



The Big Prize

The Big Prize is een bekend spelprogramma op TV. Jij bent de gelukkige deelnemer die is doorgedrongen tot de finale. Je staat voor een rij van n dozen, van links naar rechts gelabeld van 0 tot en met $n - 1$. Elke doos bevat een geschenk dat je pas kunt zien als je de doos opent. Er zijn $v \geq 2$ verschillende *soorten* prijzen. Deze soorten zijn genummerd van 1 tot en met v in *aflopende* volgorde van waarde.

Het geschenk van soort 1 is het duurst: een diamant. Precies één van alle dozen bevat een diamant. Het geschenk van soort v is het goedkoopst: een lolly. Om het spel wat leuker te maken is het aantal goedkope geschenken veel groter dan het aantal duurder geschenken. Specifieker: voor alle t met $2 \leq t \leq v$ weten we het volgende: als er k geschenken zijn van soort $t - 1$ dan zijn er *strikt meer* dan k^2 geschenken van soort t .

Jouw doel is om de diamant te winnen. Aan het einde van het spel moet je een doos openen en krijg je prijs die erin zit. Voordat je de doos kiest die je gaat openmaken kun je enkele vragen stellen aan Rambod, de gastheer van het spelprogramma. Voor elke vraag kun je een willekeurige doos i kiezen. Als antwoord geeft Rambod je een array a met twee integers. De betekenis ervan is als volgt:

- Van alle dozen links van doos i zijn er precies $a[0]$ dozen die een duurder geschenk bevatten dan datgene in doos i .
- Van alle dozen rechts van doos i zijn er precies $a[1]$ dozen die een duurder geschenk bevatten dan datgene in doos i .

Neem bijvoorbeeld aan dat $n = 8$. Voor jouw vraag kies je doos $i = 2$. Als antwoord vertelt Rambod je $a = [1, 2]$. De betekenis van dit antwoord is:

- In precies een van de twee dozen 0 en 1 zit een duurder geschenk dan in doos 2.
- In precies twee van de dozen 3, 4, \dots , 7 zit een duurder geschenk dan in doos 2.

Jouw taak is om de doos met de diamant te vinden, terwijl je zo min mogelijk vragen stelt.

Implementatiedetails

Implementeer de volgende functie:

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

- Deze functie wordt exact één keer aangeroepen door de grader.

- n : aantal dozen.
- Deze functie moet het label van de doos die de diamant bevat retourneren, oftewel het unieke getal d ($0 \leq d \leq n - 1$) zodanig doos d het geschenk bevat van soort 1.

De bovenstaande functie kan de volgende functie aanroepen:

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

- i : Het label van de doos waar je een vraag over stelt. De waarde van i moet zijn van 0 tot en met $n - 1$.
- De functie geeft een array a terug met 2 elementen. Het aantal duurdere geschenken aan de linkerkant van doos i wordt gegeven in $a[0]$ en het aantal duurdere geschenken aan de rechterkant van doos i in $a[1]$.

