

D. Siyahıyaalma (census)

Zaman məhdudiyyəti: 1 saniyə

Yaddaş məhdudiyyəti: 128 MiB

Çezenatiko barədə az bilinən faktlardan biri odur ki, bura N qadın informatikdən ibarət gizli cəmiyyətin evidir. Bu cəmiyyət doğrudan da çox gizlidir; heç bir üzv digərini tanımır. Hər bir üzvün unikal ID-si var: mənfi olmayan tam ədəd I .

Üzvlər arasındakı yeganə əlaqə birbaşa deyil, şəhərin fərqli yerlərində təbaşirlə cızma-qara edilmiş rəqəmlər vasitəsilədir. Hər 100 ildən bir cəmiyyət öz üzvlərini saymaq üçün siyahıyaalma keçirir. Siyahıyaalma bitdikdən sonra hər bir üzv cəmiyyətdəki üzvlərin ümumi sayını bilməlidir.

Siyahıyaalma bir neçə gün davam edir. Hər gün, prosesdə hələ də iştirak edən hər bir üzv düz bir hərəkət seçib icra edəcək: **oxumaq**, **yazmaq** və ya iştirakı **dayandırmaq**.

- Əgər üzv **oxumaq** seçərsə, o, P məkanını seçir. Gündüz vaxtı o, P məkanına baş çəkir və orada yazılan rəqəmi oxuyur.
- Əgər üzv **yazmaq** seçərsə, o, P məkanı və V ədədini seçir. Axşam gec saatlarda, o, P məkanına gedir və orada yazılmış ədədi V ilə dəyişdirir. Artıq qaranlıq olduğundan, o, yenisini yazmazdan əvvəl köhnə rəqəmi oxuya bilmir.
- Əgər üzv **dayandırmaq** seçərsə, o, növbəti günlərdə heç bir hərəkət etmir.

Əgər bir üzv digərinin rəqəm yazdığını görərsə, onu tanıya bilər. Buna görə də, iki və ya daha çox üzvün eyni gündə eyni məkanda yazmaq seçməsi qəti qadağandır. (Oxumaq üçün belə bir məhdudiyyət yoxdur, çünki bunu gizlicə etmək olar.)

Əgər eyni gündə bir və ya bir neçə üzv başqa bir üzvün yazmaq istədiyi məkandan oxuyarsa, bütün oxumalar yazmadan əvvəl baş verir.

Cəmiyyət hər kəsin düzgün üzv sayını öyrəndiyi günlərin sayını minimuma endirmək üçün siyahıyaalma prosesini necə planlaşdırmalıdır?

İcra

⇒ Bu, proqramınızın naməlum sayda nüsxəsinin ($1 \leq N \leq 100$) eyni vaxtda icra ediləcəyi interaktiv məsələdir. Hər bir nüsxə cəmiyyətin bir üzvünü simulyasiya edir.

10^{18} məkan var. Məkanın P nömrəsi $0 \leq P < 10^{18}$ şərtini ödəməlidir. Başlanğıcda bütün məkanlarda yazılan dəyər $V = 0$ -dir.

Məkanda yazılan yeni V dəyəri həmişə $0 \leq V \leq 10^9$ şərtini ödəyən tam ədəd olmalıdır. Əksər alt tapşırıqlarda V yalnız 0 və ya 1 ola bilər. Daha çox detal üçün Qiymətləndirmə bölməsinə bax.

Proqramının bir nüsxəsi başlayanda, əvvəlcə I və M ($0 \leq I \leq M - 1$) iki tam ədədi olan sətiri oxumalıdır: bu nüsxənin təmsil etdiyi cəmiyyət üzvlərinin unikal ID-si və mümkün olan ID-lərin ümumi sayı. Hər test daxilində bütün nüsxələr eyni M dəyərini və fərqli I dəyərlərini alacaqlar. Nəzərə al ki, heç bir üzvə təyin olunmayan ID-lər də ola bilər.

Sonra, siyahıyaalma prosesinin hər günü üçün proqramın etmək istədiyi hərəkəti seçməli və müvafiq olaraq bir sətir çapa verməlidir:

Hərəkət	Mənası
$r \ P$	Oxu P məkanını. Bu sətiri çapa verdikdən sonra proqramın P -də yazılmış cari dəyəri ehtiva edən sətiri oxumalıdır.
$w \ P \ V$	Yaz P məkanına yeni V dəyərini. Əgər eyni gündə eyni P -yə birdən çox nüsxə yazarsa, <i>Not correct</i> (Doğru deyil) qərarını alacaqsan. Nümunələr və 3-cü alt tapşırıq istisna olmaqla, sən $0 \leq V \leq 1$ yazmalısən; Qiymətləndirmə bölməsinə bax.
$! \ N$	Cavabla və dayan: N üzv olduğunu bildir və siyahıyaalmada iştirakını dayandır. Cavab verdikdən sonra, proqramın normal şəkildə sonlanmalıdır . (Nəzərə al ki, proqramının digər nüsxələri cavab verib sonlanmadan əvvəl daha çox gün ərzində işləməyə davam edə bilər.)

