

Alice, Bob, and Dövrə

Kiberland Dövrə Fondu n üzvdən ibarətdir. Hər bir üzvün öz sevimli ədədi və unikal adı var (sevimli ədədlər fərqli olmaya bilər).

Üzvlər arasında m məktub göndərilib. Hər məktubun bir göndərənini və alanı var və məktubun məzmunu göndərəninin sevimli ədədidir.

Hər bir üzv qəbul etdiyi məzmunların (göndərənlərin sevimli ədədləri) cəmini hesablayır və 65536 (yəni, 2^{16})-ya qalıqını öz yekun nəticəsi olaraq qəbul edir.

Sizin tapşırığınız bütün yekun nəticələri müəyyən etməkdir.

Ancaq vəziyyət göründüyü qədər sadə deyil. Alice, Bob və Dövrə bu problemi bir az daha mürəkkəb yolla həll etməyə qərar verirlər:

- Alice n üzvün (ad və sevimli ədəd) hamısını bilir, lakin məktublar haqqında heç bir məlumat bilmir. O, Dövrəyə uzunluğu 10^5 -dən çox olmayan ikili sətir göndərməlidir.
- Bob m məktubun (göndərən və alanın adı) hamısını bilir, lakin üzvlər haqqında heç bir məlumat bilmir. O, Dövrəyə uzunluğu 10^5 -dən çox olmayan ikili sətir göndərməlidir.
- Dövrə Alice və Bob tərəfindən göndərilən ikili sətirləri qəbul edə və sonra çıxış kimi $16n$ bitdən ibarət ikili sətir yarada bilər. Bununla belə, məhdud hesablama gücünə görə, Dövrə yalnız əsas məntiqi əməliyyatları yerinə yetirməyə qadirdir (məsələn, AND, OR, NOT).

Aşağıda dövrənin necə işlədiyini ətraflı şəkildə təqdim edəcəyik.

Dövrə Detalları

Qeit bir dövrənin əsas elementidir. Qeit sıfır və ya iki məntiqi girişdən (qeitin növündən asılı olaraq) və bir məntiqi çıxışdan ibarətdir. İki növ qeit var: giriş qeytləri və hesablama qeytləri.

- Giriş qeytlərinin girişi yoxdur və Alice və Bob tərəfindən göndərilən ikili sətirlərin bitlərini təmsil edir.
 - 0-dan $(l_A + l_B - 1)$ -ə kimi nömrələnmiş $l_A + l_B$ sayda giriş qeytləri olacaq. Burada l_A, l_B müvafiq olaraq Alice və Bobdan gələn sətirlərin uzunluqlarıdır.
 - $0 \leq i < l_A$, üçün i -ci qeytin çıxışı Alicedən gələn sətirin i -ci bitidir;
 - $0 \leq i < l_B$, üçün $(i + l_A)$ -ci qeytin çıxışı Bobdan gələn sətirin i -ci bitidir.
- Hesablama qeytlərinin iki girişi var və hesablama prosesini təmsil edir.
 - Hesablama qeytləri üçün nömrələr $(l_A + l_B)$ -dən başlayır.
 - Hər bir hesablama qeyti üçün siz onun girişinin asılı olduğu iki qeytin nömrəsini və əməliyyat növünü $p(0 \leq p \leq 15)$ təqdim etməlisiniz.
 - Sirkulyar asılılıqların qarşısını almaq üçün iki asılı qeytin nömrəsi hesablama qeytinin nömrəsindən kiçik olmalıdır.
 - Əgər onun iki asılı qeytinin çıxışları müvafiq olaraq x_0 və x_1 ($x_0, x_1 \in \{0, 1\}$) olarsa, onda hesablama qeytinin çıxışı aşağıdakı kimi hesablanır:

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

Sizin üçün faydalı ola biləcək bəzi nümunələr:

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

İmplementasiya Detalları

Nəzərə alın ki:

- Bütün massiv indeksləri 0-dan başlayır. Məsələn, a uzunluğu n olan massivdirsə, $a[0]$ -dan $a[n-1]$ -ə qədər etibarlı məlumatdır, bu diapazondan kənarda olan indekslərə daxil olmaq hüduddan kənar xəyata (out-of-bounds error) səbəb ola bilər.
- Bütün sətirlər null simvolu olan `\0` ilə dayandırılır.

Aşağıdakı prosedurları icra etməlisiniz:

Alice

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

İstiqamət	Dəyər	Uzunluq	Məna (izah)	Məhdudiyyət
Giriş	n	1	n	$0 \leq n \leq 700$
	names	n	Üzvlərin adları.	Bütün adlar fərqlidir, yalnız kiçik ingilis hərflərindən ibarətdir və uzunluğu maksimum 4 ola bilər.
	numbers	n	Üzvlərin sevimli ədədləri.	Hər bir ədəd 0-dan 65535-ə qədərdir.
Çıxış	outputs_alice	l_A	İkili sətir dövrəyə göndərilir.	
	(Funksiyadan qaytarılan dəyər)	1	l_A	$l_A \cdot 10^5$ -i keçməməlidir və eyni n -lər üçün l_A -lar sabit olmalıdır.

