



Böyük Hədiyyə

Böyük Hədiyyə məşhur TV oyun-şousudur. Bəxtiniz gətirib və oyunun final mərhələsinə irəliləmişiniz. Soldan sağa 0-dan $n - 1$ -ə qədər nömrələnmiş n sayda qutudan ibarət sıranın qarşısındasınız. Hər qutunun içində qutu açılmayınca nə olduğu bilinməyən hədiyyə var. $v \geq 2$ sayda hədiyyə növü mövcuddur. Növlər qiymətin *azalma* ardıcılığı ilə 1-dən v -yə qədər nömrələnmişdir.

Ən bahalı olan növ 1 brilyantdır. Qutularda cəmi bir dənə brilyant var. Qiyməti ən ucuz olan v növlü hədiyyə isə konfetdir. Oyunu daha da maraqlı etmək üçün, ucuz hədiyyələrin sayı bahalı hədiyyələrin sayından xeyli sayda çoxdur. Daha dəqiq desək, hər bir t üçün, belə ki, $2 \leq t \leq v$, aşağıdakı doğrudur: $t - 1$ növlü hədiyyələrin sayı k olarsa, t növlü hədiyyələrin sayı k^2 -dan *ciddi böyükdür*.

Məqsədiniz brilyantı qazanmaqdır. Oyunun sonunda hər hansı qutunu açmalısınız, və onun içindəki hədiyyəni qazanacaqsınız. Qutunu seçməkdən əvvəl, oyun şousunun aparıcısı Ramboda bəzi suallar verə bilərsiniz. Hər dəfə sual olaraq hər hansı i qutusunu seçirsiniz. Cavab olaraq Rambod sizə iki tam ədəddən ibarət massiv verəcək. Bu ədədlərin mənası aşağıdakı kimidir:

- i qutusunun solunda tam olaraq $a[0]$ sayda qutu var ki, içlərindəki hədiyyə i qutusunda olan hədiyyədən bahadır;
- i qutusunun sağında tam olaraq $a[1]$ sayda qutu var ki, içlərindəki hədiyyə i qutusunda olan hədiyyədən bahadır.

Məsələn, tutaq ki, $n = 8$. Sual olaraq $i = 2$ qutusunu seçirsiniz. Cavab olaraq Rambod sizə $a = [1, 2]$ deyir. Bu cavabın mənası aşağıdakı kimidir:

- 0 və 1 qutularından yalnız birində 2 qutusunda olan hədiyyədən daha bahalı hədiyyə var;
- 3, 4, ..., 7 qutularından yalnız ikisində 2 qutusunda olan hədiyyədən daha bahalı hədiyyələr var.

Az sayda sual verməklə brilyantın gizlədildiyi qutunu tapın.

Gerçəkləşdirmə detalları

Aşağıdakı proseduru gerçəkləşdirməlisiniz:

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

- Bu prosedur yoxlayıcı sistem tərəfindən yalnız bir dəfə çağrılır.
- n : qutuların sayıdır.
- Bu prosedur brilyant olan qutunun nömrəsini qaytarmalıdır, yəni, elə bir yeganə b (

$0 \leq b \leq n - 1$) tam ədədini ki, b qutusunun içindəki hədiyyənin növü 1 olsun.

Yuxarıdakı prosedur aşağıdakı proseduru çağırır bilər:

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

- i : barəsində sual verməyi seçdiyiniz qutunun nömrəsidir. i -nin qiyməti 0-dan $n - 1$ -ə qədər (daxil olmaqla) dəyişir.
- Bu prosedur 2 elementdən ibarət a massivini qaytarır. Burada $a[0]$, i qutusunda solda yerləşən daha bahalı hədiyyələrin sayını, $a[1]$ isə i qutusunda sağda yerləşən daha bahalı hədiyyələrin sayını göstərir.

