



## A legritkább bogarak

Pak Blangkon háza körül  $N$  bogár futkározik. A bogarakat 0-tól  $N - 1$ -ig sorszámozzuk. Minden bogárnak van típusa, ami egy legalább 0 és legfeljebb  $10^9$  értékű egész szám. Több bogárnak is lehet ugyanaz a típusa.

A bogarak típusuk alapján csoportokba sorolhatók. Annak a csoportnak az elemszámát, amelyben a legtöbb bogár van, a **leggyakoribb** bogártípus elemszámának nevezzük. Hasonlóképpen, annak a csoportnak az elemszámát, amelyben a legkevesebb bogár van, a **legritkább** bogártípus elemszámának nevezzük.

Például, tegyük fel, hogy 11 bogarunk van, amelyek típusai rendre  $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$ . Ebben az esetben a **leggyakoribb** bogártípus elemszáma 3. A 9-es és a 11-es típusú bogárból is 3 van. A **legritkább** bogártípus elemszáma 1. Ez a 7-es, 0-ás és 100-as típusú csoport, mindegyik 1 bogarat tartalmaz.

Pak Blangkon egyetlen bogár típusát sem ismeri. Van viszont egy gépe, amellyel információt szerezhet a bogarak típusáról. Kezdetben a gép üres. A géppel az alábbi három művelet végezhető:

1. Egy adott bogár gépbe helyezése.
2. Egy adott bogár kivétele a gépből.
3. Kérdező gomb megnyomása a gépen.

Mindegyik művelet legfeljebb 40 000-szer alkalmazható.

A kérdező gomb megnyomásával a gép megadja a benne lévő bogarak közül a **leggyakoribb** bogártípus elemszámát.

A feladatod az összes bogárra vonatkozóan meghatározni a **legritkább** bogártípus elemszámát a gép segítségével. Néhány részfeladat esetén a pontszámod függ a három művelet végrehajtásainak maximális számától (lásd a Részfeladatok fejezetet a további részletekért).

## Megvalósítás

A következő függvényt kell elkészítened:

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

- $N$ : a bogarak száma.
- A függvénynek az összes bogárra vonatkozó **legritkább** bogártípus elemszámát kell visszaadnia.
- A függvényt pontosan egyszer hívják meg.

A fenti függvény a következő három függvényt hívhatja meg a három művelet végrehajtására:

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

- $i$ : a gépbe helyezi az  $i$ . sorszámú bogarat. Az  $i$  értéke legalább 0 és legfeljebb  $N - 1$  lehet.
- Ha az  $i$ . bogár már a művelet előtt a gépben van, akkor a művelet hatástalan, de beleszámít az elvégzett műveletekbe.
- Ez a függvény legfeljebb 40 000-szer hívható.

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

- $i$ : kiveszi a gépből az  $i$ . sorszámú bogarat. Az  $i$  értéke legalább 0 és legfeljebb  $N - 1$  lehet.
- Ha az  $i$ . bogár a művelet előtt nincs a gépben, akkor a művelet hatástalan, de beleszámít az elvégzett műveletekbe.
- Ez a függvény legfeljebb 40 000-szer hívható.

```
int press_button()
```

- Ez a függvény megadja a gépben lévő bogarak közül a **leggyakoribb** bogártípus elemszámát.
- Ez a függvény legfeljebb 40 000-szer hívható.
- Az értékelő **nem adaptív**. Vagyis mind az  $N$  bogár típusa rögzítve van a `min_cardinality` hívása előtt.

## Példa

Tekintsük azt az esetet, amikor 6 bogár van, amelyek típusai rendre [5, 8, 9, 5, 9, 9]. A `min_cardinality` függvényt a következőképpen hívjuk meg:

```
min_cardinality(6)
```

A függvény a `move_inside`, `move_outside` és `press_button` függvényeket az alábbi sorrendben hívja:

Hívás	Visszatérési érték	A gépben lévő bogarak	A gépben lévő bogarak típusai
		{}	[]
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]

Ekkor már elegendő információ áll rendelkezésre, hogy kikövetkeztessük, hogy a legritkább bogártípus elemszáma 1. Ekképpen a `min_cardinality` függvény 1-gyel tér vissza.

Ebben a példában a `move_inside` 7-szer, a `move_outside` 1-szer, a `press_button` pedig 6-szor lett hívva.

## Korlátok

- $2 \leq N \leq 2000$

## Részfeladatok

1. (10 pont)  $N \leq 200$
2. (15 pont)  $N \leq 1000$
3. (75 pont) Nincsenek további megkötések.

Ha bármelyik teszt esetén a `move_inside`, `move_outside` vagy `press_button` függvények valamelyike nem teljesíti a Megvalósítás fejezetben kirótt feltételeket, vagy a `min_cardinality`

függvény visszatérése értéke helytelen, akkor arra a részfeladatra 0 pontot kapsz.

Legyen  $q$  a `move_inside` hívások, a `move_outside` hívások és a `press_button` hívások számának **maximuma**.

A 3. részfeladat esetén részpontszámot kaphatsz. Legyen  $m$  a tesztesetenkénti  $\frac{q}{N}$  értékek maximuma. A részfeladatra kapott pontszámodat az alábbi táblázat szerint számítják:

Feltétel	Pontszám
$20 < m$	0 (A CMS-ben "Output isn't correct" üzenatként szerepel)
$6 < m \leq 20$	$\frac{225}{m-2}$
$3 < m \leq 6$	$81 - \frac{2}{3}m^2$
$m \leq 3$	75

## Mintaértékelő

Legyen  $T$  egy  $N$  elemű, egész számokból álló tömb, ahol  $T[i]$  az  $i$ . bogár típusa.

A mintaértékelő a standard bemenetről a következő formában olvas be:

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

Ha a mintaértékelő protokollhibát érzékel, akkor a mintaértékelő kimenetén a Protocol Violation: <MSG> üzenet jelenik meg, ahol <MSG> a következők egyike lehet:

- invalid parameter:  $i < 0$  vagy  $i > N - 1$  a `move_inside` vagy `move_outside` függvények hívásakor.
- too many calls: a `move_inside`, `move_outside` vagy `press_button` **bármelyikét** 40 000-nél többször hívták meg.

Egyébként a mintaértékelő kimenete a következő formátumú:

- 1. sor: `min_cardinality` függvény visszatérési értéke
- 2. sor:  $q$