

Alisa, Bob va Sxema

Cyberland Circuit fondi(Circuit=sxema) n nafar a'zodan iborat. Bir bir a'zo o'zining takrorlanmas ismiga va sevimli soniga ega(sevimli soni takrorlanishi mumkin, lekin ismlar takrorlanmas)

Barcha fond a'zolari o'zaro m ta xat alishishgan. Har bir xat o'z yuboruvchisi va qabul qiluvchisiga ega, hamda xatning ichida yuboruvchining sevimli soni yozilgan.

Barcha fond a'zosi, o'ziga kelgan xatlarning ichidagi sonlarning yig'indisini hisoblaydi va bu qiymatni 65536 (ya'ni, 2^{16}) bo'lgan qoldig'ini topib uni o'zining natijaviy soni deb qabul qiladi.

Sizning vazifangiz barcha natijaviy sonlarni aniqlash.

Biroq, ahvol siz o'ylaganchalik sodda emas. Alisa, Bob va Sxema bu masalani qiyinroq usulda yechishni hal qilishdi:

- Alisa n nafar fond a'zosining barchasi haqida ma'lumotlarni (ismi va sevimli soni) biladi, ammo xatlar haqida hech nimani bilmaydi. U Sxemaga uzunligi 10^5 dan oshmagan binar satr yuborishi lozim.
- Bob m xatni biladi (yuboruvchi va qabul qiluvchilarining ismlarini), ammo fond a'zolari haqida ma'lumotga ega emas. U ham Sxemaga uzunligi 10^5 dan oshmagan binar satr yuborishi lozim.
- Sxema o'z navbatida Alisa va Bob yuborgan satrlarni qabul qilib bu satrlardan foydalangan holda uzunligi $16n$ bo'lgan binar satrni natija sifatida hosil qilishi lozim. Biroq, hisoblash kuchi kam bo'lganligi sababli, Sxema faqatgina sodda mantiqiy amallarni bajara oladi (misol uchun., AND, OR, NOT).

Quyida Sxemaning ishlashi to'liqroq bayon etilgan.

Sxemaning ishlashi

Yo'lak - sxemaning asosiy elementi hisoblanadi. Yo'lakka 2 ta yoki 0 ta mantiqiy qiymat kiradi(sxema turiga ko'ra) va yo'lakdan 1 ta mantiqiy qiymat(natija) chiqadi. Ikki xil turdagi yo'lak mavjud: kirituvchi va hisoblovchi.

- Kirituvchi yo'lakka hech qanday qiymat kirmaydi, unda faqat chiqish mavjud. Bu yo'laklar yordamida Alisa va Bob bergan satrlardagi bitlar ifodalanadi.
 - Jami $l_A + l_B$ ta kirituvchi yo'laklar bo'ladi, hamda ular 0 dan $(l_A + l_B - 1)$ gacha raqam, bu yerda l_A, l_B mos ravishda Alisa va Bob yuborgan satrlar uzunliklari.
 - $0 \leq i < l_A$, uchun i -yo'lak chiqishida Alisa bergan satrning i -biti beriladi.
- Hisoblovchi yo'lak 2 ta kirish va 1 ta chiqishga ega, bu turdagi yo'lak hisoblash jarayoni uchun javob beradi.
 - Hisoblovchi yo'laklar uchun raqamlash $(l_A + l_B)$ dan boshlanadi.
 - Har bir yo'lak uchun siz kirish sifatida 2 ta boshqa yo'lak raqamlarini(indekslerini) va amal turi $p(0 \leq p \leq 15)$ ni kiritishingiz lozim.
 - Sikllar hosil bo'lishini oldini olish maqsadida, kirishga belgilagan yo'lak raqamlari shu yo'lak raqamidan kichikroq bo'lishi shart.
 - Kirishga belgilagan ikki yo'laklarning natijasi(chiqish) mos ravishda x_0 va x_1 bo'ladigan bo'lsa ($x_0, x_1 \in \{0, 1\}$), unda bu hisoblovchi yo'lak natijasi quyidagi funksiya orqali hisoblanadi:

$$f(p, x_0, x_1) = \left\lfloor \frac{p}{2^{x_0+2x_1}} \right\rfloor \bmod 2$$

Quyida sizga foydali bo'lishi mumkin bo'lgan namunalar berilgan:

x_0	x_1	x_0 AND x_1 $f(8, x_0, x_1)$	x_0 OR x_1 $f(14, x_0, x_1)$	x_0 XOR x_1 $f(6, x_0, x_1)$	NOT x_0 $f(5, x_0, x_1)$
0	0	0	0	0	1
1	0	0	1	1	0
0	1	0	1	1	1
1	1	1	1	0	0

Kod yozish detallari

Iltimos, e'tibor bering:

- Barcha massivlarning indekslari 0 dan boshlanadi. Misol uchun, agar `a` bu uzunligi `n` bo'lgan massiv deyilsa, demak `a[0]` to `a[n-1]` gacha elementlari `a` massiv elementlari hisoblanadi. Oraliqdan tashqaridagi indekslarga murojaat etish chegaradan-chiqish xatolarini hosil qilishi mumkin.
- Barcha satrlar null belgisi - `\0` bilan tugaydi.