Əgər proqramının hər hansı bir nüsxəsi N üçün səhv dəyər cavab verərsə, protokolu pozarsa, 500 gündən çox vaxt sərf edərsə və ya (hər proses üçün) vaxt/yaddaş limitini aşarsa, həllin verilmiş test üçün *Not correct* kimi qiymətləndiriləcək.

Əks halda, proqramın testində (*Qismən*) *Doğru* (Partially Correct) hesab ediləcək və D dəyərində əsasən qiymətləndiriləcək: hər hansı bir nüsxənin cavab vermək üçün sərf etdiyi maksimum gün sayı. Tam bal almaq üçün hər testi $D \leq 61$ və $V \leq 1$ ilə həll etməlisən. Daha çox detal üçün Qiymətləndirmə bölməsinə bax.

Flushing (Buferi təmizləmə). Əgər verilmiş şablonlardan istifadə etmirsənsə, hər sətirdən sonra standart çıxışı flush etdiyinə (buferi təmizlədiyinə) əmin ol, əks halda proqramın *Not correct* kimi qiymətləndirilə bilər. Python-da, əgər sətirləri oxumaq üçün `input()` istifadə edirsənsə, bu avtomatik baş verir. C++-da `cout << endl`; yeni sətir çap etməklə yanaşı həm də flush edir; əgər `printf` istifadə edirsənsə, `fflush(stdout)` yaz.

Məhdudiyyətlər

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Ən çox 500 gün istifadə edə bilərsiniz.

Qiymətləndirmə

Proqramın alt tapşırıqlara qruplaşdırılmış bir neçə testlə yoxlanılacaq. Alt tapşırıqdan bal almaq üçün, onun içindəki bütün testləri doğru həll etməlisən.

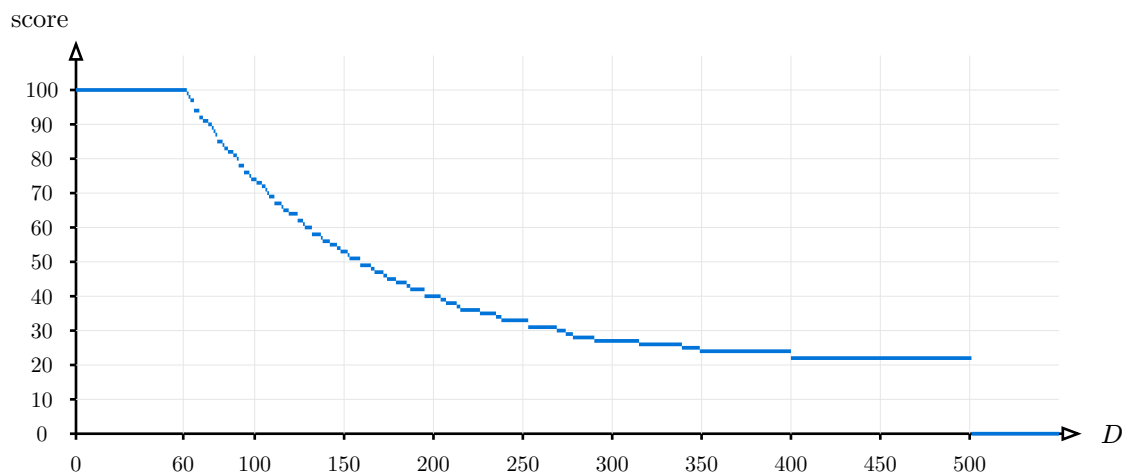
- Alt-tapşırıq 0 [0 xal]:** Nümunələr (istənilən $0 \leq V \leq 1\,000\,000\,000$ tam ədədini yazı bilərsiniz).
- Alt-tapşırıq 1 [11 xal]:** $M \leq 100$ və N üzvün ID-ləri $0, 1, \dots, N - 1$ -dir.
- Alt-tapşırıq 2 [12 xal]:** $1 \leq N \leq 2$.
- Alt-tapşırıq 3 [22 xal]:** $M \leq 8000$ və istənilən $0 \leq V \leq 1\,000\,000\,000$ tam ədədini yazı bilərsiniz.
- Alt-tapşırıq 4 [55 xal]:** Əlavə məhdudiyyət yoxdur.

1, 2 və 4-cü alt tapşırıqlarda, sən hər **Yazma** hərəkətində yalnız $V = 0$ və ya $V = 1$ yazı bilərsiniz.

