



公路通行費

日本的城市間是由公路網連結的，這個公路網由 N 個城市和 M 條公路所組成，每一條公路連接一對不同的城市，沒有兩條公路會連接相同的一對城市，城市的編號是 0 到 $N - 1$ ，公路的編號是 0 到 $M - 1$ ，你可以在任一公路上雙向通行，你也可以利用公路從任一城市到達其他的任一城市。

在公路上開車要付通行費。一條公路的通行費會根據該公路的「路況」有所不同。路況分為「輕度」和「重度」兩種。路況為「輕度」時，通行費是 A 元日幣，路況為「重度」時，通行費是 B 元日幣。 $A < B$ 保證成立，注意 A 和 B 會是已知的。

你有一部機器，給定所有公路的路況後，可以算出在兩個城市 S 和 T 之間($S \neq T$)，於此路況下，最少的總通行費。

然而這部機器只是部原型機， S 和 T 的值是固定的(也就是寫死在機器裏)，而且是未知的。你想要找出 S 和 T ，於是，你打算指派某些路況到機器中，再利用它所輸出的通行費數值來推測出 S 和 T 。因為指派路況需要成本，你並不想使用機器太多次。

實作細節

你需要完成以下程序：

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

- N : 城市的個數。
- U 和 V : 長度為 M 的兩個陣列， M 是連接城市的公路數量。對於每一個 i ($0 \leq i \leq M - 1$)，公路 i 連接城市 $U[i]$ 和 $V[i]$ 。
- A : 路況為輕度時的通行費。
- B : 路況為重度時的通行費。
- 這個程序於每一筆測資只會被呼叫一次。
- 請注意 M 的值是陣列的長度，這個值的取得如實作注意事項所示。

`find_pair` 程序可以呼叫以下的函式：

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

- w 的長度必須是 M ， w 陣列描述路況。
- 對於每一個 i ($0 \leq i \leq M - 1$)， $w[i]$ 提供公路 i 的路況。 $w[i]$ 的值不是 0 就是 1 。
 - $w[i] = 0$ 表示公路 i 的路況是輕度的。

- $w[i] = 1$ 表示公路 i 的路況是重度的。
- 這個函式回傳在兩個城市 S 和 T 之間，於 w 的路況下，最少的總通行費。
- 這個函式最多可以被呼叫 100 次(對於每一筆測資)。

`find_pair` 應該呼叫以下程序來報出答案：

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

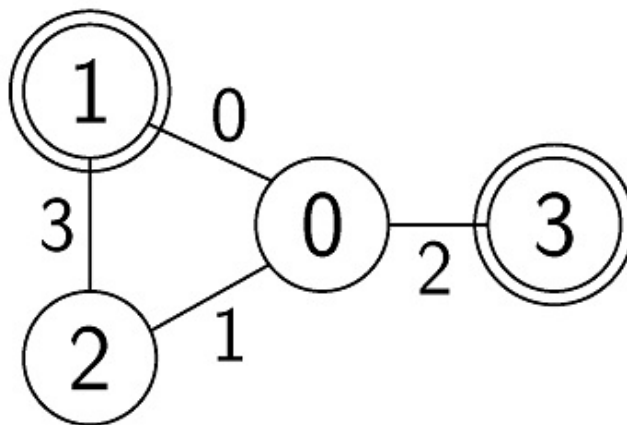
- s 和 t 必須是 S 和 T 這一對(順序沒關係)。
- 這個程序必須被呼叫恰好一次。

如果以上條件有些沒成立，你的程式會被判定為 **Wrong Answer**，否則你的程式會被判定為 **Accepted**，而你的分數會根據呼叫 `ask` 的次數來計算(詳見子題說明)。

範例

假設 $N = 4, M = 4, U = [0, 0, 0, 1], V = [1, 2, 3, 2], A = 1, B = 3, S = 1$, 和 $T = 3$

評分程式呼叫 `find_pair(4, [0, 0, 0, 1], [1, 2, 3, 2], 1, 3)`



在上圖中，邊的號碼 i 對應到公路 i 。對於 `ask` 可能出現的呼叫，和相對應的回傳數值如下所列：

Call	Return
<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

對於 `ask([0, 0, 0, 0])` 的函式呼叫，每條公路的路況都是輕度，所以每條路的通行費都是 1。從 $S = 1$ 到 $T = 3$ 最便宜的路徑是 $1 \rightarrow 0 \rightarrow 3$ ，這條路徑的總通行費是 2，因此這函式回傳 2。

在正確的答案中，`find_pair` 程序應該呼叫 `answer(1, 3)` 或 `answer(3, 1)`。

壓縮包中的檔案 `sample-01-in.txt` 對應到這個範例，壓縮包中也有其他的範例輸入。

限制

- $2 \leq N \leq 90\,000$
- $1 \leq M \leq 130\,000$
- $1 \leq A < B \leq 1\,000\,000\,000$
- 對於每一個 i ， $0 \leq i \leq M - 1$
 - $0 \leq U[i] \leq N - 1$
 - $0 \leq V[i] \leq N - 1$
 - $U[i] \neq V[i]$
- $(U[i], V[i]) \neq (U[j], V[j])$ 和 $(U[i], V[i]) \neq (V[j], U[j])$ ($0 \leq i < j \leq M - 1$)
- 你可以利用公路從任一城市到達其他的任一城市。
- $0 \leq S \leq N - 1$
- $0 \leq T \leq N - 1$
- $S \neq T$

在這個問題中，評分程式不是即時調整的。意即在評分程式開始跑的時候 S 和 T 就固定了，它不會隨著你解答的查詢而變更。

子題

1. (5 分) S 或 T 其中之一是 0， $N \leq 100$ ， $M = N - 1$
2. (7 分) S 或 T 其中之一是 0， $M = N - 1$
3. (6 分) $M = N - 1$ ， $U[i] = i$ ， $V[i] = i + 1$ ($0 \leq i \leq M - 1$)
4. (33 分) $M = N - 1$
5. (18 分) $A = 1$ ， $B = 2$
6. (31 分) 無額外限制

假設你的程式會被判定為 **Accepted**，且呼叫 `ask` 函式 X 次，則在各別子題中，你在該筆測資的得分 P 依下列方式計算：

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

請注意你的每一個子題的分數，是該子題的所有測資分數的最小值。

範例評分程式

範例評分程式以下列格式讀取輸入：

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

若你的程式會被判定為 **Accepted**，範例評分程式會印出 **Accepted: q**，其中 q 是呼叫 `ask` 的次數。

若你的程式會被判定為 **Wrong Answer**，它會印出 **Wrong Answer: MSG**，當中的 **MSG** 是下列之一：

- **answered not exactly once:** `answer` 程序被呼叫不是恰好一次。
- **w is invalid:** w 的長度提供給 `ask` 的不是 M ，或者有某個 i ($0 \leq i \leq M - 1$) 的 $w[i]$ 既不是 0 也不是 1。
- **more than 100 calls to ask:** `ask` 函式被呼叫超過 100 次。
- **{s, t} is wrong:** 用不正確的一對 s 和 t 呼叫 `answer` 程序。