



Rarest Insects

Có N con côn trùng, được đánh số từ 0 đến $N - 1$, chạy quanh nhà của Pak Blangkon. Mỗi con côn trùng thuộc một **loài**, là một số nguyên từ 0 đến 10^9 bao gồm cả hai đầu mút. Có thể có nhiều con côn trùng cùng một loài.

Giả thiết các con côn trùng được nhóm theo loài. Ta định nghĩa lực lượng của loài côn trùng **thường gặp nhất** là số lượng con côn trùng trong nhóm có số lượng con côn trùng nhiều nhất. Tương tự, lực lượng của loài côn trùng **hiếm nhất** là số lượng con côn trùng trong nhóm có số lượng con côn trùng ít nhất.

Ví dụ, có 11 con côn trùng, thuộc các loài tương ứng là $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. Trong trường hợp này, lực lượng của loài côn trùng **thường gặp nhất** là 3. Các loài có số lượng con côn trùng nhiều nhất là loài 9 và loài 11, mỗi loài gồm có 3 con côn trùng. Lực lượng của loài côn trùng **hiếm nhất** là 1. Các loài có ít con côn trùng nhất là loài 7, loài 0 và loài 100, mỗi loài gồm 1 con côn trùng.

Pak Blangkon không biết loài côn trùng nào cả. Anh ta có một chiếc máy chỉ với một nút bấm có thể cung cấp một số thông tin về các loài côn trùng. Ban đầu, máy trống rỗng. Để sử dụng máy, có thể thực hiện ba loại thao tác:

1. Di chuyển một con côn trùng vào trong máy.
2. Di chuyển một con côn trùng ra ngoài máy.
3. Nhấn nút trên máy.

Mỗi loại thao tác có thể được thực hiện nhiều nhất là 40 000 lần.

Bất cứ khi nào nút được nhấn, máy sẽ báo lực lượng loài côn trùng **thường gặp nhất**, chỉ xét các con côn trùng bên trong máy.

Nhiệm vụ của bạn là xác định lực lượng của loài côn trùng **hiếm nhất** trong số tất cả N con côn trùng trong nhà của Pak Blangkon bằng cách sử dụng máy. Ngoài ra, trong một số subtask, điểm của bạn phụ thuộc vào số lượng tối đa các thao tác của một loại đã sử dụng (xem phần Subtask để biết thêm chi tiết).

Chi tiết cài đặt

Bạn cần cài đặt hàm sau:

```
int min_cardinality(int N)
```

- N : số lượng con côn trùng.
- Hàm này trả về lực lượng của loài côn trùng **hiếm nhất** trong số tất cả N con côn trùng trong nhà của Pak Blangkon.
- Hàm này được gọi đúng một lần.

Hàm trên có thể thực hiện các lời gọi đến các hàm sau:

```
void move_inside(int i)
```

- i : chỉ số của con côn trùng được di chuyển vào trong máy. Giá trị của i phải nằm trong khoảng từ 0 đến $N - 1$ bao gồm cả hai đầu mút.
- Nếu con côn trùng này đã ở trong máy, thì lời gọi này không ảnh hưởng đến tập các con côn trùng trong máy. Tuy nhiên, nó vẫn được tính là một lời gọi riêng biệt.
- Hàm này có thể được gọi nhiều nhất là 40 000 lần.

```
void move_outside(int i)
```

- i : chỉ số của con côn trùng được di chuyển ra ngoài máy. Giá trị của i phải nằm trong khoảng từ 0 đến $N - 1$ bao gồm cả hai đầu mút.
- Nếu con côn trùng này đã ở bên ngoài máy, lời gọi này sẽ không ảnh hưởng đến tập các con côn trùng trong máy. Tuy nhiên, nó vẫn được tính là một lời gọi riêng biệt.
- Hàm này có thể được gọi nhiều nhất là 40 000 lần.

```
int press_button()
```

- Hàm này trả về lực lượng của loài côn trùng **thường gặp nhất**, chỉ xét các con côn trùng bên trong máy.
- Hàm này có thể được gọi nhiều nhất là 40 000 lần.
- Trình chấm là **không thích ứng**. Nghĩa là, các loài của N con côn trùng được cố định trước khi gọi `min_cardinality`.