Tutaq ki, X_s , s alt tapşırığı üçün verilən maksimum baldır (yuxarıda göstərilib), və D_s , s alt tapşırığındakı hər hansı testdə proqramlarının sərf etdiyi ən böyük gün sayıdır. Onda:

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

score_s dəyəri hər alt tapşırıq üzrə ən yaxın tam ədədə yuvarlaqlaşdırılır və sənin yekun balın bunların cəminə bərabərdir. Məsələdən tam bal almaq üçün hər bir testdə $D \leq 61$ və $V \leq 1$ olmalıdır.



Şəkil 1: Yekun bal, hər alt tapşırığın eyni maksimum D ilə həll edildiyi fərz edildikdə.

Nümunələr

Birinci nümunə. Hər sütun cütüyü qiymətləndiriciylə bir nüsxə arasındakı əlaqəni göstərir.

Qiy.	Nüs. 0	Qiy.	Nüs. 1	Qiy.	Nüs. 2	Qiy.	Nüs. 3	Qiy.	Nüs. 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					

İkinci nümunə.

Qiymətləndirici	Nüsxə 0
0 8000	
	w 0 0
	w 1 1
	r 2
1	
	! 2

Qiymətləndirici	Nüsxə 1
3 8000	
	w 2 1
	r 1
0	
	r 2
1	
	r 1
1	
	! 2

İzah

Birinci Nümunə. Bizdə ardıcıl 0, 1, 2, 3, 4 ID-lərinə sahib $N = 5$ üzv və $M = 100$ var (1, 3 və 4-cü alt tapşırıqlar üçün keçərlidir). i nüsxəsi i ID-sinə malik üzvə uyğun gəlir. Yuxarıdakı qarşılıqlı əlaqə sadəcə icazə verilən əməliyyat ardıcılıqlarından biridir və səmərəli və ya məntiqli bir strategiya olması **nəzərdə tutulmayıb**; o yalnız protokolun necə işlədiyini nümayiş etdirmək üçün verilib.

İkinci Nümunə. ID-ləri 0 və 3 olan $N = 2$ üzv, və $M = 8000$ var (2, 3 və 4-cü alt tapşırıqlar üçün keçərlidir). Birinci gün, 0 ID-li üzv 0-cı məkana 0 yazır (dəyişiklik yoxdur), 3 ID-li üzv isə 2-ci məkana 1 yazır.

location	0	1	2	3	4	...
value	0	0	1	0	0	...

İkinci gün, ID 0, 1-ci məkana 1 yazır, ID 3 isə həmin məkanı oxuyur. Nəzərə al ki, oxuma prosesi gündüz, axşamkı yazma prosesindən əvvəl baş verir. Beləliklə, ID 3 hələ də 0 görür.

location	0	1	2	3	4	...
value	0	1	1	0	0	...

Üçüncü gün onların hər ikisi 1-in yazıldığı 2-ci məkanı oxuyurlar.

Dördüncü gün ID 0, 2 üzv olduğunu cavab verir (doğru cavabdır), ID 3 isə 1-ci məkandakı 1-i oxuyur. ID 0 bundan sonra dərhal çıxır və gələcək günlərdə iştirak etmir.

Nəhayət, $D = 5$ günündə qalan üzv də düzgün olaraq $N = 2$ cavabını verir.

Test etmək

Həllinin test edilməsini asanlaşdırmaq üçün biz CMS-dən yükləyə biləcəyiniz sadə bir alət təqdim edirik. Bu alətdən istifadə etmək məcburi deyil. Nəzərə al ki, CMS-dəki rəsmi qiymətləndirici test alətindən fərqlidir.

Alətdən istifadə etmək üçün sənə input faylı lazımdır. Təqdim olunmuş `census.input0.txt` və `census.input1.txt` giriş nümunələrindən istifadə edə, və ya öz faylını yarada bilərsiniz. Input faylı üzvlərin N sayı və mümkün olan ID-lərin M sayı ilə başlamalıdır, ardınca cəmiyyət üzvlərinin ID-lərini təyin edən N ədəddən ibarət bir sətir gəlməlidir.

Python proqramları üçün, məsələn `census.py` (normalda `pypy3 census.py` olaraq işlədilir) test alətini belə işlət:

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

C++ proqramları üçün əvvəlcə həllini kompilyasiya et:

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

və sonra test alətini işlət:

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

Nəzərə al ki, bu məsələdə qiymətləndirici ilə əlaqə saxlamaq üçün standard output istifadə olunur, buna görə də ondan debug məqsədilə istifadə edilməməlidir. Əvəzində sən standard error output-dan (stderr) istifadə edə bilərsiniz. C++-da sən `cerr << msg << endl;` istifadə edə bilərsiniz. Python-da isə `print(msg, file=sys.stderr)` istifadə edə bilərsiniz.

Test aləti bu stderr mesajlarını sənin bütün proqram nüsxələrinin icra etdiyi sorğularla birlikdə oxuyacaq və göstərəcək. Nəzərə al ki, texniki səbəblərə görə onlar bir-birinə nəzərən cüzi sinxronizasiyadan çıxmış görünə bilər.