



Najbolj redki žužki

Okoli hiše se potepa N žužkov, oštevilčeni so od 0 do $N - 1$. Vsak žužek ima svoj **tip**, ki je celo število med 0 in 10^9 , vsebujoče. Več žužkov je lahko istega tipa.

Predpostavimo, da so žužki razporejeni v skupine. Definiramo kardinalnosti **najbolj pogostega** tipa žužkov kot število žužkov v skupini ki je najštevilčnejše. Podobno je kardinalnost **najbolj redkega** tipa žužkov število žužkov najmanj zastopanega tipa žužkov.

Na primer: predpostavimo, da obstaja 11 žužkov, ki so tipov $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. V tem primeru je kardinalnost **najbolj pogostega** tipa enaka 3. Skupini z največ žužki (3), sta tipa 9 in 11. Kardinalnost **najbolj redkega** tipa je 1. Skupine z najmanjšo številčnostjo so 7, 0, in 100, vsaka vsebuje 1 žužka.

Ne poznamo tipa kateregakoli žužka. Imamo pa stroj, ki lahko preko enega gumba poda nekaj informacij o tipih žužkov. Na začetku je stroj prazen. Za upravljanje s strojem so nam na voljo tri operacije:

1. Prestavi žužka v stroj.
2. Prestavi žužka ven iz stroja.
3. Pritisni tipko na stroju.

Vsak tip operacije je lahko klican največ 40 000-krat.

Če se pritisne tipka, stroj izpiše kardinalnost **najbolj pogostega** tipa žužkov, upoštevajoč le žužke znotraj stroja.

Tvoja naloga je ugotoviti kardinalnost **najbolj redkega** tipa žužkov izmed vseh N žužkov, ki so v hiši. Dodatno, v nekaterih podnalogah je točkovanje odvisno od maksimalnega števila izvedenih operacij določenega tipa (glej Podnaloge).

Podrobnosti implementacije

Implementiraj naslednjo funkcijo:

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

- N : število žužkov.
- Funkcija naj vrne kardinalnost **najbolj redkega** tipa žužkov izmed vseh N žužkov v hiši.

- Funkcija je klicana natanko enkrat.

Zgornja funkcija lahko kliče naslednji proceduri oz. funkcijo:

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

- i : indeks žužka, ki naj bo prestavljen v stroj. Vrednost i je med 0 in $N - 1$, vsebuje oče.
- če je dotični žužek že v stroju, ta klic nima učinka na množico žužkov v stroju. Kljub temu, šteje za klic funkcije.
- Proceduro lahko kličemo največ 40 000-krat.

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

- i : indeks žužka, ki ga naj prestavimo iz stroja. Vrednost i je med 0 in $N - 1$, vsebuje oče.
- če je dotični žužek že izven stroja, ta klic nima učinka na množico žužkov v stroju. Kljub temu, šteje za klic funkcije.
- Proceduro lahko kličemo največ 40 000-krat.

```
int press_button()
```

- Funkcija vrne kardinalnost **najbolj pogostega** tipa žužkov, upoštevajoč le žužke v stroju.
- Funkcijo lahko kličemo največ 40 000-krat.
- Ocenjevalnik **se ne prilagaja**. To pomeni, da so tipi vseh N žužkov fiksni, preden se kliče `min_cardinality`.

Primer

Predstavljajmo si scenarij, kjer obstaja 6 žužkov tipov [5, 8, 9, 5, 9, 9]. Funkcija `min_cardinality` je klicana na slednji način:

```
min_cardinality(6)
```

Funkcija lahko izvede naslednje klice `move_inside`, `move_outside`, in `press_button`:

Klic	Povratna vrednost	Žužki v stroju	Tipi žužkov v stroju
		{}	[]
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]

Na tej točki je dovolj informacij, da lahko zaključimo, da je kardinalnost najmanj zastopanega tipa žužkov enaka 1. Zatorej naj funkcija `min_cardinality` vrne 1.

V tem primeru je `move_inside` klicana 7-krat, `move_outside` 1-krat, in `press_button` 6-krat.

Omejitve

- $2 \leq N \leq 2000$

Podnaloge

1. (10 točk) $N \leq 200$
2. (15 točk) $N \leq 1000$
3. (75 točk) Brez dodatnih omejitev.

Če v katerem koli testnem primeru kliči procedur oz. funkcije `move_inside`, `move_outside`, or `press_button` ne ustrezajo zgoraj opisanim omejitvam, bo podnaloga točkovana z 0 točkami.

Naj bo q **maksimum** naslednjih treh vrednosti: število klicev `move_inside`, število klicev `move_outside`, in število klicev `press_button`.

Pri 3. podnalogi lahko dosežeš delno točkovanje. Naj bo m maksimalna vrednost $\frac{q}{N}$ vseh tesnih primerov podnaloge. Točkovanje je po slednji tabeli:

Pogoj	Točke
$20 < m$	0 (CMS vrne "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

Vzorčni ocenjevalnik

Naj bo T polje N celih števil, kjer velja $T[i]$ je tip žužka i .

Vzorčni ocenjevalnik bere vhod naslednje oblike:

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

Če vzorčni ocenjevalnik zazna kršitev protokola, na izhod izpiše `Protocol Violation: <MSG>`, kjer je `<MSG>` eno izmed slednjega:

- `invalid parameter`: pri klicu `move_inside` ali `move_outside`, vrednost i ni med 0 in $N - 1$, vsebuječe.
- `too many calls`: število klicev **katerega koli** izmed `move_inside`, `move_outside`, ali `press_button` presega 40 000.

Sicer je izhod vzorčnega ocenjevalnika naslednje oblike::

- vrstica 1: vračoča vrednost `min_cardinality`
- vrstica 2: q