



Платні дороги

В Японії міста пов'язані мережею автострад. Ця мережа складається з N міст і M автострад. Кожна автострада з'єднує два різних міста. Між містами може існувати лише одна автострада. Міста пронумеровані від 0 до $N - 1$, а автостради пронумеровані від 0 до $M - 1$. По кожній автостраді можна їхати в обох напрямках. З будь-якого міста можна добратись до будь-якого іншого використовуючи автостради.

Проїзд автострадами платний. Вартість проїзду залежить від **завантаженості** автостради. Вона буває **слабкою** або **сильною**. Коли завантаженість слабка, вартість складає A єн (Єни - японська валюта). Коли завантаженість сильна, вартість складає B єн. Гарантовано, що $A < B$. Зауважимо, що Ви знаєте величини A та B .

У Вас є пристрій, який за даними про завантаженість шляхів обчислює мінімальну вартість проїзду з міста S в місто T ($S \neq T$) за теперешніх умов завантаженості.

Однак, цей пристрій лише прототип. Значення S та T зафіксовані (тобто їх не можливо змінити на рівні пристрою) і вони вам невідомі. Ви хочете знайти значення S та T . Для цього ви плануєте подати у пристрій кілька станів завантаженості шляхів, і скористатися вартостями проїзду, що він виводить, щоб знайти S та T . Оскільки підготовка стану завантаженості обходиться не дешево, ви не бажаєте використовувати пристрій багато разів.

Деталі реалізації

Вам потрібно реалізувати наступну процедуру:

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

- N : кількість міст.
- U та V : масиви довжини M , де M - кількість автострад, що з'єднують міста. Для кожного i ($0 \leq i \leq M - 1$), автострада i з'єднує міста $U[i]$ та $V[i]$.
- A : вартість проїзду автострадою при слабкій завантаженості.
- B : вартість проїзду автострадою при сильній завантаженості.
- Ця процедура викликається рівно 1 раз для кожного тесту.
- Зауважимо, що величина M - це довжина масивів, її значення можна отримати способом, зазначеним в Зауваженнях до реалізації.

Процедура `find_pair` може викликати наступну функцію:

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

- Довжиною w має бути M . Масив w описує стан завантаженості автострад.
- Для будь-якого i ($0 \leq i \leq M - 1$), $w[i]$ задає стан завантаженості на автостраді i . Значення $w[i]$ має бути 0, або 1.
 - $w[i] = 0$ означає, що автострада i слабко завантажена.
 - $w[i] = 1$ означає, що автострада i сильно завантажена.
- Функція повертає найменшу сумарну вартість проїзду із міста S умісто T , за умови завантаженості, описаної масивом w .
- Цю функцію можна викликати не більше 100 разів (для кожного тесту).

`find_pair` повинна викликати наступну процедуру, щоб повідомити відповідь:

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

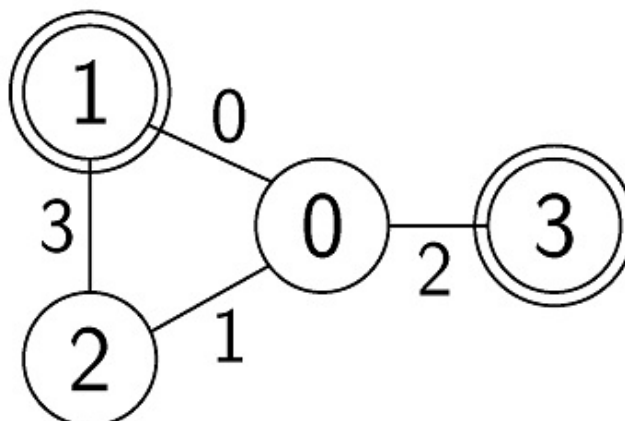
- s і t повинна бути парою міст S і T (порядок не принциповий).
- Цю процедуру потрібно викликати лише 1 раз.

