



Magistral yol rüsumları

Yaponiyada şəhərlər bir-birilə magistral yollar şəbəkəsi ilə bağlıdır (əlaqəlidir). Bu şəbəkə N şəhər və M magistraldan ibarətdir. Hər magistral iki fərqli şəhəri birləşdirir. İki şəhər arasında yalnız bir magistral var. Şəhərlər 0-dan $N - 1$ -dək, yollar isə 0-dan $M - 1$ -dək nömrələnib. Siz istənilən yolla hər iki istiqamətə hərəkət edə bilərsiniz. Siz bu yollardan istifadə etməklə hər hansı şəhərdən istənilən başqa şəhərə səyahət edə bilərsiniz.

Hər magistraldan istifadə üçün ayrıca rüsum ödəmək lazımdır. Rüsumun məbləği yolda hərəkət edən **maşın sayından (trafikdən)** asılıdır. Trafik ya **seyrək**, ya da **sıx** ola bilər. Trafik seyrək olduqda rüsum A yen (Yapon valyutası), sıx olduqda isə B yendir. Hesab edin ki, A və B əvvəlcədən məlumdur və $A < B$.

Sizin bütün magistrallardakı verilmiş trafik vəziyyətinə əsasən, S və T ($S \neq T$) şəhərləri arasında ödəniləcək ən az cəmi rüsumu hesablayan sisteminiz vardır.

Bu sistem yalnız modeldir. S və T qiymətləri dəyişməzdir (yəni sistemə əvvəlcədən yazılıb) və sizə məlum deyil. Siz S və T -ni təyin etməlisiniz. Bunu etmək üçün siz sistemə müxtəlif trafik vəziyyəti təqdim etməklə hesablanmış rüsumla əsasən, S və T -ni tapmalısınız. Magistraldakı trafik vəziyyətini müəyyənləşdirmək bəhə başa gəldiyindən siz sistemdən çox istifadə etməməlisiniz.

Gerçəkləşdirmə detalları

Siz aşağıdakı proseduru yazmalısınız:

```
find_pair(int N, int[] U, int[] V, int A, int B)
```

- N : şəhərlərin sayıdır.
- U və V : M ölçülü massivdir, burada M şəhərləri birləşdirən magistralların ümumi sayıdır. Hər bir i üçün ($0 \leq i \leq M - 1$) i magistralı $U[i]$ və $V[i]$ şəhərlərini birləşdirir.
- A : Trafik seyrək olanda rüsumun qiymətidir.
- B : Trafik sıx olanda rüsumun qiymətidir.
- Bu funksiya hər test üçün bir dəfə çağrılacaqdır.
- Nəzərə alın ki, M -in qiyməti massivin ölçüsüdür və gerçəkləşdirmə qeydində göstəriləndiyi kimi əldə edilə bilər.

find_pair proseduru aşağıdakı funksiyanı çağıra bilər:

```
int64 ask(int[] w)
```

- w -nin ölçüsü M olmalıdır. w massivi yollarda olan vəziyyəti təsvir edir.
- $w[i]$ hər bir i üçün ($0 \leq i \leq M - 1$) i -ci magistralda olan trafiki göstərir. $w[i]$ -nin qiyməti 0 və ya 1 ola bilər.
 - $w[i] = 0$ i magistralında trafikin seyrək olduğunu göstərir.
 - $w[i] = 1$ i magistralında trafikin sıx olduğunu göstərir.
- Funksiya verilmiş w şərtləri əsasında S və T şəhərləri arasında ən kiçik cəmi rüsumu qaytarır.
- Funksiya cəmi 100 dəfə çağrıla bilər (hər bir test üçün).

find_pair cavabı bildirmək üçün aşağıdakı funksiyanı çağırmalıdır:

```
answer(int s, int t)
```

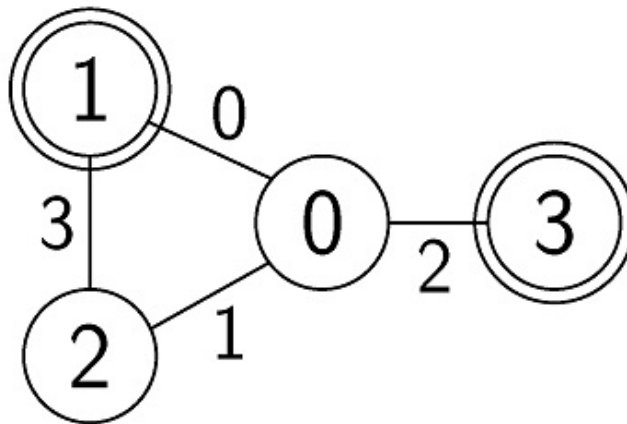
- s və t S və T cütliyüdür (ardıcılıq mənə kəsb etmir).
- Bu prosedur yalnız bir dəfə çağrılır.

