

D. Tautas skaitīšana (census)

Laika ierobežojums: 1 sekundes

Atmiņas ierobežojums: 128 MiB

Mazāk zināms fakts par Čezenatiko ir tas, ka tur mājo slepena kopiena, kurā ir N informātiķes. Šī kopiena patiešām ir ļoti slepena: neviena tās dalībniece nepazīst nevienu citu. Katrai dalībniecei ir unikāls ID — nenegatīvs vesels skaitlis I .

Vienīgā saziņā starp dalībniecēm notiek netieši: ar krītu uzrakstītu skaitļu palīdzību dažādās vietās pilsētā. Reizi 100 gados kopiena veic tautas skaitīšanu, lai saskaitītu kopienas dalībnieces. Pēc tautas skaitīšanas beigām katrai dalībniecei jāzina kopējais kopienas dalībnieču skaits.

Tautas skaitīšana notiek vairākas dienas. Katru dienu katra dalībniece, kura joprojām piedalās procesā, izvēlas un izpilda tieši vienu darbību: **lasīt**, **rakstīt** vai **pārtraukt dalību**.

- Ja dalībniece izvēlas **lasīt**, viņa izvēlas vietu P . Dienas laikā viņa apmeklē vietu P un nolasa tur uzrakstīto skaitli.
- Ja dalībniece izvēlas **rakstīt**, viņa izvēlas vietu P un skaitli V . Vēlu vakarā viņa apmeklē vietu P un nomaina tur uzrakstīto skaitli pret V . Tā kā jau ir tumšs, viņa pirms jaunā skaitļa uzrakstīšanas nevar nolasīt veco skaitli.
- Ja dalībniece izvēlas **pārtraukt dalību**, turpmākajās dienās viņa vairs neveic nekādas darbības.

Ja viena dalībniece redzētu citu dalībnieci rakstām skaitli, viņa varētu to atpazīt. Tāpēc divām vai vairākām dalībniecēm ir stingri aizliegts vienā un tajā pašā dienā izvēlēties rakstīt vienā un tajā pašā vietā. (Lasīšanai šāda ierobežojuma nav, jo to var izdarīt nemanāmi.)

Ja viena vai vairākas dalībnieces lasa vietā, kurā cita dalībniece tajā pašā dienā vēlas rakstīt, tad rakstīšana notiek tikai tad, kad atbilstošās dalībnieces ir pabeigušas lasīšanu.

Kā kopienai jāplāno tautas skaitīšanas process, lai samazinātu dienu skaitu, līdz visas dalībnieces uzzina pareizo dalībnieču skaitu?

Implementācija

⇒ Šis ir interaktīvs uzdevums, kurā vienlaikus tiks palaists nezināms skaits tavas programmas instanču ($1 \leq N \leq 100$). Katra instance simulē vienu kopienas dalībnieci.

Kopā ir 10^{18} dažādu vietu. Vietas numuram P jāapmierina nosacījums $0 \leq P < 10^{18}$. Sākotnēji visās vietās uzrakstītā vērtība ir $V = 0$.

Jaunajai vērtībai V , kas tiek uzrakstīta kādā vietā, vienmēr jābūt veselam skaitlim, kuram izpildās $0 \leq V \leq 10^9$. Vairumā apakšuzdevumu V drīkst būt tikai 0 vai 1. Sīkāku informāciju skatīt sadaļā Vērtēšana.

Kad tavas programmas instance sāk darbu, tai vispirms jānolasa rindiņa ar diviem veseliem skaitļiem, I un M ($0 \leq I \leq M - 1$): šīs dalībnieces unikālais ID un kopējais iespējamo ID skaits. Katrā testa gadījumā visas instances saņems vienādu M vērtību un atšķirīgas I vērtības. Ņem vērā, ka daži ID var nebūt piešķirti nevienai dalībniecei.

Pēc tam katrā tautas skaitīšanas procesa dienā tavai programmai jāizvēlas darbība, ko tā vēlas veikt, un attiecīgi jāizvada rindiņa:

Darbība	Nozīme
$r \ P$	Lasīt vietu P . Pēc šīs rindiņas izvadīšanas tavai programmai jānolasa rindiņa ar pašreizējo vērtību, kas uzrakstīta vietā P .
$w \ P \ V$	Rakstīt vietā P jauno vērtību V . Ja vairākas instances vienā dienā raksta vienā un tajā pašā vietā P , saņemsi vērtējumu <i>Not correct</i> . Izņemot piemērus un 3. apakšuzdevumu, tev jāraksta $0 \leq V \leq 1$; skatīt sadaļu Vērtēšana.
$! \ N$	Atbildēt un beigt : paziņot, ka ir N dalībnieces, un pārtraukt dalību tautas skaitīšanā. Pēc atbildēšanas tavai programmai jābeidz darbs . (Ņem vērā, ka citas tavas programmas instances var turpināt darboties vēl vairākas dienas pirms tās atbild un beidz darbu.)

