Nejvzácnější hmyzák (Pučmeloud)

Po domě *Paka Blangkona* běhá N hmyzáků, očíslovaných od 0 do $N-1$. Každý hmyzák má **typ**, což je celé číslo mezi 0 a 10^9 (včetně). Více hmyzáků může mít stejný typ.

Předpokládejme, že hmyzáky seskupíme podle typu. Četnost **nejhojnějšího** typu hmyzáků definujeme jako počet hmyzáků ve skupině s největším počtem hmyzáků. Analogicky, četnost **nejvzácnějšího** typu hmyzáků definujeme jako počet hmyzáků ve skupině s nejmenším počtem hmyzáků.

Předpokládejme například, že po domě běhá 11 hmyzáků, jejichž typy jsou postupně $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. V tomto případě je četnost **nejhojnějšího** typu rovna 3 a skupiny s největším počtem hmyzáků mají typy 9 a 11, každá z nich obsahuje 3 hmyzáky. Četnost **nejvzácnějšího** typu je rovna 1. Skupiny s nejmenším počtem hmyzáků mají typy 7, 0 a 100, každá z nich obsahuje 1 hmyzáka.

Pak Blangkon nezná typ žádného hmyzáka. Má však přístroj s jedním tlačítkem, který umí změřit nějaké informace o typech hmyzáků. Na začátku je ten přístroj prázdný. *Pak Blangkon* může provádět operace následujících tří typů:

1. Přemístit hmyzáka do přístroje.
2. Vyndat hmyzáka z přístroje.
3. Zmáčknout tlačítko na přístroji.

Každý typ operace může *Pak Blangkon* provést nejvýše $40\ 000$krát.

Kdykoliv *Pak Blangkon* zmáčkne tlačítko, přístroj mu zobrazí četnost **nejhojnějšího** typu hmyzáků v přístroji. Tedy ze všech typů hmyzáků vybere ten, který má v přístojí nejvyšší četnost, a zobrazí počet hmyzáků tohoto typu uvnitř přístroje.

Vaším úkolem je s pomocí přístroje určit četnost **nejvzácnějšího** typu mezi všemi N hmyzáky v *Pakově Blangkonově* domě. V některých podúlohách váš počet bodů bude záviset na největším počtu operací stejného typu, které provedete (viz sekce Podúlohy).

Implementační detaily

Implementujte následující funkci:

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

- N: počet hmyzáků.
- Tato funkce by měla vrátit četnost **nejvzácnějšího** typu hmyzáků mezi všemi N hmyzáky v *Pakově Blangkonově* domě.
- Tato funkce bude volána právě jednou.

V rámci této funkce můžete volat následující funkce:

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

- i: index hmyzáka, kterého chcete přemístit do přístroje. Hodnota i musí být mezi 0 a $N-1$ (včetně).
- Pokud tento hmyzák již uvnitř přístroje je, volání nic neudělá, ale i tak se započítá do limitu.
- Tato funkce může být zavolána nejvýše $40\ 000$ krát.

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

- i: index hmyzáka, kterého chcete vyndat z přístroje. Hodnota i musí být mezi 0 a $N-1$ (včetně).
- Pokud tento hmyzák již vně přístroje je, volání nic neudělá, ale i tak se započítá do limitu.
- Tato funkce může být zavolána nejvýše $40\ 000$ krát.

```
int press_button()
```

- Tato funkce vrátí četnost **nejhojnějšího** typu hmyzáků v přístroji. Tedy ze všech typů hmyzáků vybere ten, který má v přístojí nejvyšší četnost, a zobrazí počet hmyzáků tohoto typu uvnitř přístroje.
- Tato funkce může být zavolána nejvýše $40\ 000$ krát.
- Grader **není adaptivní**. To znamená, že typy všech N hmyzáků jsou určeny již předtím, než je zavolána funkce `min_cardinality`.

Příklad

Uvažujme situaci, kdy po domě běhá 6 hmyzáků, jejichž typy jsou postupně $[5, 8, 9, 5, 9, 9]$. Funkce `min_cardinality` je závolána následovně:

```
min_cardinality(6)
```

Tato funkce může volat `move_inside`, `move_outside`, a `press_button` následovně.

Volání	Návratová hodnota	Hmyzáci v přístroji	Typy hmyzáků v přístroji
		$\{\}$	$[]$
`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	$\{0,1,2,3,4\}$	$[5,8,9,5,9]$

V tuto chvíli máme dostatečné množství informací k tomu, abychom usoudili, že četnost nejvzácnějšího typu hmyzáků je 1, funkce `min_cardinality` by tudíž měla vrátit 1.

V tomto případě je funkce `move_inside` zavolána právě 7krát, `move_outside` jednou a `press_button` je zavolána 6krát.

Omezení

- $2 \leq N \leq 2000$

Podúlohy

1. (10 bodů) $N \leq 200$
2. (15 bodů) $N \leq 1000$
3. (75 bodů) Žádná další omezení.

Pokud v rámci kteréhokoli provádění funkce `min_cardinality` nebudou volání funkcí `move_inside`, `move_outside` nebo `press_button` splňovat podmínky uvedené v sekci

Implementační detaily, anebo návratová hodnota `min_cardinality` bude nesprávná, dostanete za příslušnou podúlohu 0 bodů.

Nechť q je **maximum** z následujících tří hodnot: počet volání `move_inside`, počet volání `move_outside` a počet volání `press_button`.

V podúloze 3 můžete získat částečné body. Nechť m je největší hodnota $\frac{q}{N}$ přes všechna volání funkce `min_cardinality` v podúloze 3. Váš počet bodů za tuto podúlohu pak bude spočítán pomocí následující tabulky:

Podmínka	Body
$20 < m$	0 (v CMS se zobrazí jako "`Output isn't correct`")
$6 < m \leq 20$	$\frac{225}{m-2}$
$3 < m \leq 6$	$81 - \frac{2}{3}m^2$
$m \leq 3$	75

Ukázkový grader

Nechť T je pole obsahující N celých čísel, kde $T[i]$ je typ hmyzáka číslo i.

Ukázkový grader přečte vstup v následujícím formátu:

- řádek 1: N
- řádek 2: $T[0]\ T[1]\ \ldots\ T[N-1]$

Pokud grader detekuje porušení protokolu, výstup graderu bude `Protocol Violation: <MSG>`, kde `<MSG>` je jedno z následujících:

- `invalid parameter`: při volání `move_inside` nebo `move_outside` neleží hodnota i mezi 0 a $N-1$ (včetně).
- `too many calls`: počet volání **některé** z funkcí `move_inside`, `move_outside` či `press_button` překročil $40\,000$.

Jinak má výstup ukázkového graderu následující formát:

- řádek 1: návratová hodnota `min_cardinality`
- řádek 2: q