Voorbeeld

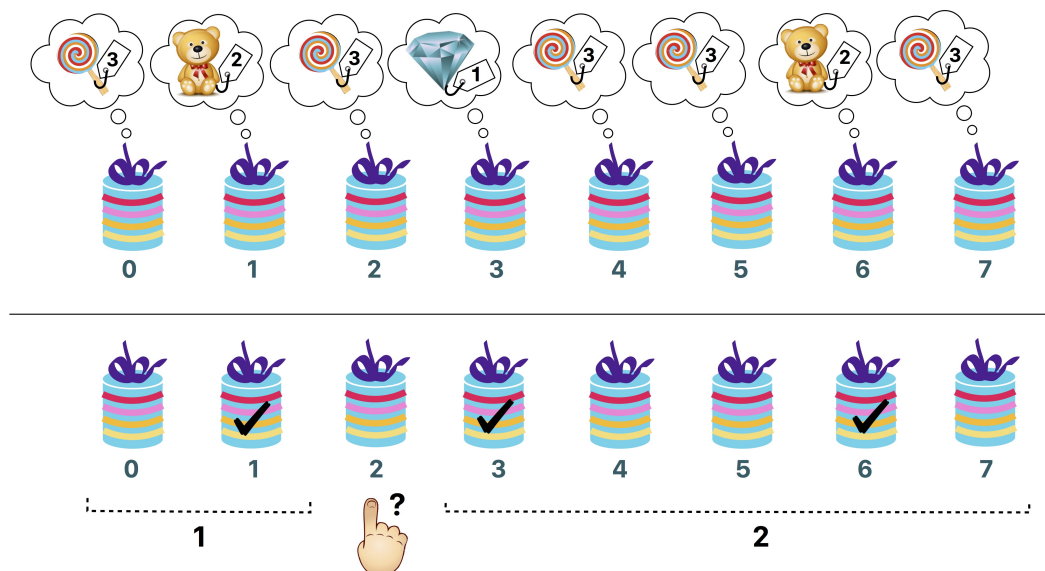
De grader roept de volgende functie aan:

```
find_best(8)
```

Er zijn dus $n = 8$ dozen. Neem aan dat de geschenken gegeven worden door $[3, 2, 3, 1, 3, 3, 2, 3]$. Alle mogelijke aanroepen aan de functie `ask` en de corresponderende waarde die wordt teruggegeven zijn hieronder weergegeven:

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

In dit voorbeeld zit de diamant in doos 3. De functie `find_best` moet dus 3 teruggeven.



Het plaatje hierboven illustreert dit voorbeeld. Het bovenste deel laat de waarden zien van de geschenken in elke doos. Het onderste deel illustreert de aanroep `ask(2)`. De gemarkeerde dozen bevatten duurdere geschenken dan die in doos 2.

Randvoorwaarden

- $3 \leq n \leq 200\,000$.
- De soort van elk geschenk ligt tussen 1 tot en met v .
- Er is precies een geschenk van type 1.
- Voor alle $2 \leq t \leq v$ geldt dat als er precies k geschenken van type $t - 1$ zijn, het aantal geschenken van soort t strict groter is dan k^2 .

Deeltaken en scoreverdeling

In sommige testgevallen is het gedrag van de grader afhankelijk van jouw programma. Dat betekent dat in deze testgevallen de grader geen vaste rij met geschenken heeft. De antwoorden van de grader zijn in plaats daarvan afhankelijk van jouw oplossing. Het staat vast dat de grader zodanig antwoordt dat na elk antwoord er altijd tenminste één rij van geschenken consistent is met de antwoorden die tot dusver zijn gegeven.

1. (20 punten) Er is precies 1 diamant en $n - 1$ lolly's (dus $v = 2$). Je mag de functie `ask` maximaal 10,000 keer aanroepen.
2. (80 punten) Geen extra voorwaarden.

In deeltaak 2 kan je een partiële score halen. Laat q het maximale aantal aanroepen zijn van de functie `ask` onder alle testgevallen in deze deeltaak. Dan wordt je score berekend volgens de volgende tabel:

Aanroepen	Score
$10\,000 < q$	0 (in CMS wordt 'Wrong Answer' teruggegeven)
$6000 < q \leq 10\,000$	70
$5000 < q \leq 6000$	$80 - (q - 5000)/100$
$q \leq 5000$	80

Voorbeeldgrader

Het gedrag van de voorbeeldgrader is niet afhankelijk van jouw programma. In plaats daarvan wordt een vast array p ingelezen met de verschillende soorten geschenken. Voor alle $0 \leq b \leq n - 1$ wordt de soort van het geschenk in doos b gegeven door $p[b]$.

De voorbeeldgrader verwacht invoer in het volgende formaat:

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

De voorbeeldgrader drukt een regel af met de geretourneerde waarde van `find_best` en het aantal aanroepen aan de functie `ask`.