

D. Rahvaloendus (census)

Ajalimiit: 1 sekundit

Mälulimiit: 128 MiB

Vähetuntud fakt Cesenatico kohta on, et siin tegutseb salaselts, mille liikmeteks on N naisinformaatikut. See selts on tõesti väga salajane: mitte ükski liige ei tunne ühtegi teist liiget. Igal liikmel on unikaalne ID: mittenegatiivne täisarv I .

Ainus suhtlus liikmete vahel on kaudne: see käib läbi arvude, mis on linna erinevates kohtades kriidiga seina peale kirjutatud. Iga 100 aasta tagant korraldab selts rahvaloenduse, et oma liikmed kokku lugeda. Kui rahvaloendus läbi on, siis peaks iga seltsiliige teadma, kui mitu liiget seltsis on.

Rahvaloendus toimub mitme päeva vältel. Iga päev teeb iga liige, kes ikka veel protsessist osa võtab, ühe järgnevast kolmest tegevusest: **loeb**, **kirjutab**, või **lõpetab** osa võtmise.

- Kui liige otsustab **lugeda**, siis valib ta mingi asukoha P . Päeval külastab ta asukohta P ja loeb seina pealt arvu, mis sinna kirjutatud on.
- Kui liige otsustab **kirjutada**, siis valib ta asukoha P ja arvu V . Hilisõhtul külastab ta asukohta P ja kirjutab seina peale praeguse arvu asemele arvu V . Kuna juba on pime, siis ei näe ta enne seinale kirjutamist seinal olevat vana arvu.
- Kui liige otsustab **lõpetada**, siis ei tee ta järgnevatel päevadel enam mitte ühtegi tegevust.

Kui üks liige näeb teist seina peale numbrit kirjutamas, siis on oht, et ta tunneb ta ära. Seega on kahel või enamal liikmel samal päeval samasse asukohta kirjutamine rangelt keelatud. (Lugemise jaoks ei ole mingit sellist piirangut, kuna seda on võimalik diskreetselt teha.)

Kui üks või mitu liiget loevad asukohast, kuhu teine liige soovib samal päeval midagi kirjutada, siis toimuvad kõik lugemised enne kirjutamist.

Kuidas peaks salaselts oma rahvaloendust planeerima, et minimeerida päevade arv, enne kui kõik liikmed teavad täpset liikmete arvu?

Implementatsioon

⇒ See on interaktiivne ülesanne, kus tundmatu arv ($1 \leq N \leq 100$) eksemplare sinu programmist jooksevad paralleelselt. Iga eksemplar simuleerib ühte salaseltsi liiget.

Asukohti on kokku 10^{18} tükki. Iga asukoha arv P peab rahuldama tingimust $0 \leq P < 10^{18}$. Alguses on igasse asukohta kirjutatud arv $V = 0$.

Asukohta kirjutatud uus väärtus V peab alati olema täisarv, kusjuures $0 \leq V \leq 10^9$. Enamus alam-ülesannetes võib V olla ainult 0 või 1. Vaata lähemalt Hindamise sektsioonist.

Kui sinu programmi mingi eksemplar alustab, siis peab see esiteks lugema sisse ühe rea kahe täisarvuga I ja M ($0 \leq I \leq M - 1$): need on vastavalt praeguse liikme unikaalne ID ja võimalike ID-de koguarv.

Igas testis saavad kõik eksemplar sama M väärtuse ja erineva I väärtuse. Pane tähele, et võib olla ID-sid, mis ei kuulu ühelegi liikmele.

Edasi peab programm iga rahvaloenduse päeva jaoks valima tegevuse, mida see sooritada soovib, ja väljastama vastava rea:

| Tegevus | Tähendus |
|-------------|--|
| $r \ P$ | Loe asukohta P . Pärast selle rea väljastamist tuleb programmil lugeda sisendist üks rida, mis on praegune väärtus, mis on kirjutatud asukohta P . |
| $w \ P \ V$ | Kirjuta asukohta P uus väärtus V . Kui mitu eksemplari proovivad samal päeval samasse asukohta kirjutada, siis saab lahendus hindeks <i>Not correct</i> . Alati tuleb kirjutada väärtus $0 \leq V \leq 1$, välja arvatud näidetes ja alamülesandes 3; vaata lähemalt Hindamise seksioonist. |
| $! \ N$ | Vasta ja lõpeta : raporteeri, et seltsis on täpselt N liiget, ja lõpeta rahvaloendusest osa võtmine. Pärast vastamist peab sinu programm normaalselt töö lõpetama . (Pane tähele, et muud eksemplarid sinu programmist võivad oma tööd veel jätkata nii mitu päeva, kui vaja, enne kui nad vastavad ja töö lõpetavad.) |

Kui ükski eksemplar sinu programmist vastab vale N väärtusega, rikub suhtlusprotokoll, kasutab rohkem kui 500 päeva, või ületab (ühe protsessi kohta antud) aja/mälulimiiti, siis saab sinu lahendus antud testis hindeks *Not correct*.

