



Rarest Insects (最罕見的昆蟲)

有 N 只昆蟲，序號從 0 到 $N - 1$ ，在 Pak Blangkon 的房子周圍跑來跑去。每個昆蟲都有一個 **類型 (type)**，它是一個 0 到 10^9 之間的整數。多種昆蟲可能具有相同的類型。

假設昆蟲按類型分組。我們將**最常見的**昆蟲類型的基數定義為昆蟲數量最多的一組昆蟲的數量。同樣，**最罕見的**昆蟲類型的基數是昆蟲數量最少的一組昆蟲的數量。

例如，假設有 11 只昆蟲，它們的類型是 $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$ 。在這種情況下，**最常見的**昆蟲類型的基數是 3。昆蟲數量最多的組是 類型9 和 類型11，每組由 3 只昆蟲組成。**最罕見的**昆蟲類型的基數是 1。昆蟲數量最少的組是 類型7、類型0 和 類型100，每組由 1 只昆蟲組成。

Pak Blangkon 並不知道任何昆蟲的類型。他有一台帶有單個按鈕的機器，可以提供有關昆蟲類型的一些信息。最初，機器是空的。使用機器時，他可以執行以下三種類型的操作：

1. 將一只昆蟲移到機器內。
2. 將一只昆蟲移出機器外。
3. 按下機器上的按鈕。

每種類型的操作最多可以執行 40 000 次。

每當按下按鈕時，機器都會報告出在機器內的昆蟲**最常見的**昆蟲類型的基數。

您的任務是使用機器確定 Pak Blangkon 家中所有 N 昆蟲中**最罕見的**昆蟲類型的基數。此外，在某些子任務中，您的分數取決於執行的給定類型的最大操作數（有關詳細信息，請參閱子任務部分）。

編程實現細節

你應該編寫以下的子程序：

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

- N ：昆蟲的數量。
- 這個子程序應返回 Pak Blangkon 家中所有 N 只昆蟲中**最罕見的**昆蟲類型的基數。
- 這個子程序只會被調用恰好1次。

上述過程可以調用以下子程序：

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

- i : 要移動到機器內的昆蟲的序號。 i 的值必須在 0 到 $N - 1$ 之間(含)。
- 如果該昆蟲已經在機器內部，則這次調用對機器內的昆蟲的集合沒有影響。但是，它仍然算作一次單獨的調用。
- 這個子程序最多可以調用 40 000 次。

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

- i : 要移動到機器外的昆蟲的序號。 i 的值必須在 0 到 $N - 1$ 之間(含)。
- 如果該昆蟲已經在機器外部，則這次調用對機器內的昆蟲的集合沒有影響。但是，它仍然算作一次單獨的調用。
- 這個子程序最多可以調用 40 000 次。

```
int press_button()
```

- 這個子程序會返回在機器內的昆蟲集中**最常見的**昆蟲類型的基數。
- 這個子程序最多可以調用 40 000 次。
- 樣例評分程式**不是自適應的**。也就是說，在調用 `min_cardinality` 之前所有 N 只昆蟲的類型都已經被固定。

樣例

考慮以下一個場景，假設有 6 只昆蟲，類型分別為 $[5, 8, 9, 5, 9, 9]$ 。子程序 `min_cardinality` 會以下列方式被調用：

```
min_cardinality(6)
```

而子程序 `move_inside`、`move_outside` 及 `press_button` 則可能以下列順序被調用。

調用	返回值	機器內的昆蟲	機器內的昆蟲的類型
		{}	[]
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]

在這一時刻，我們已經有足夠的信息可以得出結論，最稀有昆蟲類型的基數是 1。因此，過程 `min_cardinality` 應該返回 1。

在這個樣例中，`move_inside` 被調用了 7 次，`move_outside` 被調用了 1 次，而 `press_button` 被調用了 6 次。

限制條件

- $2 \leq N \leq 2000$

子任務

1. (10 分) $N \leq 200$
2. (15 分) $N \leq 1000$
3. (75 分) 沒有附加條件。

如果在任何測試用例中，對子程序 `move_inside`、`move_outside` 或 `press_button` 的調用不符合編程實現細節中描述的約束，或者 `min_cardinality` 的返回值不正確，則您的解決方案對該子任務的得分為 0。

令 q 為以下三個值的**最大值**：調用 `move_inside` 的次數、調用 `move_outside` 的次數和調用 `press_button` 的次數。

在子任務 3 中，您可以獲得部分分數。令 m 為該子任務中所有測試樣例中 $\frac{q}{N}$ 的最大值。根據下表計算您對該子任務的分數：

條件	分數
$20 < m$	0 (reported as "Output isn't correct" in CMS)
$6 < m \leq 20$	$\frac{225}{m-2}$
$3 < m \leq 6$	$81 - \frac{2}{3}m^2$
$m \leq 3$	75

樣例評分程式

令 T 是一個由 N 個整數組成的數組，其中 $T[i]$ 是昆蟲 i 的類型。

樣例評分程式按照以下格式讀取輸入：

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

如果樣例評分程式檢測到協議違規，則樣例評分程式的輸出為 `Protocol Violation: <MSG>`，其中 `<MSG>` 是以下之一：

- `invalid parameter`：在其中一次對 `move_inside` 或 `move_outside` 的調用中， i 的值不在 0 到 $N-1$ 之間。
- `too many calls`：對 `move_inside`、`move_outside` 或 `press_button` 的**任意一種**的調用次數超過 40 000。

否則，樣例評分程式會以下面的形式輸出：

- 第 1 行: `min_cardinality` 的返回值。
- 第 2 行: q