Nümunə

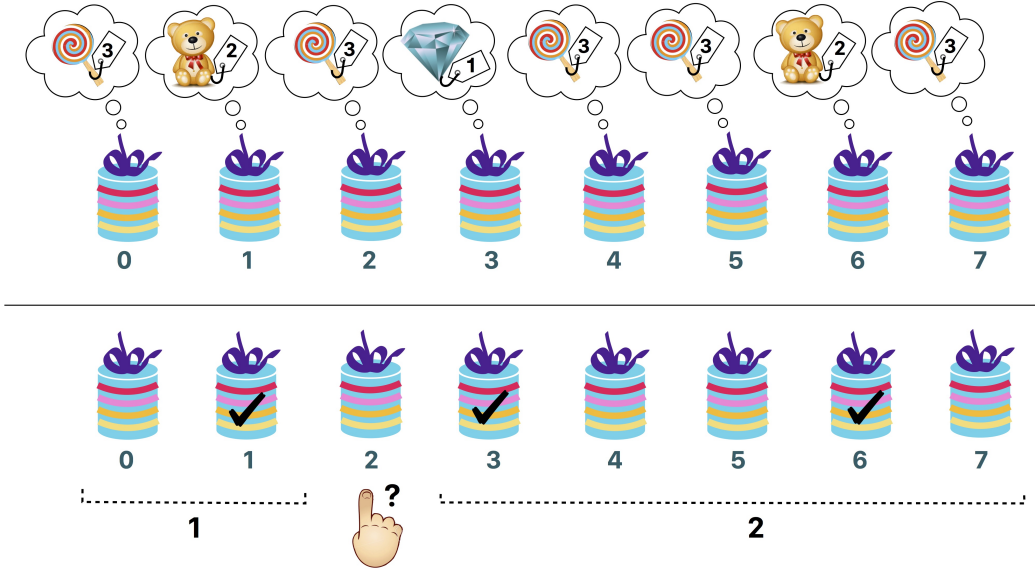
Yoxlayıcı sistem aşağıdakı proseduru çağırır:

```
find_best(8)
```

$n = 8$ dənə qutu var. Tutaq ki, hədiyyə növləri yuxarıdakı şəkildə göstərildiyi kimidir: $[3, 2, 3, 1, 3, 3, 2, 3]$. `ask` prosedurunun bütün mümkün çağırışları və uyğun cavabları aşağıdakı siyahıda verilmişdir:

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

Bu nümunədə, axtarılan hədiyyə 3 qutusunun içindədir. Yəni, `find_best` prosedurunun nəticəsi 3 olmalıdır.



Yuxarıdakı şəkil bu nümunəni təsvir edir. Yuxarı hissə hər qutuda olan hədiyyələrin növünü göstərir. Aşağı hissə $ask(2)$ sorğusunu təsvir edir. Nişanlanmış qutularda olan hədiyyələr 2 qutusunda olan hədiyyədən daha bahalıdır.

Məhdudiyyətlər

- $3 \leq n \leq 200\,000$.
- Hədiyyələrin qiyməti 1-dən v -yə qədər dəyişir.
- 1 növlü hədiyyədən bir dənədir.
- Hər bir $2 \leq t \leq v$ üçün, $t - 1$ növlü k sayda hədiyyə varsa, t növlü hədiyyələrin sayı k^2 -dan ciddi böyükdür.

Altməsələlər və xalların hesablanması

Bəzi testlər üçün yoxlayıcı sistem uyğunlaşandır (adaptivdir). Bu o deməkdir ki, bu testlər üçün yoxlayıcı sistem əvvəlcədən hədiyyələrin sıralamasını təyin etmir. Bunun yerinə, yoxlayıcının verdiyi cavablar sizin həllinizin sorğularından asılı ola bilər. Yoxlayıcının verdiyi bütün cavablara uyğun ən azı bir hədiyyələr sırasının mövcud olmasına təminat verilir.

1. (20 xal) Yalnız 1 brilyant və $n - 1$ konfet var. ($v = 2$.) ask prosedurunun ən çoxu 10 000 dəfə çağırmaq olar.
2. (80 xal) Əlavə məhdudiyyət yoxdur.

İkinci altməsələdə qismən nəticə əldə etmək olar. Deyək ki, q bu alt-tapşırığın bütün testləri arasında ask prosedurunun çağırıldığı maksimum sayı göstərir. O zaman, bu alt-tapşırıq üçün sizin xallarınız növbəti cədvəle əsasən hesablanır:

Suallar	Nəticə
$10\,000 < q$	0 (CMS-də 'Wrong Answer' kimi göstərilir)
$6000 < q \leq 10\,000$	68
$4800 < q \leq 6000$	$80 - (q - 4800)/100$
$q \leq 4800$	80

Nümunə yoxlayıcı

Nümunə yoxlayıcı uyğunlaşan (adaptiv) deyil. Bunun əvəzinə, əvvəlcədən təyin edilmiş hədiyyə növlərinin p massivi istifadə olunur: b qutusunda hədiyyənin qiyməti $p[b]$ -dir. Nümunə yoxlayıcı girişdən aşağıdakı formatda oxuyur:

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

Nümunə yoxlayıcı `find_best` prosedurunun nəticəsini və `ask` proseduruna olan sorğuların sayını yeganə sətirdə çap edir.