Quyidagi funksiyalarni e'lon qilishingiz kerak bo'ladi:

Alisa

```
int alice(const int n, const char names[][5], const unsigned short numbers[], bool outputs_alice[]);
```

Yo'nalish	Nomi	Uzunligi	Ma'nosi	Cheklovlari
Kirish	<code>n</code>	1	n	$0 \leq n \leq 700$
	<code>names</code>	n	Fond a'zolari ismlari.	Barcha ismlar unikal(takrorlanmas) va ko'pi bilan 4 ta kichik Ingliz alifbosi harflaridan tashkil topgan.
	<code>numbers</code>	n	Mos fond a'zosining sevimli soni.	Barcha sonlar 0 dan 65535 gacha oraliqdan olingan butun sonlar.
Output	<code>outputs_alice</code>	l_A	Sxemaga yuboriladigan binar satr.	
	Qaytarish(return) qiymati	1	l_A	l_A uzunligi 10^5 dan oshmasligiga qarashingiz kerak, bundan tashqari bir xil n lar uchun l_A qiymati bir xil bo'lishi lozim.

Bob

```
int bob(const int m, const char senders[][5], const char recipients[][5], bool outputs_bob[]);
```

Yo'nalish	Nomi	Uzunligi	Ma'nosi	Cheklovlari
Kirish	<code>m</code>	1	m	$0 \leq m \leq 1000$
	<code>senders</code>	m	Har bir xatning yuboruvchisi ismi.	Barcha ismlar Alisaga berilgan (kirish sifatida).
	<code>recipients</code>	m	Har bir xatni qabul qiluvchisining ismi..	
Chiqish	<code>outputs_bob</code>	l_B	Sxemaga yuboriladigan binar satr.	
	Qaytarish(return) qiymati	1	l_B	l_B uzunligi 10^5 dan oshmasligiga qarashingiz kerak, bundan tashqari bir xil m lar uchun l_B qiymati bir xil bo'lishi lozim.

Sxema

Sxemaning hisoblash jarayoni, haqiqiy sxemaniki bilan bir xilligiga ishonch hosil qilish uchun, sxema Alisa va Bobdan qaytgan satrlarni qabul qilmaydi. Sizga faqatgina Alisa va Bob qaytargan satr uzunliklari beriladi, siz sxema tuzilishini chiqarishingiz kerak.

```
int circuit(const int la, const int lb, int operations[], int operands[][2], int outputs_circuit[][16]);
```

Yo'nalish	Nomi	Uzunligi	Ma'nosi	Cheklovlari
-----------	------	----------	---------	-------------

Yo'nalish	Nomi	Uzunligi	Ma'nosi	Cheklovlari
Kirish	1a	1	l_A	
	1b	1	l_B	
Chiqish	operations	l	Har bir hisoblovchi yo'lak bajaradigan amal turi.	0 dan 15 gacha oraliqdan butun son.
	operands	l	Kirish sifatida berilgan 2 ta boshqa yo'lak raqamlari - operandlar.	Bu 2 raqamning har biri ushbu yo'lak raqamidan kichik bo'lishi shart.
	outputs_circuit	n	Sxema ishlashidan so'ng natija sifatida qabul qilinadigan yo'laklar.	<code>outputs_circuit[i][j]</code> i-a'zoning natijaviy soni eng kichik bitdan boshlab sanaganda j -biti. Fond a'zolari Alisaga berilgan ro'yxat bo'yicha tartiblangan.
	Qaytarish(return) qiymati	1	l , kirish yo'laklarini inobatga olganda, jami ishlatilgan yo'laklar soni.	$l \leq 2 \times 10^7$ ekanligiga qarashingiz kerak

Siz raqami(indeksi) $l_A + l_B$ dan kichik bo'lgan yo'lak ma'lumotlarini(`operations` va `operands`) ham o'zgartirishingiz, ammo greyder buni inobatga olmaydi.