Ví dụ

Xét một kịch bản trong đó có 6 con côn trùng thuộc các loài tương ứng [5, 8, 9, 5, 9, 9]. Hàm `min_cardinality` được gọi như sau:

```
min_cardinality(6)
```

Hàm có thể gọi `move_inside`, `move_outside`, và `press_button` như sau.

Lời gọi	Giá trị trả về	Các con côn trùng trong máy	Loài của các con côn trùng trong máy
		{}	[]
move_inside(0)		{0}	[5]
press_button()	1	{0}	[5]
move_inside(1)		{0,1}	[5,8]
press_button()	1	{0,1}	[5,8]
move_inside(3)		{0,1,3}	[5,8,5]
press_button()	2	{0,1,3}	[5,8,5]
move_inside(2)		{0,1,2,3}	[5,8,9,5]
move_inside(4)		{0,1,2,3,4}	[5,8,9,5,9]
move_inside(5)		{0,1,2,3,4,5}	[5,8,9,5,9,9]
press_button()	3	{0,1,2,3,4,5}	[5,8,9,5,9,9]
move_inside(5)		{0,1,2,3,4,5}	[5,8,9,5,9,9]
press_button()	3	{0,1,2,3,4,5}	[5,8,9,5,9,9]
move_outside(5)		{0,1,2,3,4}	[5,8,9,5,9]
press_button()	2	{0,1,2,3,4}	[5,8,9,5,9]

Tại thời điểm này, có đủ thông tin để kết luận rằng lực lượng của loài côn trùng hiếm nhất là 1. Do đó, hàm `min_cardinality` cần trả về 1.

Trong ví dụ này, `move_inside` được gọi 7 lần, `move_outside` được gọi 1 lần và `press_button` được gọi 6 lần.

Ràng buộc

- $2 \leq N \leq 2000$

Subtask

1. (10 điểm) $N \leq 200$
2. (15 điểm) $N \leq 1000$
3. (75 điểm) Không có ràng buộc gì thêm.

Nếu trong bất kì trường hợp thử nghiệm nào, các lời gọi đến các hàm `move_inside`, `move_outside` hoặc `press_button` không tuân theo các ràng buộc được mô tả trong phần Chi

tiết cài đặt hoặc giá trị trả về của `min_cardinality` không đúng, điểm của bạn cho subtask đó sẽ là 0.

Gọi q là giá trị **lớn nhất** trong ba giá trị sau: số lần gọi hàm `move_inside`, số lời gọi hàm `move_outside` và số lần gọi hàm `press_button`.

Trong subtask 3, bạn có thể nhận được một phần điểm. Gọi m là giá trị lớn nhất của $\frac{q}{N}$ trong tất cả các trường hợp thử nghiệm của subtask này. Điểm của bạn cho subtask này được tính theo bảng sau:

Điều kiện	Điểm
$20 < m$	0 (thông báo "output isn't correct" trong CMS)
$6 < m \leq 20$	$\frac{225}{m-2}$
$3 < m \leq 6$	$81 - \frac{2}{3}m^2$
$m \leq 3$	75

Trình chấm mẫu

Gọi T là một mảng gồm N số nguyên trong đó $T[i]$ là loài của con côn trùng i .

Trình chấm mẫu đọc dữ liệu vào theo định dạng sau:

- dòng 1: N
- dòng 2: $T[0] T[1] \dots T[N-1]$

Nếu trình chấm mẫu phát hiện được sự vi phạm giao thức, trình chấm mẫu sẽ xuất ra Protocol Violation: `<MSG>`, trong đó `<MSG>` thuộc một trong các loại sau:

- `invalid parameter`: trong lời gọi hàm `move_inside` hoặc `move_outside`, giá trị của i không thuộc đoạn 0 và $N-1$ bao gồm cả hai đầu mút.
- `too many calls`: số lượng lời gọi tới **một trong** `move_inside`, `move_outside`, hoặc `press_button` vượt quá 40 000.

Ngược lại, trình chấm sẽ xuất ra theo khuôn dạng sau:

- dòng 1: giá trị trả về của `min_cardinality`
- dòng 2: q