



Insectos Raros

Hay N insectos, numerados de 0 a $N - 1$, merodeando alrededor de la casa de Pak Blangkon. Cada insecto tiene un **tipo**, que es un entero entre 0 y 10^9 inclusive. Múltiples insectos pueden tener el mismo tipo.

Supongamos que los insectos están agrupados por tipo. Definimos la cardinalidad del tipo de insecto **más frecuente** como el número de insectos en un grupo con el mayor número de insectos. De forma similar, la cardinalidad del tipo de insecto **más raro** es el número de insectos en un grupo con el menor número de insectos.

Por ejemplo, supongamos que hay 11 insectos, cuyos tipos son $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. En este caso, la cardinalidad del tipo de insecto **más frecuente** es 3. Los grupos con el mayor número de insectos son los de tipo 9 y tipo 11, cada uno conformado por 3 insectos. La cardinalidad del tipo de insecto **más raro** es 1. Los grupos con el menor número de insectos son los de tipo 7, tipo 0, y tipo 100, cada uno conformado por 1 insecto.

Pak Blangkon no conoce el tipo de ningún insecto. Él tiene una máquina con un solo botón que puede proporcionar información acerca del tipo de los insectos. Inicialmente, la máquina está vacía. Para usar la máquina, tres tipos de operaciones pueden ser ejecutadas:

1. Mover un insecto dentro de la máquina.
2. Mover un insecto fuera de la máquina.
3. Presionar el botón de la máquina.

Cada tipo de operación puede ser ejecutada como máximo 40 000 veces.

Cuando el botón es presionado, la máquina reporta la cardinalidad del tipo de insecto **más frecuente**, considerando solo los insectos dentro de la máquina.

Tu tarea es determinar la cardinalidad del tipo de insecto **más raro** de entre todos los N insectos en la casa de Pak Blangkon, usando la máquina. Adicionalmente, en algunas subtareas, tu puntaje dependerá del máximo número de operaciones de cierto tipo que son ejecutadas (ver la sección de Subtareas para más detalles)

Detalles de Implementación

Debes implementar la siguiente función:

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

- N : el número de insectos.
- Esta función debe retornar la cardinalidad del tipo de insecto **más raro** de entre todos los N insectos en la casa de Pak Blangkon.
- La función es llamada exactamente una vez.

La función antes descrita puede hacer llamadas a las siguientes funciones:

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

- i : el índice del insecto que será movido dentro de la máquina. El valor de i debe estar entre 0 y $N - 1$ inclusive.
- Si este insecto ya está dentro de la máquina, la llamada no tiene efecto en el conjunto de insectos en la máquina. Sin embargo, esta es contada como una llamada.
- Esta función puede ser llamada como máximo 40 000 veces.

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

- i : el índice del insecto que será movido fuera de la máquina. El valor de i debe estar entre 0 y $N - 1$ inclusive.
- Si este insecto ya está fuera de la máquina, la llamada no tiene efecto en el conjunto de insectos en la máquina. Sin embargo, esta es contada como una llamada.
- Esta función puede ser llamada como máximo 40 000 veces.

```
int press_button()
```

- Esa función retorna la cardinalidad del tipo de insecto **más frecuente**, considerando solamente los insectos dentro de la máquina.
- Esta función puede ser llamada como máximo 40 000 veces.
- El grader **no es adaptativo**. Es decir, los tipos de los N insectos son establecidos antes de la llamada a `min_cardinality`.

Ejemplo

Considera el siguiente escenario en el cual hay 6 insectos de tipos $[5, 8, 9, 5, 9, 9]$ respectivamente. La función `min_cardinality` es llamada de la siguiente forma:

```
min_cardinality(6)
```

La función puede llamar `move_inside`, `move_outside`, y `press_button` como sigue.

| Llamada | Valor de Retorno | Insectos en la máquina | Tipos de los insectos en la máquina |
|-----------------|------------------|------------------------|-------------------------------------|
| | | {} | [] |
| 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] |

En este punto, hay suficiente información para concluir que la cardinalidad del tipo de insecto más raro es 1. Entonces, la función `min_cardinality` debe retornar 1.

En este ejemplo, `move_inside` es llamado 7 veces, `move_outside` es llamado 1 vez, y `press_button` es llamado 6 veces.

Restricciones

- $2 \leq N \leq 2000$

Subtareas

1. (10 puntos) $N \leq 200$
2. (15 puntos) $N \leq 1000$
3. (75 puntos) Sin restricciones adicionales.

Si en alguno de los casos de prueba, las llamadas a las funciones `move_inside`, `move_outside`, o `press_button` no están conforme a las restricciones descritas en la sección de Detalles de

Implementación, o el valor de retorno de `min_cardinality` es incorrecto, el puntaje de tu solución para esa subtarea será 0.

Sea q el número **máximo** de los siguientes tres valores: el número de llamadas a `move_inside`, el número de llamadas a `move_outside`, y el número de llamadas a `press_button`.

En la subtarea 3, tu puedes obtener un puntaje parcial. Sea m el valor máximo de $\frac{q}{N}$ de entre todos los casos de prueba de esta subtarea. Tu puntaje para esta subtarea es calculada acorde a la siguiente tabla:

| Condición | Puntos |
|-----------------|---|
| $20 < m$ | 0 (reportado como "Output isn't correct" en el CMS) |
| $6 < m \leq 20$ | $\frac{225}{m-2}$ |
| $3 < m \leq 6$ | $81 - \frac{2}{3}m^2$ |
| $m \leq 3$ | 75 |

Grader de ejemplo

Sea T un vector de N enteros donde $T[i]$ es el tipo del insecto i .

El grader de ejemplo lee la entrada en el siguiente formato:

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

Si el grader de ejemplo detecta una violación de protocolo, la salida del grader de ejemplo es `Protocol Violation: <MSG>`, donde `<MSG>` puede ser uno de los siguientes mensajes:

- `invalid parameter`: en una llamada a `move_inside` o `move_outside`, el valor de i no está entre 0 y $N - 1$ inclusive.
- `too many calls`: el número de llamadas a **cualquiera** de las funciones `move_inside`, `move_outside`, o `press_button` excede 40 000.

De otro modo, la salida del grader de ejemplo sigue el siguiente formato:

- línea 1: el valor de retorno de `min_cardinality`
- línea 2: q