Əgər yuxarıda göstərilən şərtlərdən hansısa biri ödənilməsə, sizin proqram **Wrong Answer** kimi dəyərləndiriləcəkdir. Əks halda, proqram **Accepted** kimi dəyərləndiriləcək və sizin bal ask-a olan çağırışların sayı kimi dəyərləndiriləcək (Altməsələlərə bax).

Nümunələr

Tutaq ki, $N = 4$, $M = 4$, $U = [0, 0, 0, 1]$, $V = [1, 2, 3, 2]$, $A = 1$, $B = 3$, $S = 1$, və $T = 3$ olsun.

Yoxlayıcı sistem find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3) çağırır.



Yuxarıdakı şəkildə, i nömrəli til i sayılı magistrala uyğundur. ask-ın çağırılmasına bəzi nümunələr və müvafiq cavablar aşağıda göstərilmişdir:

Çağırış	Cavab
ask([0, 0, 0, 0])	2
ask([0, 1, 1, 0])	4
ask([1, 0, 1, 0])	5
ask([1, 1, 1, 1])	6

ask([0, 0, 0, 0]) çağırışı üçün bütün magistrallarda trafik seyrəkdir və hər bir magistralın rüsumu 1-ə bərabərdir. $S = 1$ və $T = 3$ arasında ən ucuz yolun qiyməti $1 \rightarrow 0 \rightarrow 3$ -dür. Bu məsafə üçün cəmi rüsum 2-ə bərabərdir. Bunun üçün funksiya 2 qaytarır.

Düzgün cavab üçün find_pair proseduru answer(1, 3) və ya answer(3, 1) çağırılmalıdır.

Qoşma sıxılmış (zip) paketdəki sample-01-in.txt faylı bu nümunəyə uyğundur. Paketdə digər oxşar giriş verilənləri də vardır.

Məhdudiyyətlər

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Hər bir $0 \leq i \leq M - 1$ üçün
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ and $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)
- Hər hansı şəhərdən digər şəhərə yalnız magistralla hərəkət etmək olar.
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

Bu məsələdə yoxlayıcı sistem adaptiv deyil. Bu o deməkdir ki, S və T yoxlayıcı sistemin işə salınması zamanı təyin edilir və sizin proqramın işləməsindən asılı olaraq dəyişmir.

Altməsələlər

1. (5 xal) S və ya T -nin biri 0 olduqda, $N \leq 100$, $M = N - 1$
2. (7 xal) S və ya T -nin biri 0 olduqda, $M = N - 1$
3. (6 xal) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 xal) $M = N - 1$
5. (18 xal) $A = 1$, $B = 2$

6. (31 xal) Heç bir əlavə məhdudiyətlər yoxdur.

Hesab edin ki, sizin proqram **Accepted** kimi dəyərləndirilib və askfunksiyasını X dəfə çağırır. Sonra sizin P balınız altməsələnin nömrəsindən asılı olaraq, aşağıdakı kimi hesablanır:

- Altməsələ 1. $P = 5$.
- Altməsələ 2. Əgər $X \leq 60$, onda $P = 7$. Əks halda $P = 0$.
- Altməsələ 3. Əgər $\text{If } X \leq 60$, onda $P = 6$. Əks halda $P = 0$.
- Altməsələ 4. Əgər $\text{If } X \leq 60$, onda $P = 33$. Əks halda $P = 0$.
- Altməsələ 5. Əgər $\text{If } X \leq 52$, onda $P = 18$. Əks halda $P = 0$.
- Altməsələ 6.
 - Əgər $X \leq 50$, $P = 31$.
 - Əgər $51 \leq X \leq 52$, $P = 21$.
 - Əgər $53 \leq X$, $P = 0$.

Nəzərə alın ki, sizin hər bir altməsələ üçün balınız verilmiş altməsələyə aid testlərin qiymətlərinin minimumu kimi hesablanacaqdır.

Nümunə yoxlayıcı sistem

Nümunə yoxlayıcı sistem giriş verilənlərini aşağıdakı formatda oxuyur:

- Sətir 1: $N M A B S T$
- Sətir $2 + i$ ($0 \leq i \leq M - 1$): $U[i] V[i]$

Sizin proqram **Accepted** kimi qiymətləndirilibsə, nümunə yoxlayıcı sistem ask funksiyasına olan çağırışların sayı q -nü göstərməklə **Accepted: q** yazacaq.

Sizin proqram **Wrong Answer** kimi qiymətləndirilibsə, o **Wrong Answer: MSG** verəcək, belə ki, **MSG** aşağıdakılardan biri ola bilər:

- **answered not exactly once**: answer proseduru bir dəfədən çox çağırılıb.
- **w is invalid**: ask-a verilmiş w -nin ölçüsü M deyil və ya $w[i]$ hər hansı i ($0 \leq i \leq M - 1$) üçün nə 0 nə də 1 qiyməti almayıb.
- **more than 100 calls to ask**: ask funksiyası 100 dəfədən çox çağırılıb.
- **{s, t} is wrong**: answer proseduru yanlış s and t ilə çağırılıb.