



## Rarest Insects

Возле дома Пака Блангкона живут  $N$  насекомых, пронумерованных от 0 до  $N - 1$ . Каждое из насекомых имеет определенный **тип**, который является целым числом от 0 до  $10^9$ , включительно. Несколько насекомых могут иметь один и тот же тип.

Сгруппируем насекомых по типу. Назовем **максимальной частотой** количество насекомых в группе, которая имеет наибольший размер. Аналогичным образом **минимальной частотой** назовем количество насекомых в группе, которая имеет наименьший размер.

К примеру, пусть всего есть 11 насекомых с типами  $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$ . Тогда **максимальная частота** равна 3. Группы с наибольшим количеством насекомых соответствуют типам 9 и 11, каждая из них состоит из 3 насекомых. **Минимальная частота** равна 1. Группы с наименьшим количеством насекомых соответствуют типам 7, 0 и 100, каждая из них состоит из 1 насекомого.

Пак Блангкон не знает типы насекомых. У него есть специальный аппарат с единственной кнопкой, сообщающий какую-то информацию о типах насекомых. Изначально аппарат пуст. Для использования аппарата можно выполнить три типа действий:

1. Поместить насекомое в аппарат.
2. Достать насекомое из аппарата.
3. Нажать кнопку на аппарате.

Каждое из действий может быть выполнено не более 40 000 раз.

При нажатии кнопки аппарат сообщает **максимальную частоту** среди тех насекомых, которые находятся внутри аппарата.

Ваша задача состоит в том, чтобы определить **минимальную частоту** среди всех  $N$  насекомых Пака Блангкона с помощью предоставленного аппарата. В некоторых подзадачах ваш балл дополнительно зависит от максимального числа действий определенного вида, которые были выполнены с аппаратом (смотрите секцию Subtasks).

## Implementation Details

Вы должны реализовать функцию:

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

- $N$ : количество насекомых.
- Эта функция должна возвращать **минимальную частоту** среди всех  $N$  насекомых в доме Пака Блангкона.
- Эта функция будет вызвана ровно один раз.

Вышеописанная функция может вызывать функцию

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

- $i$ : номер насекомого, которого требуется поместить в аппарат. Значение  $i$  должно находиться в диапазоне от 0 до  $N - 1$ , включительно.
- Если насекомое уже находится в аппарате, то этот вызов не влияет на множество насекомых внутри аппарата. Однако, он все еще считается отдельным вызовом функции.
- Эта функция может быть вызвана не более 40 000 раз.

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

- $i$ : номер насекомого, которого требуется достать из аппарата. Значение  $i$  должно находиться между 0 и  $N - 1$ , включительно.
- Если насекомое уже не находится в аппарате, то этот вызов не влияет на множество насекомых внутри аппарата. Однако, он все еще считается отдельным вызовом функции.
- Эта функция может быть вызвана не более 40 000 раз.

```
int press_button()
```

- Эта функция возвращает **максимальную частоту** среди насекомых внутри аппарата.
- Эта функция может быть вызвана не более 40 000 раз.
- Грейдер в этой задаче **не адаптивный**. Это означает, что типы всех  $N$  насекомых зафиксированы перед первым вызовом функции `min_cardinality`.

## Example

Рассмотрим пример, в котором есть 6 насекомых с типами  $[5, 8, 9, 5, 9, 9]$ , соответственно. Функция `min_cardinality` будет вызвана следующим образом:

```
min_cardinality(6)
```

Эта функция может вызывать функции `move_inside`, `move_outside` и `press_button` как показано ниже:

Вызов	Возвращаемое значение	Насекомые внутри аппарата	Типы насекомых внутри аппарата
		{}	[]
<code>move_inside(0)</code>		{0}	[5]
<code>press_button()</code>	1	{0}	[5]
<code>move_inside(1)</code>		{0, 1}	[5, 8]
<code>press_button()</code>	1	{0, 1}	[5, 8]
<code>move_inside(3)</code>		{0, 1, 3}	[5, 8, 5]
<code>press_button()</code>	2	{0, 1, 3}	[5, 8, 5]
<code>move_inside(2)</code>		{0, 1, 2, 3}	[5, 8, 9, 5]
<code>move_inside(4)</code>		{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]
<code>move_inside(5)</code>		{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
<code>press_button()</code>	3	{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
<code>move_inside(5)</code>		{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
<code>press_button()</code>	3	{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
<code>move_outside(5)</code>		{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]
<code>press_button()</code>	2	{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]

После всех вышеописанных вызовов мы обладаем информацией для того, чтобы заключить, что наименьшей частотой среди всех насекомых является число 1. Поэтому функция `min_cardinality` должна вернуть 1.

В данном примере функция `move_inside` была вызвана 7 раз, `move_outside` была вызвана 1 раз, а функция `press_button` была вызвана 6 раз.

## Constraints

- $2 \leq N \leq 2000$

## Subtasks

1. (10 баллов)  $N \leq 200$
2. (15 баллов)  $N \leq 1000$

3. (75 баллов) Без дополнительных ограничений.

Если в каком-либо из тестов подзадачи аргументы вызовов функций `move_inside`, `move_outside` или `press_button` не удовлетворяют ограничениям, описанным в разделе `Implementation Details`, или же возвращаемое значение функции `min_cardinality` не является корректным, вы получите 0 баллов за эту подзадачу.

Пусть  $q$  обозначает **максимальное** из трех значений: числа вызовов функции `move_inside`, числа вызовов функции `move_outside` и числа вызовов функции `press_button`.

В подзадаче 3 вы можете также получить частичный балл. Пусть  $m$  обозначает максимальное значение  $\frac{q}{N}$  среди всех тестов данной подзадачи. Ваш балл за эту подзадачу вычисляется согласно приведенной ниже таблице:

Условие	Баллы
$20 < m$	0 (отображается как "Output isn't correct" в CMS)
$6 < m \leq 20$	$\frac{225}{m-2}$
$3 < m \leq 6$	$81 - \frac{2}{3}m^2$
$m \leq 3$	75

## Sample Grader

Пусть  $T$  — массив из  $N$  целых чисел, где  $T[i]$  обозначает тип насекомого  $i$ .

Доступный вам грайдер считывает данные в следующем формате:

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

Если грайдер обнаруживает нарушение протокола взаимодействия, то он выдает сообщение `Protocol Violation: <MSG>`, где `<MSG>` это одна из следующих фраз:

- `invalid parameter`: при вызове функции `move_inside` или функции `move_outside` значение  $i$  не находилось в диапазоне  $0$  and  $N - 1$ , включительно.
- `too many calls`: число вызовов функции к **любой** из трех функций `move_inside`, `move_outside` или `press_button` превысило 40 000.

В противном случае грайдер выводит данные в следующем формате:

- строка 1: возвращаемое значение функции `min_cardinality`
- строка 2:  $q$