Якщо якісь із описаних вимог не задовольняються, ваша програма отримає **Wrong Answer**. В іншому випадку результат буде **Accepted**, а оцінка обраховується за кількістю викликів функції `ask` (див. Підзадачі).

Приклад

Нехай $N = 4$, $M = 4$, $U = [0, 0, 0, 1]$, $V = [1, 2, 3, 2]$, $A = 1$, $B = 3$, $S = 1$, і $T = 3$.

Модуль перевірки викликає `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`.



На малюнку вище, ребро з номером i відповідає автостраді i . Кілька можливих викликів `ask` і відповідні відповіді наведені нижче:

Виклик	Результат
<code>ask([0, 0, 0, 0])</code>	2
<code>ask([0, 1, 1, 0])</code>	4
<code>ask([1, 0, 1, 0])</code>	5
<code>ask([1, 1, 1, 1])</code>	6

Для виклику `ask([0, 0, 0, 0])` завантаженість на всіх автострадах слабка і відповідна вартість складає 1. найдешевший маршрут із міста $S = 1$ до міста $T = 3$ наступний: $1 \rightarrow 0 \rightarrow 3$. Сумарна вартість проїзду ним складає 2. Тому функція повертає значення 2.

Для правильної відповіді, процедура `find_pair` повинна викликати `answer(1, 3)`, або `answer(3, 1)`.

Файл `sample-01-in.txt` у zip-архіві, що додається відповідає описаному прикладу. Там також доступні кілька інших прикладів.

Обмеження

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- Для кожного $0 \leq i \leq M - 1$
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ і $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)
- З будь-якого міста можна проїхати до будь-якого іншого, використовуючи автомагістралі.
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

В цій задачі модуль перевірки НЕ є адаптивним. Це означає, що S і T фіксуються при запуску модуля перевірки і не залежать від запитів які робить ваша програма.

Підзадачі

1. (5 балів) або S , або T дорівнює 0, $N \leq 100$, $M = N - 1$
2. (7 балів) або S , або T дорівнює 0, $M = N - 1$
3. (6 балів) $M = N - 1$, $U[i] = i$, $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 бала) $M = N - 1$

- 5. (18 балів) $A = 1, B = 2$
- 6. (31 бал) Без додаткових обмежень.

Припустимо, що ваша програма отримала результат **Accepted**, і зробила X запитів `ask`. Тоді оцінка P за тест, в залежності від номеру підзадачі, обраховується наступним чином:

- Підзадача 1. $P = 5$.
- Підзадача 2. Якщо $X \leq 60$, $P = 7$. Інакше $P = 0$.
- Підзадача 3. Якщо $X \leq 60$, $P = 6$. Інакше $P = 0$.
- Підзадача 4. Якщо $X \leq 60$, $P = 33$. Інакше $P = 0$.
- Підзадача 5. Якщо $X \leq 52$, $P = 18$. Інакше $P = 0$.
- Підзадача 6.
 - Якщо $X \leq 50$, $P = 31$.
 - Якщо $51 \leq X \leq 52$, $P = 21$.
 - Якщо $53 \leq X$, $P = 0$.

Зверніть увагу, що результат по підзадачі обчислюється як мінімум результатів по кожному тесту цієї підзадачі.

Модуль перевірки із прикладу

Модуль перевірки з прикладу читає вхідні дані у наступному форматі:

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

Якщо Ваша програма отримує результат **Accepted**, модуль перевірки виводить `Accepted: q`, де q - кількість викликів `ask`.

Якщо Ваша програма отримує результат **Wrong Answer**, він виводить `Wrong Answer: MSG`, де `MSG` може бути наступним:

- `answered not exactly once`: Процедура `answer` була викликана не рівно 1 раз.
- `w is invalid`: довжина w , що передано до `ask` не M , або $w[i]$ не є 0 або 1 для якогось i ($0 \leq i \leq M - 1$).
- `more than 100 calls to ask`: Функцію `ask` викликали більше ніж 100 разів.
- `{s, t} is wrong`: Процедура `answer` викликана з невірною парою `s i t`.