Muul juhul saab lahendus selles testis hindeks (*Partially*) *Correct* ja saab punkte vastavalt väärtusele D : see on maksimaalne arv päevi, mis ühelgi eksemplaril kulus, et vastus leida. Täispunktide saamiseks tuleb iga test lahendada, kasutades $D \leq 61$ ja $V \leq 1$. Vaata lähemalt Hindamise seksioonist.

Puhvrite tühjendamine. Kui sa ei kasuta ühte antud mallidest, siis kontrolli, et tühjendad standardväljundi puhvri pärast iga rea väljastamist. Vastasel juhul võib programm saada hindeks *Not correct*. Pythonis juhtub see automaatselt, kui kasutada sisendi lugemiseks funktsiooni `input()`. C++is `cout << endl`; väljastab reavahetuse ja tühjendab puhvri; kui kasutad selle asemel `printf`-i, siis kasuta `fflush(stdout)`.

Piirangud

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Võid kasutada ülimalt 500 päeva.

Hindamine

Sinu programmi testitakse mitmel testil, mis on grupeeritud alamülesanneteks. Et alamülesande eest punkte saada, pead korrektselt lahendama kõik sellesse kuuluvad testid.

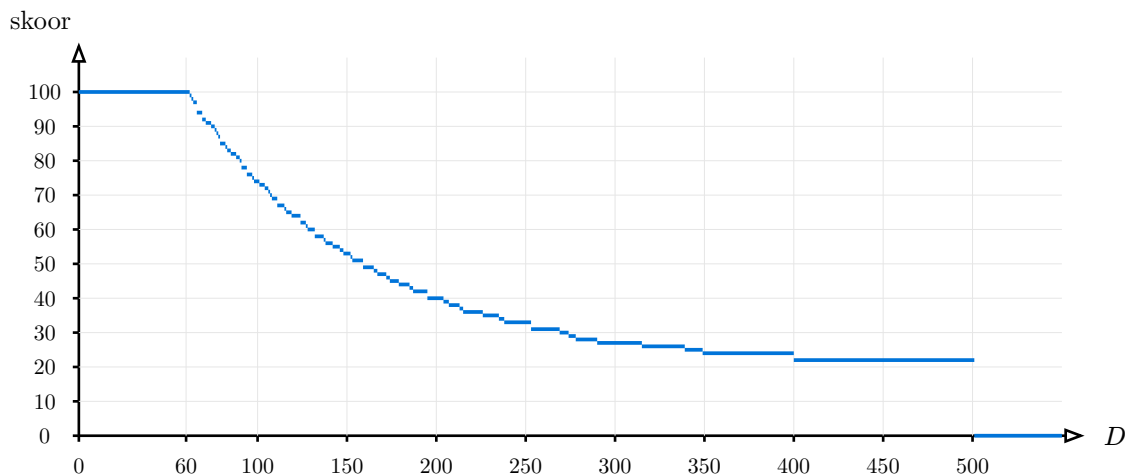
- **Alamülesanne 0 [0 punkti]**: Näited (võid kirjutada ükskõik milliseid arve $0 \leq V \leq 1\,000\,000\,000$).
- **Alamülesanne 1 [11 punkti]**: $M \leq 100$, ja N liikmel on ID-d $0, 1, \dots, N - 1$.
- **Alamülesanne 2 [12 punkti]**: $1 \leq N \leq 2$.
- **Alamülesanne 3 [22 punkti]**: $M \leq 8000$, ja võid kirjutada ükskõik milliseid arve $0 \leq V \leq 1\,000\,000\,000$.
- **Alamülesanne 4 [55 punkti]**: Lisapiirangud puuduvad.

Alamülesannetes 1, 2 ja 4 tohid kirjutamise tegevuses kirjutada ainult väärtusi $V = 0$ või $V = 1$!

Olgu X_s maksimaalne punktisumma alamülesande s jaoks (toodud üleval), ja D_s suurim arv päevi, mis ühelgi sinu programmi eksemplaril üheski selle alamülesande testis kulus. Siis:

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

Arv skoor_s ümardatakse iga alamülesande jaoks lähima täisarvuni, ja sinu kogupunktisumma on nende summa. Selleks, et saada ülesandest täispunktid, tuleb igas testis kasutada $D \leq 61$ ja $V \leq 1$.



