დიდი პრიზი

"დიდი პრიზი" - არის ცნობილი სატელევიზიო შოუ. საბედნიეროდ, თქვენ მიაღწიეთ ამ შოუს ფინალს. თქვენს წინ განლაგებულია n რაოდენობის ყუთი, გადანომრილი 0 დან ($n-1$)-ის ჩათვლით.

ყოველ ყუთში განთავსებულია თითო პრიზი, რომელსაც ვერ ხედავთ, სანამ ყუთი არ გაიხსნება. პრიზების განსხვავებული სახეობების რაოდენობაა $v \geq 2$. სახეობები გადანომრილია 1-დან v-ს ჩათვლით პრიზების ღირებულების კლებადობის შესაბამისად.

1-ლი სახეობის პრიზი წარმოადგენს ბრილიანტს და ყველაზე ძვირია. ერთადერთი ბრილიანტი მხოლოდ ერთ ყუთშია განთავსებული. პრიზი სახეობით v, ლოლიპოპი, ყველაზე იაფია. თამაშის გართულების მიზნით უფრო იაფი პრიზების რაოდენობა ჭარბობს უფრო ძვირ პრიზებთან შედარებით. უფრო ზუსტად, ყველა t -თვის ცნობილია, რომ თუ $t-1$ სახეობის პრიზების რაოდენობაა k, მაშინ t სახეობის პრიზების რაოდენობა არის მკაცრად მეტი, ვიდრე k^2.

თქვენი მიზანია მოიგოთ ბრილიანტი. თამაშის ბოლოს თქვენ უნდა გახსნათ ყუთი და მიიღებთ, ანუ მოიგებთ ამ ყუთში განთავსებულ პრიზს. სანამ შეარჩევთ და გახსნით ყუთს, შეგიძლიათ დაუსვათ რამბოდს, შოუს წამყვანს, რამდენიმე შეკითხვა. ყოველ შეკითხვის დროს თქვენ ირჩევთ რომელიმე ყუთს ნომრით i. პასუხად რამბოდი გაწვდით a მასივის სახით ორ მთელ რიცხვს:

- i-ური ყუთის მარცხნივ მდებარე ყუთებს შორის არის ზუსტად $a[0]$ რაოდენობის ყუთი, რომელთაგან თითოეულში უფრო ძვირი პრიზია, ვიდრე i-ურ ყუთში მდებარე პრიზი;
- i-ური ყუთის მარჯვნივ მდებარე ყუთებს შორის არის ზუსტად $a[1]$ რაოდენობის ყუთი, რომელთაგან თითოეულში უფრო ძვირი პრიზია, ვიდრე i-ურ ყუთში მდებარე პრიზი.

მაგალითად, თუ $n=8$ და შეკითხვის დროს თქვენი არჩევანია $i=2$, ხოლო რამბოდის პასუხია $a=[1,2]$, ეს ნიშნავს:

- ზუსტად ერთ ყუთში ყუთებიდან 0 და 1 განთავსებულია უფრო ძვირი პრიზი, ვიდრე ყუთში 2;
- ზუსტად ორ ყუთში ყუთებიდან $3,4,\ldots,7$ განთავსებულია უფრო ძვირი პრიზები, ვიდრე ყუთში 2.

თქვენი ამოცანაა, რაც შეიძლება ნაკლები შეკითხვების დასმის შედეგად მოძებნოთ, რომელ ყუთშია განთავსებული ბრილიანტი.

იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ შემდეგი პროცედურის იმპლემენტაცია:

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

- n: ყუთების რაოდენობა.
- ამ პროცედურამ უნდა დააბრუნოს იმ ყუთის ნომერი, სადაც განთავსებულია ბრილიანტი, ანუ უნიკალური მთელი d ($0 \leq d \leq n-1$) ისე, რომ d ნომრის მქონე ყუთში განთავსებული იყოს 1-ლი სახეობის პრიზი.

ზემოთ აღწერილ პროცედურიდან შეგიძლიათ გამოიძახოთ შემდეგი პროცედურა:

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

- i: თქვენს მიერ შერჩეული ყუთის ნომერი. i სიდიდით უნდა იყოს 0-დან ($n-1$)-ის ჩათვლით.
- ეს პროცედურა დააბრუნებს ორელემენტიან მასივს, სადაც $a[0]$ არის i-ური ყუთის მარცხნივ უფრო ძვირი პრიზების რაოდენობა, ვიდრე i-ურ ყუთში მდებარე პრიზია. ხოლო $a[1]$ არის i ყუთის მარჯვნივ უფრო ძვირი პრიზების რაოდენობა, ვიდრე i-ურ ყუთში მდებარე პრიზია.

