



Najrzadsze owady

W domu Paka Blangkona łązi N owadów indeksowanych od 0 do $N - 1$. Każdy owad ma **typ**, będący liczbą całkowitą z przedziału od 0 do 10^9 włącznie. Różne owady mogą mieć ten sam typ.

Załóżmy, że grupujemy owady według typów. Dla każdego podzbioru owadów definiujemy licznosc **najczęstszego** typu owadów jako największą liczbę owadów mających ten sam typ w tym podzbiorze. Podobnie definiujemy licznosc **najrzadszego** typu owadów, jako liczbę owadów typu, który ma najmniejszą ich liczbę w tym podzbiorze.

Przypuśćmy, że mamy na przykład podzbiór 11 owadów, których typy to $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. W tym przypadku licznosc **najczęstszego** typu owadów, to 3. Tę licznosc osiągają owady o typach 9 i 11. Z kolei licznosc **najrzadszego** typu owadów, to 1. Tę licznosc osiągają typy 7, 0 i 100, ponieważ każdy z tych typów zawiera jednego owada.

Pak Blangkon nie zna typu żadnego z owadów. Dysponuje jednak maszyną z pojedynczym guzikiem, która może dostarczyć pewnej informacji o typach owadów znajdujących się w maszynie. Początkowo maszyna jest pusta. Maszyna udostępnia trzy rodzaje operacji:

1. Włóż owada do maszyny.
2. Usuń owada z maszyny.
3. Naciśnij przycisk.

Każda z tych operacji może być wykonana co najwyżej 40 000 razy.

Po wciśnięciu przycisku maszyna podaje licznosc **najczęstszego** typu owadów, biorąc pod uwagę tylko owady znajdujące się właśnie w maszynie.

Twoje zadanie polega na określeniu za pomocą tej maszyny licznosci **najrzadszego** typu owadów spośród wszystkich N owadów w domu Paka Blangkona. Dodatkowo, w niektórych podzadaniach, Twój wynik będzie zależał od maksimum z liczby operacji wszystkich rodzajów, które wykonasz na maszynie (patrz rozdział Podzadania).

Szczegóły implementacyjne

Powinieneś zaimplementować następującą funkcję

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

- N : liczba owadów.
- Funkcja ta powinna wyznaczyć licznosc **najrzadszego** typu owadów spośród wszystkich N owadów w domu Paka Blangkona.
- Ta funkcja będzie wywołana tylko raz.

Powyzsza funkcja będzie wywoływała następujące procedury i funkcje:

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

- i : indeks owada, który ma być umieszczony w maszynie. Wartość i musi mieścić się w przedziale od 0 do $N - 1$ włącznie.
- Jeśli owad jest w maszynie, to wywołanie tej procedury nie przynosi żadnego efektu, jednak to wywołanie liczy się jako odrębne.
- Ta procedura może być wywołana co najwyżej 40 000 razy.

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

- i : indeks owada, który ma być usunięty z maszyny. Wartość i musi mieścić się w przedziale od 0 do $N - 1$ włącznie.
- Jeśli owada nie ma w maszynie, to wywołanie tej procedury nie przynosi żadnego efektu, jednak to wywołanie liczy się jako odrębne.
- Ta procedura może być wywołana co najwyżej 40 000 razy.

```
int press_button()
```

- Ta funkcja podaje w wyniku wywołania licznosc **najczęstszego** typu owadów, biorąc pod uwagę tylko owady znajdujące się aktualnie w maszynie.
- Ta procedura może być wywołana co najwyżej 40 000 razy.
- Program oceniający **nie jest adaptacyjny**, to znaczy typy wszystkich N owadów są ustalone przed wywołaniem `min_cardinality`.

Przykład

Rozważmy scenariusz, w którym mamy 6 owadów o typach $[5, 8, 9, 5, 9, 9]$. Procedura `min_cardinality` wywołana będzie w następujący sposób:

```
min_cardinality(6)
```

Procedura może wywołać następującą sekwencję `move_inside`, `move_outside` oraz `press_button`.

| Nazwa | Wynik | Owady w maszynie | Typy owadów w maszynie |
|-----------------|-------|--------------------|------------------------|
| | | {} | [] |
| 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] |

W tym momencie mamy już wystarczająco dużo informacji, żeby wywnioskować, że liczność najrzadszego typu owadów, to 1. Zatem funkcja powinna podać wynik 1.

W przykładzie tym procedura `move_inside` jest wywoływana 7 razy, `move_outside` 1 raz, zaś `press_button` 6 razy.

Ograniczenia

- $2 \leq N \leq 2000$

Podzadania

1. (10 punktów) $N \leq 200$
2. (15 punktów) $N \leq 1000$
3. (75 punktów) Brak dodatkowych ograniczeń.

Jeśli w którykolwiek z przypadków testowych wywołania `move_inside`, `move_outside` lub `press_button` nie spełniają wymagań opisanych w "Szczegółach implementacji" lub wynik funkcji `min_cardinality` jest niepoprawny, dostaniesz 0 punktów za to podzadanie.

Niech q będzie **maksimum** z następujących trzech liczb: liczba wywołań `move_inside`, liczba wywołań `move_outside`, liczba wywołań `press_button`.

W podzadaniu 3, możesz uzyskać częściowy wynik. Niech m będzie największą wartością $\frac{q}{N}$ dla wszystkich przypadków testowych tego podzadania. Twój wynik będzie obliczony według następującej tabeli:

| Warunek | Punkty |
|-----------------|--|
| $20 < m$ | 0 (komunikowane jako "Output isn't correct" w CMS) |
| $6 < m \leq 20$ | $\frac{225}{m-2}$ |
| $3 < m \leq 6$ | $81 - \frac{2}{3}m^2$ |
| $m \leq 3$ | 75 |

Przykładowy program oceniający

Niech T będzie tablicą N wartości całkowitych, gdzie $T[i]$ jest typem owada i .

Przykładowy program oceniający czyta wejście w następującym formacie:

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

Jeśli przykładowy program oceniający wykryje niezgodność z założeniami, wynikiem będzie wypisanie `Protocol Violation: <MSG>`, gdzie `<MSG>` jest jednym z następujących komunikatów:

- `invalid parameter`: przy wywołaniu `move_inside` lub `move_outside`, wartość i jest spoza przedziału od 0 do $N - 1$ włącznie.
- `too many calls`: liczba wywołań **którejkolwiek** z procedur lub funkcji `move_inside`, `move_outside`, lub `press_button` przekracza 40 000.

W przeciwnym razie przykładowy program oceniający wypisuje na wyjście:

- wiersz 1: wynik wywołania `min_cardinality`
- wiersz 2: q