Namunalar

Quyidagicha funksiya chaqiruvini ko'rib chiqaylik:

```
alice(3, {"alic", "bob", "circ"}, {10000, 20000, 30000}, outputs_alice);
bob(5, {"alic", "bob", "bob", "circ", "circ"}, {"circ", "circ", "alic", "circ", "circ"}, outputs_bob);
```

Bu quyidagi holatni tasvirlaydi:

- Alisa fondda 3 ta a'zo borligini biladi, `alic` ismli a'zoning seveli soni 10000 va h.k. `alice()` funksiyasi quyidagicha ishlashi mumkin (bu namuna, to'g'ri strategiyani kafolatlamaydi),
 - `alice()` ning qaytarish qiymati 2, bu $l_A = 2$ ni ifodalaydi.
 - `alice()` funksiyasining ichida `outputs_alice[0] = 1`, `outputs_alice[1] = 0`, deb o'rnatiladi, ya'ni natijaviy satr 10 ko'rinishida.
- Bob jami 5 xat borligini biladi, birinchi xat `alic` dan `circ` ga va h.k. `bob()` funksiyasi quyidagicha ishlashi mumkin (bu namuna, to'g'ri strategiyani kafolatlamaydi),
 - `bob()` funksiyasining qaytarish qiymati 3, bu $l_B = 3$ ni ifodalaydi.
 - `bob()` funksiyasining ichida `outputs_bob[0] = 1`, `outputs_bob[1] = 1`, `outputs_bob[2] = 0` deb o'rnatiladi, ya'ni natijaviy satr 110 ko'rinishida.

Oldin chaqirilgan `alice()` va `bob()` funksiyalarining natijalariga tayangan holda, quyidagicha chaqiruv qilsak bo'ladi:

```
circuit(2, 3, operations, operands, outputs_circuit);
```

To'g'ri natijaga olib keluvchi holatlardan biri:

- `circuit()` funksiyasi qaytaradigan natija 7, ya'ni biz yana qo'shimcha 2 ta yo'lak qo'shamiz, 5 va 6 yo'laklarini.
- `circuit()` funksiyasini ichida, `operations` va `operands`ni hamda `outputs_circuit`ni quyidagicha o'rnatish:
 - `operations = {-1, -1, -1, -1, -1, 8, 14}`, -1 bu yerda kirituvchi yo'laklarni anglatadi. Sababi kirituvchi yo'laklar hisoblay olmaydi;
 - `operands = {{-1, -1}, {-1, -1}, {-1, -1}, {-1, -1}, {-1, -1}, {0, 4}, {2, 5}}`;
 - `outputs_circuit = {{5, 5, 5, 5, 5, 6, 5, 5, 6, 6, 6, 5, 5, 6, 5}, ...}`. Massiv uzunroq, shunga uni ilova qilingan `abc.cpp` faylidan ochib ko'rishingiz mumkin.

Tepadagi funksiyalarga tayangan holda, natija quyidagicha hisoblanadi:

- Hisoblash amali 8 bo'lgan hisoblovchi yo'lak qo'shing, unga kirish 0 va 4 yo'lak chiqishlari bo'lsin. 0-yo'lak Alisa qaytargan satrning 0-biti bilan bir xil, demak 0-yo'lak natijasi 1. Chunki Alisa 10 qaytargan edi; 4-yo'lak Bob qaytargan satrning 2-bitini anglatadi, bu

esa 0, chunki Bob 110 satrini qaytargan edi. Shunday qilib bu yangi 5-yo'lak natijasi $f(8, 0, 1) = 0 \text{ AND } 1 = 0$.

- Hisoblash amali 14 bo'lgan yangi yo'lak qo'shing, unga kirish 2 va 5 yo'lak chiqishlari bo'lsin. 2 yo'lak Bob qaytargan satrning 0-biri bilan bir xil, demak 2-yo'lak natijasi 1; 5-yo'lak natijasi 0. Shunday qilib bu yangi 6-yo'lak natijasi $f(14, 1, 0) = 1 \text{ OR } 0 = 1$.
- `output_circuit[0]` `alic` ning natijaviy sonini anglatadi, ya'ni $(0100111000100000)_2 = 20000$. `alic` faqatgina `bob` dan xat olgan ekan, undagi natija haqiqatdan ham 20000.
- `bob` ning natijaviy soni 0, chunki u hech kimdan xat olmagan; `circ` ning natijaviy soni $(10000 + 20000 + 30000 + 30000) \bmod 65536 = 24464$ bo'ladi.

