

B. 都市の代表者の交換 (Swapping Cities)

Time limit	2 s
Memory limit	512 MB

問題文

インドネシアには 0 から $N - 1$ までの番号が付けられた N 個の都市が存在する。さらに M 本の双方向に通行可能な道路があり、 0 から $M - 1$ までの番号が付けられている。各道路は 2 つの異なる都市を繋いでいる。道路 i は都市 $U[i]$ と都市 $V[i]$ を繋いでおり、車で通行すると $W[i]$ 単位のガソリンを消費する。また、任意の 2 つの異なる都市のペアについて、 1 本以上の道路を通ることで互いに行き来することが可能である。

これからの Q 日間のそれぞれの日において、ある 2 都市が政治的関係の樹立を試みる。特に、 j 日目には、都市 $X[j]$ が都市 $Y[j]$ との政治的関係の樹立を試みる。そのためには、都市 $X[j]$ は代表者を車で都市 $Y[j]$ に派遣する必要がある。同様に、都市 $Y[j]$ も代表者を車で都市 $X[j]$ に派遣する必要がある。

渋滞を避けるため、双方の車はどの時間にもすれ違ってはならない。特に、双方の車は同じ時間に、同じ都市にいてはならない。また、双方の車は同時に、同じ道を互いに反対方向に走ってはならない。さらに、道路を通行する車は必ずその道路を走りきり、反対側の都市に着く必要がある (つまり、道路の途中で車が U ターンすることは許されない)。しかし、車が同じ都市や道路を複数回訪れることは許される。また、車はいつでも今いる都市で待機することもできる。

燃料を多く貯められる車は高価になるので、両都市は、双方の車の燃料積載量の最大値が最小になるように、双方の車の進み方を決めたい。各都市には無限にガソリンを供給することができるガソリンスタンドが設置されているため、 1 つの車が必要とする燃料積載量は、その車が通行する各道路におけるガソリン消費量の最大値と等しい。

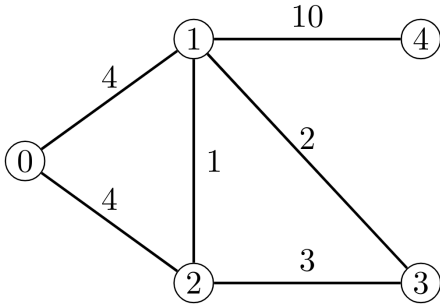
課題

関数 `init` および `getMinimumFuelCapacity` を実装せよ。

- `init(N, M, U, V, W)` - この関数は、`getMinimumFuelCapacity` が呼び出されるより前に、採点プログラム (grader) によってちょうど 1 回呼び出される。
 - N : 都市の個数を表す整数。
 - M : 道路の本数を表す整数。
 - U : M 個の整数からなる配列で、道路が繋ぐ 2 都市のうち、片方の都市を表す。
 - V : M 個の整数からなる配列で、道路が繋ぐ 2 都市のうち、もう片方の都市を表す。
 - W : M 個の整数からなる配列で、道路におけるガソリン消費量を表す。
- `getMinimumFuelCapacity(X, Y)` - この関数は、採点プログラムによってちょうど Q 回呼び出される。
 - X : 1 番目の都市を表す整数。
 - Y : 2 番目の都市を表す整数。
 - この関数は、問題文で説明されているルールに従って、都市 X が代表者を都市 Y に派遣し、また、都市 Y が代表者を都市 X に派遣できる場合は、双方の車の燃料積載量の最大値を最小化したときの値を整数として返す必要がある。ただし、問題文のルールに従って両都市が互いに代表者を派遣できない場合は、 -1 を返さなければならない。

例

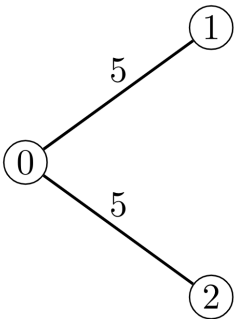
最初の例は、 $N = 5$, $M = 6$, $U = [0, 0, 1, 1, 1, 2]$, $V = [1, 2, 2, 3, 4, 3]$, $W = [4, 4, 1, 2, 10, 3]$, $Q = 3$, $X = [1, 2, 0]$, $Y = [2, 4, 1]$ である。以下は、この例を図示したものである:



採点プログラムは、最初に `init(5, 6, [0, 0, 1, 1, 1, 2], [1, 2, 2, 3, 4, 3], [4, 4, 1, 2, 10, 3])` を呼び出す。その後、採点プログラムは以下の関数を呼び出す:

- `getMinimumFuelCapacity(1, 2)`. はじめに、都市 1 にある車が都市 3 に移動する。次に、都市 2 にある車が都市 1 に移動する。最後に、都市 3 にある車が都市 2 に移動する。このとき、双方の車の燃料積載量の最大値は、都市 3 から都市 2 に移動するときに必要な 3 単位のガソリンである。これより少ない燃料積載量で達成する進み方は存在しないので、この関数は 3 を返さなければならない。
- `getMinimumFuelCapacity(2, 4)`. 都市 4 を発着するどのような車も 10 単位の燃料積載量を必要とするため、この関数は 10 を返さなければならない。
- `getMinimumFuelCapacity(0, 1)`. この関数は 4 を返さなければならない。

2 番目の例は、 $N = 3, M = 2, U = [0, 0], V = [1, 2], W = [5, 5], Q = 1, X = [1], Y = [2]$ である。以下は、この例を図示したものである:



採点プログラムは、最初に `init(3, 2, [0, 0], [1, 2], [5, 5])` を呼び出す。その後、採点プログラムは以下の関数を呼び出す:

- `getMinimumFuelCapacity(1, 2)`. 都市 1 にいる車が、もう一方の車とすれ違わずに、都市 2 に移動することは不可能なので、この関数は -1 を返さなければならない。

制約

- $2 \leq N \leq 100\,000$.
- $N - 1 \leq M \leq 200\,000$.
- $0 \leq U[i] < V[i] < N$.
- 任意の 2 つの都市のペアについて、それらを繋ぐ道は 1 本以下である。
- 任意の 2 つの都市のペアについて、道路を 1 本以上使うことで互いに行き来できる。
- $1 \leq W[i] \leq 10^9$.
- $1 \leq Q \leq 200\,000$.
- $0 \leq X[j] < Y[j] < N$.

小課題 1 (6 点)

- 各都市に繋がっている道路は 2 本以下である.

小課題 2 (7 点)

- $M = N - 1$.
- $U[i] = 0$.

小課題 3 (17 点)

- $Q \leq 5$.
- $N \leq 1\,000$.
- $M \leq 2\,000$.

小課題 4 (20 点)

- $Q \leq 5$.

小課題 5 (23 点)

- $M = N - 1$.

小課題 6 (27 点)

- 追加の制約はない.

採点プログラムのサンプル

採点プログラムのサンプルは, 入力を以下の形式で読み込む:

```
N M
U[0] V[0] W[0]
U[1] V[1] W[1]
.
.
.
U[M-1] V[M-1] W[M-1]
Q
X[0] Y[0]
X[1] Y[1]
.
.
.
X[Q-1] Y[Q-1]
```

`getMinimumFuelCapacity` を呼び出すたびに, 採点プログラムのサンプルは, 関数の戻り値を出力する.