Retākie insekti

Pa Paka Blangkona māju skraida N insekti, kas sanumurēti no 0 līdz $N-1$. Katram insektam ir **tips**, kas ir vesels skaitlis robežās no 0 līdz 10^9 (ieskaitot). Vairākiem insektiem var būt viens un tas pats tips.

Pieņemsim, ka insekti ir sagrupēti pēc to tipa. Definēsim **biežākā** insektu tipa *kardinalitāti* kā insektu skaitu grupā ar vislielāko insektu skaitu. Līdzīgi, **retāko** insektu tipa kardinalitāti definēsim kā insektu skaitu grupā ar vismazāko insektu skaitu.

Pieņemsim, ka ir 11 insekti, kuru tipi ir $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. Šajā gadījumā **biežākā** insektu tipa kardinalitāte ir 3. Grupas ar vislielāko insektu skaitu ir tipiem 9 un 11, kur katrā grupā ir 3 insekti. Savukārt, **retākā** insektu tipa kardinalitāte ir 1. Grupas ar mazāko insektu skaitu ir tipiem 7, 0 un 100, kur katrā grupā ir 1 insekts.

Paks nezin katra insekta tipu, bet viņam ir mašīna, kas ar vienu pogas nospiedienu ļauj iegūt informāciju par insektu tipiem. Sākotnēji mašīna ir tukša. Lai darbinātu mašīnu, var izmantot trīs veidu darbības:

1. Ievietot insektu mašīnā.
2. Izņemt insektu no mašīnas.
3. Nospiest mašīnas pogu.

Katra veida darbību var izpildīt ne vairāk kā $40\ 000$ reizes.

Kad mašīnas poga tiek nospiesta, mašīna paziņo **biežākā** insektu tipa kardinalitāti, ņemot vērā tikai tobrīd mašīnā ievietotos insektus.

Jūsu uzdevums ir, izmantojot aprakstīto mašīnu, visiem N Paka mājā esošajiem insektiem noteikt **retākā** insektu tipa kardinalitāti. Papildus, dažos apakšuzdevumos, jūsu rezultāts būs atkarīgs no lielākā noteikta veida izpildīto darbību skaita (skat. sadaļu *Apakšuzdevumi*).

Realizācijas detaļas

Ir jārealizē šāda procedūra:

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

- N: insektu skaits.

- Procedūrai jāatgriež visu N Paka mājā esošo insektu **retākā** insektu tipa kardinalitāte.
- Šī procedūra tiek izsaukta tieši vienreiz.

Iepriekšaprakstītā procedūra var veikt šādu procedūru izsaukumus:

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

- i: insekta, kas tiek ievietots mašīnā, indekss. i vērtībai jābūt robežās no 0 līdz $N-1$ (ieskaitot).
- Ja šis insekts jau atrodas mašīnā, tad šis izsaukums nekādi neietekmē insektu kopu mašīnas iekšpusē, bet tas tiks ieskaitīts kā atsevišķs procedūras izsaukums.
- Šo procedūru var izsaukt ne vairāk kā $40\ 000$ reizes.

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

- i: insekta, kas tiek izņemts no mašīnas, indekss. i vērtībai jābūt robežās no 0 līdz $N-1$ (ieskaitot).
- Ja šis insekts jau atrodas ārpus mašīnas, tad šis izsaukums nekādi neietekmē insektu kopu mašīnas iekšpusē, bet tas tiks ieskaitīts kā atsevišķs procedūras izsaukums.
- Šo procedūru var izsaukt ne vairāk kā $40\ 000$ reizes.

```
int press_button()
```

- Šī procedūra atgriež **biežākā** insektu tipa kardinalitāti, ņemot vērā tikai insektus mašīnas iekšpusē.
- Šo procedūru var izsaukt ne vairāk kā $40\ 000$ reizes.
- Vērtētājs ir **neadaptīvs** – t.i., visu N insektu tipi ir fiksēti pirms `min_cardinality` izsaukuma.

Piemērs

Aplūkosim scenāriju, kur ir 6 insekti, un to tipi, attiecīgi: $[5, 8, 9, 5, 9, 9]$. Procedūra `min_cardinality` tiek izsaukta šādi:

```
min_cardinality(6)
```

Procedūra var izsaukt `move_inside`, `move_outside` un `press_button` šādi:

Izsaukums	Atgriežamā vērtība	Insekti mašīnā	Mašīnā esošo insektu tipi
		$\{\}$	$[]$
`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	$\{0,1,2,3,4\}$	$[5,8,9,5,9]$

Šajā brīdī pietiek informācijas, lai secinātu, ka visretākā insektu tipa kardinalitāte ir 1. Tādējādi, procedūrai `min_cardinality` jāatgriež vērtība 1.

Šajā piemērā, `move_inside` tiek izsaukta 7, `move_outside` tiek izsaukta 1 un `press_button` tiek izsaukta 6 reizes.

Ierobežojumi

- $2 \leq N \leq 2000$

Apakšuzdevumi

1. (10 punkti) $N \leq 200$
2. (15 punkti) $N \leq 1000$
3. (75 punkti) Bez papildu ierobežojumiem.

Ja kādā no testiem procedūru `move_inside`, `move_outside` vai `press_button` izsaukumi neatbildīs sadaļā *Realizācijas detaļas* aprakstītajiem ierobežojumiem, vai arī `min_cardinality` atgrieztā vērtība nebūs pareiza, jūs par šo apakšuzdevumu saņemsiet 0 punktus.

Ar q apzīmēcim **lielāko** no šādām trim vērtībām: `move_inside` izsaukumu skaits, `move_outside` izsaukumu skaits un `press_button` izsaukumu skaits.

Trešajā apakšuzdevumā jūs varat iegūt daļēju punktu skaitu. Ar reālu skaitli m apzīmēsim lielāko no vērtībām $\frac{q}{N}$ starp visiem šī uzdevuma apakštestiem. Jūsu iegūto punktu skaits šim apakšuzdevumam tiek aprēķināts izmantojot šādu tabulu:

Nosacījums	Punkti
$20 < m$	0 (CMS paziņojums "`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

Paraugvērtētājs

T ir N veselu skaitļu masīvs, kur i-tā insekta tips ir $T[i]$.

Paraugvērtētājs lasa ievaddatus šādā formātā:

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

Ja paraugvērtētājs atklāj protokola pārkāpumu, tad tā izvads ir: `Protocol Violation: <MSG>`, kur `<MSG>` ir viens no ziņojumiem:

- `invalid parameter`: `move_inside` vai `move_outside` izsaukumā i vērtība neatrodas starp 0 un $N-1$ (ieskaitot).
- `too many calls`: izsaukumu skaits **jebkurai** no procedūrām `move_inside`, `move_outside` vai `press_button` pārsniedz $40\ 000$.

Citādāk, paraugvērtētāja izvads ir šādā formātā:

- 1. rinda: `min_cardinality` atgrieztā vērtība
- 2. rinda: q