`abc.cpp` dagi yechim mana shu testdan o'tadi, boshqa testlardan o'tishini kafatlamaymiz.

Chegaralar

Barcha testlar uchun:

- $0 \leq n \leq 700, 0 \leq m \leq 1000$.
- Barcha ismlar unikal, hamda faqatgina kichik Ingliz alifbo harflaridan iborat. Barcha ismlarning uzunligi 4 dan katta emas.
- Har bir fond a'zosining sevimli soni 0 dan 65535 gacha oraliqdan olingan.
- Xatlarning yuboruvchilari ismlari Alisaning `names` massivida uchragan.
- `alice()` va `bob()` funksiyalari chaqiruvlari 2048 MiB xotira chegarasiga va 0.02 sekund vaqt chegaralariga ega.
- `circuit()` funksiyasi esa 2048 MiB xotira chegarasiga va 7 sekund vaqt chegarasiga ega.

To'liq baholash uchun, `alice()` va `bob()` funksiyalari bitta test ichida bir necha marta chaqirilishi mumkin. Har bir chaqiruv 0.02 sekund vaqt chegarasiga ega bo'ladi.

Qism masalalar

Qism masalalar, tip A (12 ball)

1,2,3 qism masalalar A tipdagi qism masalalardir. Bu yerda $n = 1$.

Har bir qism masala quyidagicha qo'shimcha cheklavlarga ega:

- Subtask 1 (4 ball): $m = 0$.
- Subtask 2 (4 ball): $0 \leq m \leq 1$.
- Subtask 3 (4 ball): $0 \leq m \leq 1000$.

Qism masalalar, tip B (54 ball)

4,5,6 qism masalalar B tipdagi qism masalalardir. Bu yerda:

- $0 \leq n \leq 30, \frac{n}{2} \leq m \leq n^2$.
- Bir vaqtning o'zida bir xil yuboruvchisi va qabul qiluvchisi bo'lgan ikkita xat, ya'ni ikkita bir xil xat mavjud emas.
- Barcha fond a'zolarining ismlari Bobning kirishiga beriladi(ya'ni, har bir fond a'zosi kamida bir marta xat yuboradi yoki kamida bir marta xat qabul qiladi).

Har bir qism masala quyidagicha qo'shimcha cheklavlarga ega:

- Subtask 4 (24 ball): $n = 26$, barcha fond a'zolarining ismlari uzunligi 1 ga teng, hamda Alisaning kirishiga ular aniq a dan z gacha tartiblangan holda kiritiladi.
- Subtask 5 (24 points): $n = 26$.
- Subtask 6 (6 ball): Maxsus cheklavlarsiz.

Qism masalalar, tip C (34 ball)

7,8,9 qism masalalar C tipdagi qism masalalardir. Bu yerda: $0 \leq n \leq 700, 0 \leq m \leq 1000$.

Har bir qism masala quyidagicha qo'shimcha cheklavlarga ega:

- Subtask 7 (18 points): $n = 676$, barcha fond a'zolarining ismlari uzunligi 2 ga teng, hamda Alisaning kirishiga ular aniq leksigrafik saralangan holda kiritiladi (ya'ni, $aa, ab, ac, \dots, az, ba, \dots, bz, ca, \dots, zz$).
- Subtask 8 (10 points): $n = 676$.
- Subtask 9 (6 points): Maxsus cheklavlarsiz.

Namunaviy greyder

Namunaviy greyder ma'lumotlarni quyidagi formatda o'qiydi:

- qator 1: $n \ m$
- qator $2 + i (0 \leq i \leq n - 1)$: $names_i \ numbers_i$
- qator $2 + n + i (0 \leq i \leq m - 1)$: $senders_i \ recipients_i$.

Namunaviy greyder natijani quyidagi formatda chiqaradi:

- Agar dastur muvaffaqiyatli yakunlansa, namunaviy greyder har birida bittadan son bo'lgan n ta qator chiqaradi. Har bir son funksiyalaringiz hisoblagan navbatdagi fond a'zosining natijaviy sonini chiqaradi.
- Aks holda, greyder chiqish oqimiga hech narsa chiqarmaydi va xatolik haqida ma'lumotlarni `abc.log` ga papkadagi yozadi.
- Qo'shimcha ravishda, namunaviy greyder l_A, l_B, l ning qiymatlarini va har bir funksiyaning chqiruvi olgan vaqttni `abc.log` ga yozib qoldiradi.

Namunaviy greyder xotira chegarasini va bir xil n / m lar uchun l_A / l_B larning bir xilligini tekshirmaydi.