Bob

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

İstiqamət	Dəyər	Uzunluq	Məna (izah)	Məhdudiyyət
Giriş	m	1	m	$0 \leq m \leq 1000$
	senders	m	Hər bir məktubu göndərəninin adı.	Bütün adlar Alice-in girişində görünür
	recipients	m	Hər bir məktubu alanın adı.	
Çıxış	outputs_bob	l_B	İkili sətir dövrəyə göndərilir	
	(Funksiyadan qaytarılan dəyər)	1	l_B	$l_B \cdot 10^5$ -i keçməməlidir və eyni m -lər üçün l_B -lər sabit olmalıdır.

Dövrə

Dövrənin hesablama prosesinin ümumi dövrəyə bənzədiyinə əmin olmaq üçün siz Alice və Bob tərəfindən göndərilən ikili sətirləri birbaşa əldə edə bilməzsiniz. Siz yalnız bu iki sətirin uzunluğunu bilirsiniz və dövrə strukturunu çıxarırsınız.

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

İstiqamət	Dəyər	Uzunluq	Məna (izah)	Məhdudiyyət
Giriş	la	1	l_A	
	lb	1	l_B	
Çıxış	operations	l	Dövrədəki hər bir qeit tərəfindən yerinə yetirilən əməliyyat növü.	0-dan 15-ə qədər tam ədəd.
	operands	l	Hər bir qeita daxil olan giriş qeitləri	Nömrə cari qeitin nömrəsindən az olmalıdır.

İstiqamət	Dəyər	Uzunluq	Məna (izah)	Məhdudiyyət
	outputs_circuit	n	Dövrə çıxışının qeit nömrəsi.	outputs_circuit[i][j] i -ci üzv üçün yekun nəticənin j -ci bitini (ən az əhəmiyyətli bitdən saymaqla) bildirən qeitin nömrəsini bildirir. Üzvlər Alice-in giriş verilənlərinə uyğun olaraq sıralanır.
	(Funksiyadan qaytarılan dəyər)	1	l . Bu l qeytlərin ümumi sayını (giriş qeytləri daxil olmaqla) bildirir.	$l \leq 2 \times 10^7$ olduğundan əmin olmalısınız

operations və operands massivlərində indeksləri $l_A + l_B$ -dən kiçik olan qeytlərin məlumatlarını dəyişdirə bilsəniz də, qreyder belə dəyişiklikləri nəzərə almayacaq (görməzdən gələcək).