მაგალითი

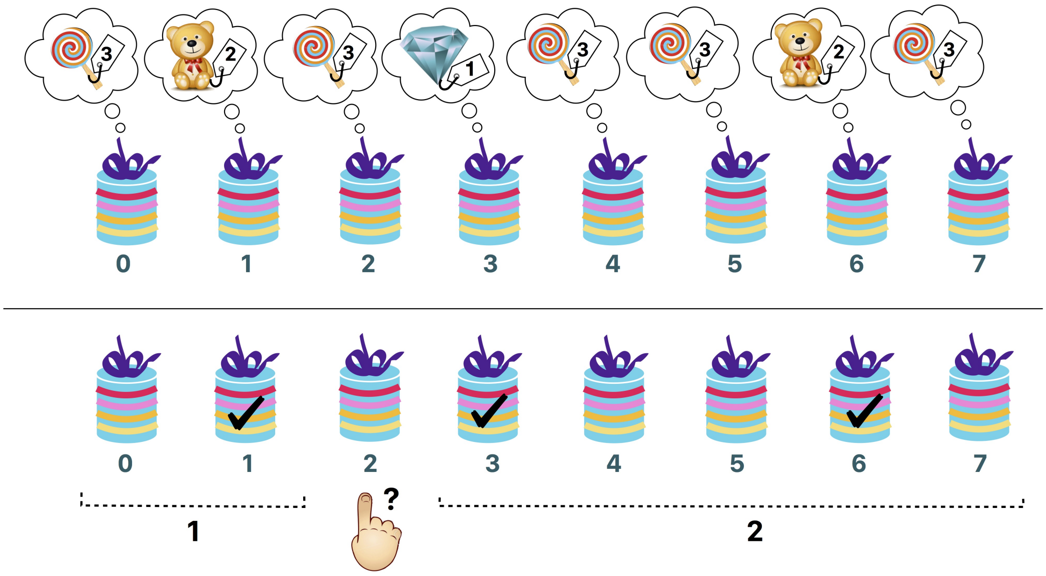
გრადერი მოახდენს შემდეგი პროცედურის გამოძახებას:

```
find_best(8)
```

მოცემულია $n=8$ ყუთი. დავუშვათ, პრიზების სახეობები შემდეგია: $[3,2,3,1,3,3,2,3]$. ask პროცედურის ყველა შესაძლო გამოძახებები და შესაბამისი პასუხები მოყვანილია შემდეგი სიის სახით:

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

ამ მაგალითში ბრილიანტი განთავსებულია ყუთში 3. ამიტომ პროცედურამ უნდა დააბრუნოს 3.



ზემოთ მოყვანილ ნახატზე მოცემულია მაგალითის ასახვა. ზედა ნაწილში ნაჩვენებია ყუთებში განთავსებული პრიზები. ქვედა ნაწილში ასახულია ინტერაქტივი `ask(2)`. მონიშნულ ყუთებში უფრო ძვირი პრიზებია, ვიდრე ყუთში 2.

შეზღუდვები

- $3 \leq n \leq 200\,000$;
- პრიზების სახეობები ეკუთვნის დიაპაზონს : $1..v$;
- 1-ლი სახეობის პრიზი ზუსტად ერთია;
- ყველა t -სათვის, სადაც $2 \leq t \leq v$, თუ $t-1$ სახეობის პრიზების რაოდენობაა k, მაშინ t სახეობის პრიზების რაოდენობა იქნება მკაცრად მეტი, ვიდრე k^2.

ქვეამოცანები და შეფასება

ზოგიერთ ტესტში გრადერის ქცევა ადაპტიურია. ეს ნიშნავს, რომ ამ ტესტებში გრადერს წინასწარ არ აქვს დაფიქსირებული პრიზების მიმდევრობა. ამის მაგივრად გრადერის მიერ გაცემული პასუხები დამოკიდებულია თქვენს მიერ დასმულ შეკითხვებზე. გარანტირებულია, რომ ყოველი შეკითხვის შემდეგ გრადერი პასუხობს ისე, რომ არსებობდეს პრიზების მინიმუმ ერთი კორექტული მიმდევრობა, რომელიც შეესაბამება ინტერაქტივის ოქმს.

1. (20 ქულა) მოცემულია ზუსტად 1 ბრილიანტი და $(n-1)$ რაოდენობის ლოლიპოპი (ანუ $v=2$). თქვენ შეგიძლიათ გამოიძახოთ პროცედურა `ask` მაქსიმუმ $10\,000$ -ჯერ.
2. (80 ქულა) დამატებითი შეზღუდვების გარეშე.

მე-2 ქვეამოცანაში შესაძლოა მიიღოთ ნაწილობრივი ქულა. ვთქვათ, q არის ამ ქვეამოცანაში ყველა ტესტში თქვენს მიერ დასმული `ask` შეკითხვების მაქსიმალური რაოდენობა. მაშინ, ამ ქვეამოცანაში თქვენს მიერ მიღებული ქულა დაითვლება შემდეგი ცხრილის მიხედვით:

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

სანიმუშო გრადერი

სანიმუშო გრადერი არ არის ადაპტიური. ის მხოლოდ კითხულობს და იყენებს პრიზების სახეობების დაფიქსირებულ p მასივს. b ყუთში, სადაც $0 \leq b \leq n-1$, განთავსებული პრიზის სახეობა მოიცემა როგორც $p[b]$. სანიმუშო გრადერი კითხულობს მონაცემებს შემდეგი ფორმატით:

- სტრიქონი 1: n
- სტრიქონი 2: $p[0]\ p[1]\ \ldots\ p[n-1]$

სანიმუშო გრადერი ბეჭდავს ერთადერთ სტრიქონს, რომელიც მოიცავს `find_best` პროცედურის მიერ დაბრუნებულ პასუხს და `ask` პროცედურის გამოძახებების რაოდენობას.