



Kõige haruldasemad putukad

Pak Blangkoni majas jooksevad ringi N putukat, nummerdatud 0 kuni $N - 1$. Igal putukal on **tüüp**, mis on täisarv 0 ja 10^9 vahel (k.a.). Mitmel putukal võib olla sama tüüp.

Olgu putukad tüübi järgi rühmitatud. Defineerime **kõige sagedasema** tüübi võimsuse kui putukate arvu suurima arvu putukatega rühmas. Sarnaselt on **kõige haruldasema** tüübi võimsus putukate arv vähima arvu putukatega rühmas.

Olgu meil näiteks 11 putukat, mille tüübid on $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$. Sel juhul on **kõige sagedasema** tüübi võimsus 3. Kõige rohkemate putukatega rühmad on tüüp 9 ja tüüp 11, kummaski 3 putukat. **Kõige haruldasema** tüübi võimsus on 1. Vähima arvu putukatega rühmad on tüüp 7, tüüp 0 ja tüüp 100, igaühes 1 putukas.

Pak Blangkon ei tea ühegi putuka tüüpi. Tal on ühe nupuga masin, mis saab anda natuke infot putukate tüüpide kohta. Alguses on masin tühi. Masinal saab teha kolme toimingut:

1. Pane putukas masinasse.
2. Võta putukas masinast välja.
3. Vajuta masina nuppu.

Iga toimingutüüpi võib teha ülimalt 40 000 korda.

Nuppu vajutades annab masin teada **kõige sagedasema** putukatüübi, võttes arvesse ainult masinas olevaid putukaid.

Sinu ülesanne on teha kindlaks **kõige haruldasema** putukatüübi võimsus kõigi N Pak Blangkoni majas oleva putuka seas. Lisaks sõltub mõnes alamülesandes sinu punktisumma maksimaalsest hulgast üht liiki toimingutest, mida sooritatakse (vt Alamülesanded).

Realisatsioon

Sul tuleb kirjutada järgmine funktsioon:

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

- N : putukate arv.
- See funktsioon peab tagastama **kõige haruldasema** putukatüübi võimsuse kõigi Pak Blangkoni majas oleva N putuka seas.

- Seda funktsiooni kutsutakse välja täpselt üks kord.

Ülalolev funktsioon võib kutsuda välja järgmised funktsioonid:

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

- i : masina sisse pandava putuka indeks. Väärtus i peab olema 0 ja $N - 1$ vahel (k.a.).
- Kui putukas on juba masinas, siis ei mõjuta väljakutse masinas olevaid putukaid, aga loeb siiski eraldi väljakutsena.
- Seda funktsiooni võib välja kutsuda ülimalt 40 000 korda.

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

- i : masinast välja võetava putuka indeks. Väärtus i peab olema 0 ja $N - 1$ vahel (k.a.).
- Kui putukas on juba masinast väljas, siis ei mõjuta väljakutse masinas olevaid putukaid, aga loeb siiski eraldi väljakutsena.
- Seda funktsiooni võib välja kutsuda ülimalt 40 000 korda.

```
int press_button()
```

- See funktsioon tagastab **kõige sagedasema** putukatüübi võimsuse, võttes arvesse vaid masinas olevaid putukaid.
- Seda funktsiooni võib välja kutsuda ülimalt 40 000 korda.
- Hindaja **ei ole adaptiivne**. See tähendab, et kõigi N putuka tüübid on paika pandud enne `min_cardinality` väljakutsumist.

Näide

Vaatame juhtu 6 putukaga, mille tüübid on vastavalt `[5, 8, 9, 5, 9, 9]`. Funktsiooni `min_cardinality` kutsutakse välja järgmiselt:

```
min_cardinality(6)
```

Funktsioon võib välja kutsuda `move_inside`, `move_outside` ja `press_button` järgmiselt.

Kutse	Tagastatav väärtus	Putukad masinas	Masinas olevate putukate tüübid
		{}	[]
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]

Nüüd on võimalik teha kindlaks, et haruldaseima putukatüübi võimsus on 1. Seega peaks funktsioon `min_cardinality` tagastama 1.

Selles näites kutsutakse `move_inside` välja 7 korda, `move_outside` 1 kord ja `press_button` 6 korda.

Piirangud

- $2 \leq N \leq 2000$

Alamülesanded

1. (10 punkti) $N \leq 200$
2. (15 punkti) $N \leq 1000$
3. (75 punkti) Lisapiirangud puuduvad.

Kui ükskõik millises testis ei vasta funktsioonide `move_inside`, `move_outside` ja `press_button` väljakutsed Realisatsioonis välja toodud piirangutele või kui `min_cardinality` tagastatav väärtus

on vale, siis on lahenduse punktisumma selle alamülesande eest 0.

Olgu q **maksimum** järgmisest kolmest väärtusest: `move_inside` väljakutsete arv, `move_outside` väljakutsete arv ja `press_button` väljakutsete arv.

Alamülesandes 3 võid saada osalisi punkte. m on $\frac{q}{N}$ maksimaalne väärtus kõigis antud alamülesande testides. Sinu punktisumma selles alamülesandes arvutatakse järgmiselt:

Tingimus	Punktid
$20 < m$	0 (kuvatakse CMSis kui "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

Näidishindaja

Olgu T massiiv N täisarvuga, kus $T[i]$ on putuka i tüüp i .

Näidishindaja loeb sisendit järgmises vormingus:

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

Kui näidishindaja tuvastab protokollirikkumise, siis on ta väljundiks `Protocol Violation: <SÕNUM>`, kus `<SÕNUM>` on üks järgmistest:

- `invalid parameter`: väljakutses `move_inside` või `move_outside` ei ole i väärtus 0 ja $N - 1$ vahel (k.a.).
- `too many calls`: mõne funktsiooni (`move_inside`, `move_outside` või `press_button`) väljakutsete arv ületab 40 000.

Vastasel juhul on näidishindaja väljund järgmises vormingus:

- rida 1: `min_cardinality` tagastatav väärtus
- rida 2: q