



## 最稀有的昆蟲 (Rarest Insects)

有  $N$  隻昆蟲，編號由 0 至  $N - 1$ ，在 Pak Blangkon 的房子周遭爬來爬去。每隻昆蟲都隸屬於一「種類」，為一個 0 至  $10^9$  間的整數，包含 0 與  $10^9$ 。多隻昆蟲可屬同一種類。

假設昆蟲以種類來分群。我們定義「最常見」的昆蟲種類基數 (cardinality) 為最多昆蟲的群之昆蟲數。類似地，「最稀有」的昆蟲種類基數為最少昆蟲的群之昆蟲數。

舉例來說，假設有 11 隻昆蟲，種類為  $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$ 。在這個例子中，「最常見」的昆蟲種類基數為 3。擁有最多昆蟲的群為種類 9 與種類 11，每群都由 3 隻昆蟲組成。「最稀有」的昆蟲種類基數為 1。擁有最少昆蟲的群為種類 7、種類 0 與種類 100，每群都由 1 隻昆蟲組成。

Pak Blangkon 不知道任何一隻昆蟲的種類。他有一台只有單一按鈕的機器，可以提供某些關於這些昆蟲種類的資訊。初始時，這台機器是空的，可以使用三種操作來運行這台機器：

1. 將一隻昆蟲放進這台機器。
2. 將一隻昆蟲拿出這台機器。
3. 按下這台機器上的按鈕。

每種操作至多可使用 40 000 次。

當按鈕被按下後，這台機器會回報在機器裡的昆蟲中「最常見」的昆蟲種類基數。

你的任務是利用這台機器來判斷 Pak Blangkon 的房子周遭所有  $N$  隻昆蟲中「最稀有」的昆蟲種類基數。此外，在某些子任務中，你的分數取決於某些操作被使用的最大次數 (詳細請參閱子任務一節)。

## 實作細節 (Implementation Details)

你應實作下列程序：

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

- $N$ : 昆蟲數量。
- 此程序應回傳 Pak Blangkon 的房子周遭「最稀有」的昆蟲種類基數。
- 此程序被呼叫恰好一次。

上述的程序可以呼叫下列幾個程序：

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

- $i$ : 要被放進機器的昆蟲編號。 $i$  的值必須介於 0 與  $N - 1$  之間，包含 0 與  $N - 1$ 。
- 若這隻昆蟲已經在機器裡，這個呼叫並不會對機器中的昆蟲集合產生任何影響。然而，這個呼叫仍會被視為一次呼叫來累計。
- 此程序可被呼叫至多 40 000 次。

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

- $i$ : 要被移出機器的昆蟲編號。 $i$  的值必須介於 0 與  $N - 1$  之間，包含 0 與  $N - 1$ 。
- 若這隻昆蟲已經在機器外面，這個呼叫並不會對機器中的昆蟲集合產生任何影響。然而，這個呼叫仍會被視為一次呼叫來累計。
- 此程序可被呼叫至多 40 000 次。

```
int press_button()
```

- 此程序回傳機器中「最常見」的昆蟲種類基數。
- 此程序可被呼叫至多 40 000 次。
- 評分程式不會依現狀而改變測資 (not adaptive)。  
意即，所有  $N$  隻昆蟲的種類在 `min_cardinality` 被呼叫前就已經是固定的。

## 範例 (Example)

考慮一個情境，其中有 6 隻種類分別為 [5, 8, 9, 5, 9, 9] 的昆蟲。程序 `min_cardinality` 以下述方式被呼叫：

```
min_cardinality(6)
```

此程序可能如下表所列來呼叫 `move_inside`、`move_outside` 以及 `press_button`。

呼叫	回傳值	機器中的昆蟲	機器中的昆蟲種類
		{}	[]
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]

此刻，已有足夠資訊來推論最稀有的昆蟲種類基數為 1。因此，程序 `min_cardinality` 應回傳 1。

在此範例中，`move_inside` 被呼叫 7 次、`move_outside` 被呼叫 1 次、`press_button` 被呼叫 6 次。

## 限制 (Constraints)

- $2 \leq N \leq 2000$

## 子任務 (Subtasks)

1. (10 points)  $N \leq 200$
2. (15 points)  $N \leq 1000$
3. (75 points) 無額外限制。

若在任何測試中，並未遵照實作細節 (Implementation Details) 中所描述的限制來呼叫 `move_inside`、`move_outside` 與 `press_button`，或是 `min_cardinality` 的回傳值不正確，則對於此子任務你的得分為 0。

令  $q$  為下列三值之「最大值」：呼叫 `move_inside` 的次數、呼叫 `move_outside` 的次數以及呼叫 `press_button` 的次數。

在子任務 3，你可能得到部分分數。令  $m$  為這個子任務中所有測試之  $\frac{q}{N}$  的最大值。對於此子任務，你的得分依下表計算：

Condition	Points
$20 < m$	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

## 範例評分程式 (Sample Grader)

令  $T$  為一個由  $N$  個整數構成的陣列，其中  $T[i]$  為昆蟲  $i$  的種類。

此範例評分程式以下列格式讀取輸入：

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

若範例評分程式偵測到違反限制的情形 (protocol violation)，則輸出的訊息為 `Protocol Violation: <MSG>`，其中 `<MSG>` 為下列之一：

- `invalid parameter`: 某個 `move_inside` 或 `move_outside` 的呼叫， $i$  的值並未介於 0 至  $N-1$  之間，包含 0 與  $N-1$ 。
- `too many calls`: `move_inside`、`move_outside`、`press_button` 任一程序的呼叫次數超過 40 000。

否則，範例評分程式會以下面的格式輸出：

- line 1: `min_cardinality` 的回傳值
- line 2:  $q$