Ja kāda tavas programmas instance atbild ar nepareizu N vērtību, pārkāpj protokolu, izmanto vairāk nekā 500 dienas vai pārsniedz laika/atmiņas ierobežojumu vienai instancei, tavs iesūtījums attiecīgajā testa gadījumā tiks vērtēts kā *Not correct*.

Pretējā gadījumā tava programma testa gadījumā būs (*Partially*) *Correct* un tiks vērtēta pēc D vērtības: maksimālā dienu skaita, kas jebkurai instancei bija nepieciešams atbildes sniegšanai. Lai iegūtu pilnu punktu skaitu, tev jāatrisina katrs tests ar $D \leq 61$ un $V \leq 1$. Sīkāku informāciju skatīt sadaļā Vērtēšana.

Izvades bufera iztukšošana. Ja neizmanto uzdevumā pieejamās koda sagatavnes, pārliecinies, ka pēc katras rindiņas izvadīšanas iztukšo standarta izvades buferi. Pretējā gadījumā tava programma var tikt vērtēta kā *Not correct*. Python valodā tas notiek automātiski, ja rindiņu nolasīšanai izmantojat `input()`. C++ valodā `cout << endl`; iztukšo buferi un papildus izvada jaunu rindiņu; ja izmanto `printf`, jālieto `fflush(stdout)`.

Ierobežojumi

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Katrai instancei ir pieejamas ne vairāk kā 500 dienas.

Vērtēšana

Tava programma tiks pārbaudīta uz vairākiem testiem, kas sagrupēti apakšuzdevumos. Lai iegūtu punktus par apakšuzdevumu, pareizi jāatrisina visi tajā iekļautie testi.

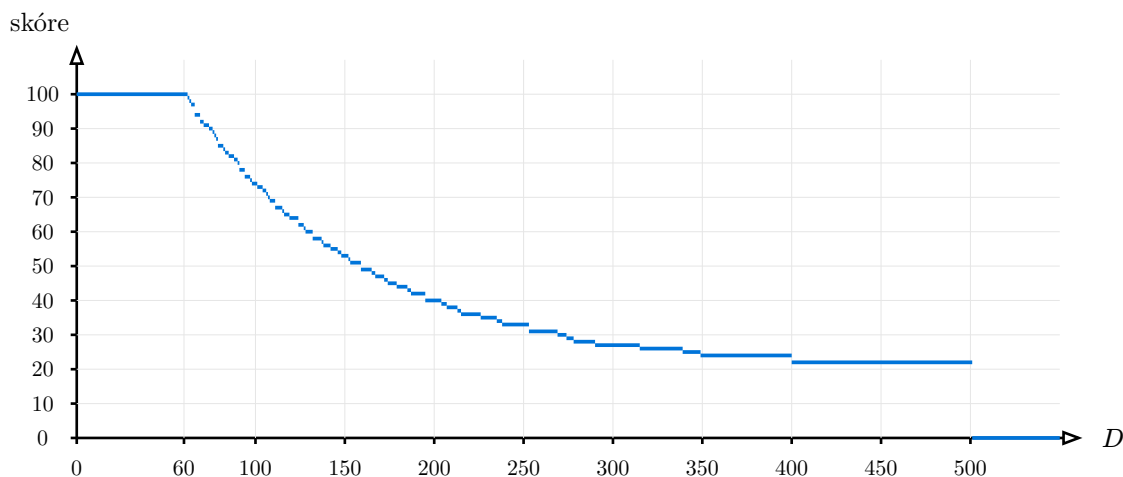
- Apakšuzdevums 0 [0 punkti]:** Piemēri (drīkst rakstīt jebkuru veselu skaitli $0 \leq V \leq 1\,000\,000\,000$).
- Apakšuzdevums 1 [11 punkti]:** $M \leq 100$, un N dalībnieču ID ir $0, 1, \dots, N - 1$.
- Apakšuzdevums 2 [12 punkti]:** $1 \leq N \leq 2$.
- Apakšuzdevums 3 [22 punkti]:** $M \leq 8000$, un drīkst rakstīt jebkuru veselu skaitli $0 \leq V \leq 1\,000\,000\,000$.
- Apakšuzdevums 4 [55 punkti]:** Nav papildu ierobežojumu.

Apakšuzdevumos 1, 2 un 4 katrā darbībā „Rakstīt“ drīkst rakstīt tikai $V = 0$ vai $V = 1$.

Ja X_s ir maksimālais punktu skaits apakšuzdevumam s (apakšuzdevumi parādīti augstāk), un D_s ir lielākais dienu skaits, ko kāda no tavas programmas instancēm izmanto kādā apakšuzdevuma s testā, tad:

$$\text{score}_s = \begin{cases} X_s & \text{ja } D_s \leq 61 \\ X_s \cdot (0.2 + 0.8 \cdot 1.01^{(60-D_s)}) & \text{ja } 61 < D_s \leq 500 \\ 0 & \text{ja } 500 < D_s. \end{cases}$$

score_s vērtība katram apakšuzdevumam tiek noapaļota līdz tuvākajam veselajam skaitlim, un tavs kopējais punktu skaits ir šo vērtību summa. Lai iegūtu pilnu punktu skaitu par uzdevumu, katrā testā jāizpildās $D \leq 61$ un $V \leq 1$.



