



Rečiausi vabzdžiai

N vabzdžių, sunumeruotų nuo 0 iki $N - 1$, bėgioja po Pak Blangkon namą. Kiekvienas vabzdys yra kurios nors **rūšies**, kuri nusakoma sveikuoju skaičiumi nuo 0 iki 10^9 imtinai. Skirtingi vabzdžiai gali būti tos pačios rūšies.

Tarkime, kad vabzdžiai yra sugrupuoti pagal jų rūšis. **Dažniausia** rūšimi nurodytoje vietoje vadinsime tą, kuriai priklauso daugiausia toje vietoje esančių vabzdžių. Analogiškai, **rečiausia** rūšimi vadinsime tą, kuriai priklauso mažiausia vabzdžių.

Pavyzdžiui, yra 11 vabzdžių, kurių rūšys yra $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. Šiuo atveju **dažniausios** rūšies vabzdžių skaičius yra 3. Daugiausiai vabzdžių turinčios rūšys yra 9 ir 11, joms priklauso po 3 vabzdžius. **Rečiausios** rūšies vabzdžių skaičius 1. Mažiausiai vabzdžių turinčios rūšys yra 7, 0 ir 100, joms priklauso po 1 vabzdį.

Pak Blangkon nežino nei vieno vabzdžio rūšies. Jis turi mašiną su vienu mygtuku, kuri gali jam suteikti informacijos apie vabzdžių rūšis. Iš pradžių mašina yra tuščia. Naudojant mašiną galima atlikti trijų rūšių operacijas:

1. Įdėti vabzdį į mašiną.
2. Išimti vabzdį iš mašinos.
3. Spustelėti mašinos mygtuką.

Kiekvieno tipo operacijas galima atlikti daugiausia 40 000 kartų.

Kai mygtukas paspaudžiamas, mašina parodo **dažniausios** mašinos viduje esančių vabzdžių rūšies vabzdžių skaičių. T. y., pirma ji nustato rūšį, kurios vabzdžių mašinoje yra daugiausia. Tada ji parodo tos rūšies vabzdžių, esančių mašinoje, skaičių.

Naudodami mašiną, nustatykite **rečiausios** rūšies vabzdžių skaičių, kai nagrinėjame visus N Pak Blangkon name esančius vabzdžius. Kai kuriose dalinėse užduotyse jūsų gaunami taškai priklausys nuo to, kiek atliksite kiekvieno tipo operacijų (detaliau aprašyta skyrelyje *Dalinės užduotys*).

Realizacija

Parašykite tokią funkciją:

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

- N : vabzdžių skaičius.
- Ši funkcija turi gražinti **rečiausios** rūšies vabzdžių, esančių Pak Blangkon name, skaičių (tarp visų N vabzdžių).
- Ši funkcija iškviečiama lygiai vieną kartą.

Ši funkcija gali kviešti žemiau aprašytas funkcijas:

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

- i : vabzdžio, įdedamo į mašinos vidų, numeris. i reikšmė turi būti nuo 0 iki $N - 1$ imtinai.
- Jei šis vabzdys jau yra mašinos viduje, niekas nesikeičia, t.y. mašinoje lieka tie vabzdžiai, kurie joje buvo. Tačiau iškvietimas vis tiek skaičiuojamas kaip atskiras.
- Šią funkciją galima iškviešti daugiausia 40 000 kartų.

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

- i : vabzdžio, išimamo iš mašinos, numeris. i reikšmė turi būti nuo 0 iki $N - 1$ imtinai.
- Jei šio vabzdžio nėra mašinoje, niekas nesikeičia, t.y. mašinoje lieka vabzdžiai, kurie joje buvo. Tačiau iškvietimas vis tiek skaičiuojamas kaip atskiras.
- Šią funkciją galima iškviešti daugiausia 40 000 kartų.

```
int press_button()
```

- Ši funkcija gražina **dažniausios** mašinos viduje esančių vabzdžių rūšies vabzdžių skaičių.
- Šią funkciją galima iškviešti daugiausia 40 000 kartų.
- Vertinimo programa **nėra adaptyvi**. T.y. visų N vabzdžių rūšys yra fiksuojamos prieš iškviečiant `min_cardinality`.

Pavyzdys

Tarkime, yra 6 vabzdžiai, kurių rūšys yra atitinkamai [5, 8, 9, 5, 9, 9]. Funkcija `min_cardinality` iškviečiama tokiu būdu:

```
min_cardinality(6)
```

Gali iškviešti `move_inside`, `move_outside` ir `press_button` kaip pavaizduota žemiau.

Iškvietimas	Grąžinama reikšmė	Mašinoje esantys vabzdžiai	Mašinoje esančių vabzdžių rūšys
		{}	[]
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]

Atlikę šiuos iškvietimus jau turime pakankamai informacijos, kad nustatytume, jog rečiausios namuose esančių vabzdžių rūšies vabzdžių skaičius yra 1. Taigi, funkcija `min_cardinality` turėtų grąžinti 1.

Šiuo atveju `move_inside` yra iškviečiama 7 kartus, `move_outside` iškviečiama 1 kartą ir `press_button` yra iškviečiama 6 kartus.

Ribojimai

- $2 \leq N \leq 2000$

Dalinės užduotys

1. (10 taškų) $N \leq 200$
2. (15 taškų) $N \leq 1000$
3. (75 taškai) Papildomų ribojimų nėra.

Jei bent vieno testo vykdymo metu `move_inside`, `move_outside` arba `press_button` iškvietimai neatitinka skyrelyje *Realizacija* aprašytų sąlygų arba `min_cardinality` grąžinama reikšmė yra neteisinga, už atitinkamą dalinę užduotį gausite 0 taškų.

Tegu q yra **didžiausia** iš šių trijų reikšmių: `move_inside` iškvietimų skaičiaus, `move_outside` iškvietimų skaičiaus ir `press_button` iškvietimų skaičiaus.

Už 3-iają dalinę užduotį galite gauti dalinius taškus. Tegu m yra maksimali $\frac{q}{N}$ reikšmė tarp visų šiai dalinės užduoties testų. Jūsų gaunami taškai skaičiuojami pagal šią lentelę:

Sąlyga	Taškai
$20 < m$	0 (CMS pranešimas "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

Pavyzdinė vertinimo programa

Tegu T yra masyvas, sudarytas iš N sveikųjų skaičių, kur $T[i]$ yra i -ojo vabzdžio rūšis.

Pavyzdinė vertinimo programa skaito duomenis tokiu formatu:

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

Jei pavyzdinė vertinimo programa nustato, kad jūsų programos daromi funkcijų iškvietimai neatitinka aprašytų sąlygų, ji išveda `Protocol Violation: <žinutė>`, kur `<žinutė>` yra viena iš:

- `invalid parameter`: kviečiant `move_inside` arba `move_outside` pateikta i reikšmė nėra nuo 0 iki $N-1$ imtinai.
- `too many calls`: **bent viena** iš `move_inside`, `move_outside` ir `press_button` funkcijų buvo iškviesta daugiau nei 40 000 kartų.

Priešingu atveju pavyzdinė vertinimo programa duomenis išveda tokiu formatu:

- 1-oji eilutė: `min_cardinality` grąžinta reikšmė
- 2-oji eilutė: q