Joonis 1: Kogupunktid, kui kõik alamülesanded on lahendatud sama maksimaalse D väärtusega.

Näited

Esimene näide. Iga paar tulpasid näitab suhtlust hindamisprogrammi ja ühe lahenduse eksemplari vahel.

| Hnd. | Lah. 0 | Hnd. | Lah. 1 | Hnd. | Lah. 2 | Hnd. | Lah. 3 | Hnd. | Lah. 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 | | | | |

Teine näide.

| Hindaja | Lahendus 0 |
|---------|------------|
| 0 8000 | |
| | w 0 0 |
| | |
| | w 1 1 |
| | |
| | r 2 |
| 1 | |
| | ! 2 |
| | |
| | |

| Hindaja | Lahendus 1 |
|---------|------------|
| 3 8000 | |
| | w 2 1 |
| | |
| | r 1 |
| 0 | |
| | r 2 |
| 1 | |
| | r 1 |
| 1 | |
| | ! 2 |

Selgitus

Esimene näide. Siin on $N = 5$ liiget järjestikuste ID-dega 0, 1, 2, 3, 4 ja $M = 100$ (lubatud sisend alamülesannete 1, 3 ja 4 jaoks). Eksemplar i vastab liikmele ID-ga i . Üaltoodud suhtlus on vaid üks võimalik lubatud tegevuste jada ja see **ei ole** mõeldud olema näide efektiivsest või mõistlikust strateegiast; see illustreerib ainult seda, kuidas protokoll töötab.

Teine näide. Siin on $N = 2$ liiget, ID-dega 0 ja 3, ja $M = 8000$ (lubatud sisend alamülesannetes 2, 3 ja 4). Esimesel päeval kirjutab liige ID-ga 0 väärtuse 0 asukohta 0 (s.t. ei muuda seda), ja liige ID-ga 3 kirjutab väärtuse 1 asukohta 2.

| asukoht | 0 | 1 | 2 | 3 | 4 | ... |
|---------|---|---|---|---|---|-----|
| väärtus | 0 | 0 | 1 | 0 | 0 | ... |

Teisel päeval kirjutab ID 0 väärtuse 1 asukohta 1, ja ID 3 loeb seda sama asukohta. Pane tähele, et see lugemine juhtub päevasel ajal, enne kui öhtul kirjutamine toimub. Seega ID 3 näeb ikka veel väärtust 0.

| asukoht | 0 | 1 | 2 | 3 | 4 | ... |
|---------|---|---|---|---|---|-----|
| väärtus | 0 | 1 | 1 | 0 | 0 | ... |

Kolmandal päeval loevad mõlemad eksemplarid asukohta 2, kuhu on kirjutatud väärtus 1.

Neljandal päeval vastab ID 0, et seltsis on 2 liiget (õige), samas ID 3 loeb väärtuse 1 asukohast 1. ID 0 lõpetab siis töö ja ei osale ülejäänud päevades.

Viimaks päeval $D = 5$ vastab allesjäänud liige ka korrektselt $N = 2$.

Testimine

Lahenduse testimise hõlbustamiseks on saadaval lihtne tööriist, mis on võimalik CMS-ist alla laadida. Selle tööriista kasutamine on valikuline. Pane tähele, et CMS-is kasutatav ametlik hindamisprogramm erineb testimistööriistast.

Tööriista kasutamiseks on vaja sisendfaili. Võid kasutada antud näidisfaile `census.input0.txt` ja `census.input1.txt`, või võid sisendi ise luua. Sisendfaili esimesel real peab olema kaks täisarvu N ja M : vastavalt liikmete arv ja võimalike ID-de arv. Teisel real peab olema N täisarvu: iga seltsiliikme ID-d.

Pythoni programmide jaoks, näiteks `census.py` (mida jooksutaksid tavaliselt kui `pypy3 census.py`) jookсутa testimistööriista järgnevalt:

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

C++ programmide jaoks kõigepealt kompileeri oma lahendus:

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

ja seejärel jookсутa testimistööriista:

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

Pane tähele, et selles ülesandes kasutatakse standardväljundit hindamisprogrammiga suhtlemiseks, seega ei tohiks seda debugimise jaoks kasutada. Selle asemel võib kasutada veaväljundit (`stderr`). C++is võib kasutada `cerr << msg << endl;`. Pythonis võib kasutada `print(msg, file=sys.stderr)`. Testimistööriist loeb neid veaväljundi sõnumeid ja kuvab neid koos kõikide programmi eksemplaride sooritatud päringutega.

Pane tähele, et tehnilistel põhjustel võivad need ilmuda üksteise suhtes natuke vales järjekorras.