Attēls 1: Kopējais punktu skaits, pieņemot, ka katrs apakšuzdevums tiek atrisināts ar vienu un to pašu maksimālo D .

Piemēri

Pirmais piemērs. Katrs kolonnu pāris parāda saziņu starp vērtētāju un vienu instanci.

Vērt.	Inst. 0	Vērt.	Inst. 1	Vērt.	Inst. 2	Vērt.	Inst. 3	Vērt.	Inst. 4
0 100		1 100		2 100		3 100		4 100	
	w 12 1		w 50 1		w 99 0		w 7 1		r 5
								0	
	r 50		r 7		r 12		w 1 1		! 5
1		1		1					
	! 5		r 1		w 0 0		! 5		
		1							
		! 5			! 5				

Otrais piemērs.

Vērtētājs	Instance 0
0 8000	
	w 0 0
	w 1 1
	r 2
1	
	! 2

Vērtētājs	Instance 1
3 8000	
	w 2 1
	r 1
0	
	r 2
1	
	r 1
1	
	! 2

Skaidrojums

Pirmais piemērs. Mums ir $N = 5$ dalībnieces ar ID secīgi 0, 1, 2, 3, 4 un $M = 100$ (atbilst 1., 3. un 4. apakšuzdevumam). Instance i atbilst dalībniecei ar ID i . Iepriekš parādītā notikumu izpilde ir tikai viena iespējama korekta operāciju secība, un tā **nav** domāta kā efektīva vai saprātīga stratēģija; tā parādīta tikai, lai ilustrētu protokola darbību.

Otrais piemērs. Mums ir $N = 2$ dalībnieces ar ID 0 un 3, un $M = 8000$ (atbilst 2., 3. un 4. apakšuzdevumam). Pirmajā dienā dalībniece ar ID 0 vietā 0 ieraksta 0 (nekas nemainās), un dalībniece ar ID 3 vietā 2 ieraksta 1.

Vieta	0	1	2	3	4	...
Vērtība	0	0	1	0	0	...

Otrajā dienā ID 0 vietā 1 ieraksta 1 un ID 3 nolasa to pašu vietu. Ņem vērā, ka lasīšana notiek dienas laikā — pirms rakstīšanas vakarā. Tāpēc ID 3 joprojām redz 0.

Vieta	0	1	2	3	4	...
Vērtība	0	1	1	0	0	...

Trešajā dienā abas nolasa vietu 2, kur ir uzrakstīts 1.

Ceturtajā dienā ID 0 atbild, ka ir 2 dalībnieces (pareizi), kamēr ID 3 nolasa vietu 1 un redz skaitli 1. ID 0 uzreiz pēc tam beidz darbu un turpmākajās dienās vairs nepiedalās.

Visbeidzot, dienā $D = 5$ atlikusī dalībniece arī pareizi atbild, ka $N = 2$.

Testēšana

Lai atvieglotu tava risinājuma testēšanu, ir pieejams vienkāršs rīks, ko var lejupielādēt no CMS. Rīka izmantošana nav obligāta. Ņem vērā, ka oficiālais vērtētājs CMS sistēmā atšķiras no testēšanas rīka.

Lai izmantotu rīku, ir nepieciešams ievaddatu fails. Var izmantot piedāvātos ievades piemērus `census.input0.txt` un `census.input1.txt`, vai izveidot savu. Ievades failam jābūt ar dalībnieču skaitu N un iespējamo ID skaitu M , kam seko rindiņā ar N skaitļiem, kas norāda kopienas dalībnieču ID.

Python programmām, piemēram, `census.py` (parasti palaiž kā `pypy3 census.py`), testēšanas rīku palaid šādi:

```
python3 testing_tool.py pypy3 census.py < census.input0.txt
```

C++ programmām vispirms nokompilē savu risinājumu:

```
g++ -DEVAL -std=gnu++20 -O2 -pipe -static -s -o census census.cpp
```

un tad palaid testēšanas rīku:

```
python3 testing_tool.py ./census < census.input0.txt
```

Ņem vērā, ka šajā uzdevumā standarta izvade tiek izmantota saziņai ar vērtētāju, tāpēc to nevajadzētu izmantot koda atklādošanai. Tā vietā vari izmantot standarta kļūdu izvadi (`stderr`). C++ valodā vari izmantot `cerr << msg << endl;`. Python valodā vari izmantot `print(msg, file=sys.stderr)`.

Testēšanas rīks nolasīs un parādīs šos `stderr` ziņojumus kopā ar visiem tavas programmas instanču veiktajiem pieprasījumiem. Ņem vērā, ka tehnisku iemeslu dēļ tie var parādīties nedaudz nesinhroni.