Nümunə

Aşağıdakı prosedurlara nəzər yetirək:

```
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 aşağıdakı ssenarini təmsil edir:

- Alice bilir ki, 3 üzv var, alic adlı üzvün sevimli nömrəsi 10000-dir və s. alice() üçün mümkün bir nəticə belə ola bilər,
 - alice()-in qaytardığı dəyər 2-dir, bu $l_A = 2$ deməkdir.
 - alice() funksiyasının daxilində, nəticənin ikili sətrinin 10 olduğunu göstərən outputs_alice[0] = 1, outputs_alice[1] = 0 təyin edin.
- Bob bilir ki, 5 məktub var, birinci məktub alic-dən circ-ədir və s. bob() üçün mümkün bir nəticə belə ola bilər,
 - bob()-un qaytardığı dəyər 3-dür, bu $l_B = 3$ deməkdir.
 - bob() funksiyasının daxilində, nəticənin ikili sətrinin 110 olduğunu göstərən outputs_bob[0] = 1, outputs_bob[1] = 1, outputs_bob[2] = 0 təyin edin.

alice() və bob() üçün yuxarıdakı çıxışlara əsasən aşağıdakı çağırış olacaq:

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

Bu funksiya üçün düzgün çıxış aşağıdakı kimi olacaq:

- circuit()-in qaytardığı dəyər 7-dir. Bu o deməkdir ki, biz 5 və 6 ilə nömrələnmiş iki hesablama qeiti əlavə edirik.
- circuit() daxilində operations, operands və outputs_circuit-i aşağıdakı kimi təyin edin:
 - operations = {-1, -1, -1, -1, -1, 8, 14}, burada biz giriş qeytlərindən nəzərə alınmayan məlumatları bildirmək üçün -1 istifadə edirik;
 - 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, 5, 6, 6, 6, 5, 5, 6, 5}, ...}. Massiv bir az uzundur, tam massiv üçün əlavələrdə verilmiş abc.cpp faylına nəzər yetirə bilərsiniz.

Çıxış əsasən hesablama proseduru belədir ki,

- Girişləri 0 və 4 nömrəli qeytlərdən, əməliyyat növü 8 olan hesablama qeiti əlavə edin. 0 nömrəli qeitin çıxışı Alice-dən gələn sətrin 0-cı bitidir və onun dəyəri 1-dir; 4 nömrəli qeitin çıxışı Bob-dan gələn sətrin 2-ci bitidir və onun dəyəri 0-dir. Beləliklə, 5 nömrəli qeitin çıxışı $f(8, 0, 1) = 0 \text{ AND } 1 = 0$ -dir.
- Girişləri 2 və 5 nömrəli qeytlərdən, əməliyyat növü 14 olan hesablama qeiti əlavə edin. 2 nömrəli qeitin çıxışı Bob-dan gələn sətrin 0-cı bitidir və onun dəyəri 1-dir; 5 nömrəli qeitin çıxışı 0-dir. Beləliklə, 6 nömrəli qeitin çıxışı $f(14, 1, 0) = 1 \text{ OR } 0 = 1$ -dir.
- output_circuit[0] alic üçün yekun nəticəni bildirir və bu $(0100111000100000)_2 = 20000$ -dir. alic yalnız bob-dan bir məktub aldığından alic-in yekun nəticəsi 20000-dir.
- bob-un yekun nəticəsi 0 olmalıdır, çünki o, heç bir məktub almır; circ-in yekun nəticəsi $(10000 + 20000 + 30000 + 30000) \bmod 65536 = 24464$ olmalıdır.

Əlavələrdəki abc.cpp bu nümunəni keçə bilər, lakin biz onun digər testləri keçə biləcəyinə zəmanət vermirik.

Məhdudiyyətlər

Bütün testlər üçün:

- $0 \leq n \leq 700, 0 \leq m \leq 1000$.
- Bütün adlar fərqlidir, yalnız kiçik ingilis hərflərindən ibarətdir və maksimum 4 simvol uzunluğuna malikdir.
- Hər bir üzvün sevimli nömrəsi 0-dan 65535-ə qədərdir.
- Bütün məktub göndərənlərin və alanların adları Alice-in `names` massivində görünür.
- `alice()` və `bob()` 2048 MiB yaddaş limitinə və 0.02 saniyə zaman limitinə malikdir.
- `circuit()` 2048 MiB yaddaş limitinə və 7 saniyə zaman limitinə malikdir.

Yekun qiymətləndirmə üçün, `alice()` və `bob()` bir test daxilində bir neçə dəfə çağırılı bilər. 0.02 saniyə zaman limiti hər bir çağırış üçündür.

Alt tapşırıqlar

Alt tapşırıq növü A (12 bal)

Alt tapşırıq 1,2,3 A alt tapşırığına aiddir, burada $n = 1$.

Hər bir alt tapşırığın aşağıdakı əlavə məhdudiyyətləri var:

- Alt tapşırıq 1 (4 bal): $m = 0$.
- Alt tapşırıq 2 (4 bal): $0 \leq m \leq 1$.
- Alt tapşırıq 3 (4 bal): $0 \leq m \leq 1000$.

Alt tapşırıq növü B (54 bal)

Alt tapşırıq 4,5,6 B alt tapşırığına aiddir, burada:

- $0 \leq n \leq 30, \frac{n}{2} \leq m \leq n^2$.
- Eyni göndərəni və alanı olan iki məktub yoxdur.
- Bütün üzv adları Bobun giriş verilənlərində var (yəni, hər bir üzv ya ən azı bir məktub göndərir, ya da ən azı bir məktub alır).

Hər bir alt tapşırığın aşağıdakı əlavə məhdudiyyətləri var:

- Alt tapşırıq 4 (24 bal): $n = 26$, Bütün üzvlərin adları bir kiçik hərfdən ibarətdir və Alice-in girişində onlar a-dan z-yə doğru görünür.
- Alt tapşırıq 5 (24 bal): $n = 26$.
- Alt tapşırıq 6 (6 bal): Xüsusi məhdudiyyət yoxdur.

Alt tapşırıq növü C (34 bal)

Alt tapşırıq 7,8,9 C alt tapşırığına aiddir, burada $0 \leq n \leq 700, 0 \leq m \leq 1000$.

Hər bir alt tapşırığın aşağıdakı əlavə məhdudiyyətləri var:

- Alt tapşırıq 7 (18 bal): $n = 676$, bütün üzvlərin adları iki kiçik hərfdən ibarətdir və Alice-in girişində onlar leksikoqrafik ardıcılıqla görünür (məsələn, aa, ab, ac, ..., az, ba, ..., bz, ca, ..., zz).
- Alt tapşırıq 8 (10 bal): $n = 676$.
- Alt tapşırıq 9 (6 bal): Əlavə məhdudiyyət yoxdur.

Nümunə Qreyder

Nümunə qreyder girişi aşağıdakı formatda oxuyur:

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

Nümunə qreyder çıxışa aşağıdakı formatda çap edir:

- Proqram uğurla başa çatarsa, nümunə qreyder hər bir üzv üçün həyata keçirdiyiniz funksiyalarla hesablanmış yekun nəticəni əks etdirən, hər birində bir tam ədəd olan n sətir çıxaracaq.
- Əks halda, qreyder standart çıxışa heç nə çıxarmayacaq və xəta mesajlarını mövcud qovluqda `abc.log` faylına yazacaq.
- Bundan əlavə, nümunə qreyder l_A, l_B, l dəyərlərini və hər bir funksiyanın işləmə müddətini `abc.log` faylına yazacaq.

Nümunə qreyder yaddaş limitini və eyni n üçün l_A -lərin, eyni m üçün l_B -lərin bərabər olmalı olduğu məhdudiyyətini yoxlamayacaqdır.