer, pretpostavimo da je  $n = 8$ . Za svoje pitanje odabrao si kutiju  $i = 2$ . Rambodov odgovor je  $a = [1, 2]$ . Značenje ovog odgovora je:

- Tačno jedna od kutija 0 i 1 sadrži nagradu koja je skuplja od nagrade u kutiji 2.
- Tačno dvije od kutija 3, 4, ..., 7 sadrže nagradu koja je skuplja od nagrade u kutiji 2.

Tvoj zadatak je da nađeš kutiju koja sadrži dijamant postavljajući mali broj pitanja.

## Detalji implementacije

Potrebno je da implementiraš sljedeću funkciju:

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

- $n$ : broj kutija.
- Ova funkcija treba da vrati oznaku kutije koja sadrži dijamant, tj. cijeli broj  $d$  ( $0 \leq d \leq n - 1$ ) takav da kutija  $d$  sadrži nagradu tipa 1.

Ova funkcija može da poziva sljedeću funkciju:

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

- $i$ : oznaka kutije za koju postavljaš pitanje. Vrijednost  $i$  mora biti između 0 i  $n - 1$ , uključno.
- Ova funkcija vraća niz  $a$  sa 2 elementa. Ovdje je  $a[0]$  broj skupljih nagrada u kutijama lijevo od kutije  $i$ , a  $a[1]$  je broj skupljih nagrada u kutijama desno od kutije  $i$ .

## Primjer

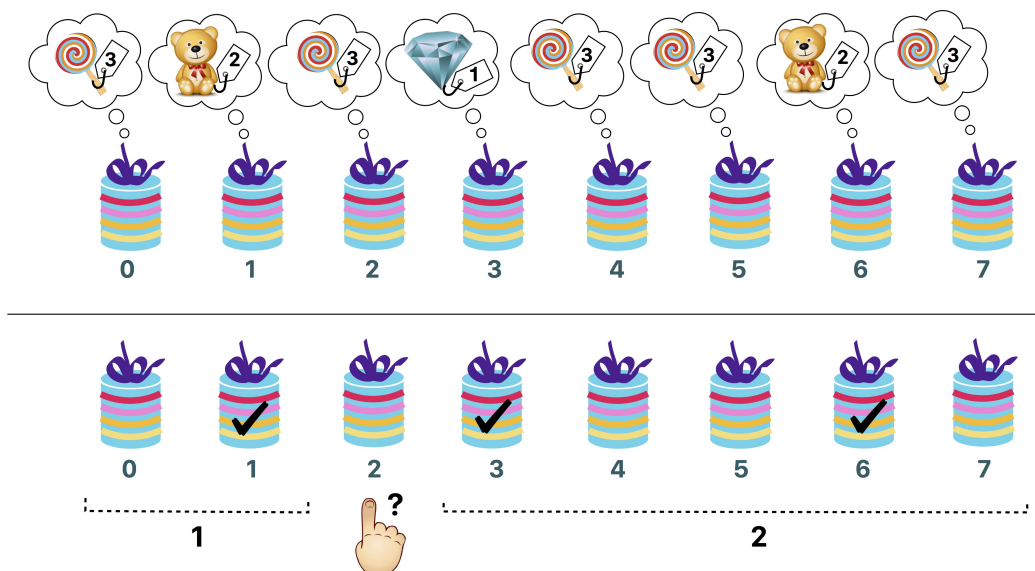
Grader poziva sljedeću proceduru:

```
find_best(8)
```

Postoji  $n = 8$  kutija. Pretpostavimo da su tipovi nagrada  $[3, 2, 3, 1, 3, 3, 2, 3]$ . Svi mogući pozivi procedure `ask` i odgovarajući odgovori su navedeni ispod.

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

U ovom primjeru, dijamant je u kutiji 3, te zbog toga procedura `find_best` treba da vrati 3.



Slika iznad prikazuje ovaj primjer. Gornji dio slike prikazuje tipove nagrada u svakoj kutiji. Donji dio slike prikazuje upit `ask(2)`. Čekirane kutije sadrže vrednije nagrade od nagrade koja se nalazi u kutiji 2.

## Ograničenja

- $3 \leq n \leq 200\,000$ .
- Tipovi nagrada u svakoj od kutija su između 1 i  $v$ , uključujući te brojeve.
- Postoji tačno jedna nagrada tipa 1.
- Za sve  $2 \leq t \leq v$ , ukoliko postoji  $k$  nagrada tipa  $t - 1$ , onda postoji *strogo* više od  $k^2$  nagrada tipa  $t$ .

## Podzadaci i bodovanje

U nekim test primjerima grader može da se ponaša adaptivno. To znači da u tim test primjerima grader nema fiksni niz nagrada, nego odgovor gradera može da zavisi od pitanja koje mu tvoj program postavi. Garantuje se da će grader odgovarati tako da nakon svakog odgovora postoji barem jedan niz nagrada konzistentan sa svim odgovorima datim do tada.

1. (20 points) Postoji tačno 1 dijamant i  $n - 1$  lizalica ( $v = 2$ ). Možeš pozvati proceduru `ask` najviše 10 000 puta.
2. (80 points) Nema dodatnih ograničenja.

U podzadatku 2 možeš da dobiješ parcijalne bodove. Neka je  $q$  najveći broj poziva procedure `ask` među svim test primjerima u ovom podzadatku. Onda će se tvoj broj bodova za ovaj podzadatak računati prema sljedećoj tabeli:

Broj pitanja	Bodovi
$10\,000 < q$	0 (prikazano u CMS-u kao 'Wrong Answer')
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

## Primjer grejdera

Primjer grejdera nije adaptivan. Umesto toga, primjer grejdera samo učitava i koristi fiksni niz tipova nagrada  $p$ . Za sve  $0 \leq b \leq n - 1$ , tip nagrade u kutiji  $b$  je dat sa  $p[b]$ . Primjer grejdera očekuje ulaz u sljedećem formatu:

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

Primjer grejdera ispisuje samo jednu liniju koja sadrži vraćenu vrijednost od funkcije `find_best` i broj poziva procedure `ask`.