L3/M1, 2005–2006

Algorithmique effective – TD 9
Graphes et flots

1 Sabotage

Exercice 1 Sabotage. UVa : 10480.
The regime of a small but wealthy dictatorship has been abruptly overthrown by an unexpected rebellion. Because of the enormous disturbances this is causing in world economy, an imperialist military super power has decided to invade the country and reinstall the old regime.

For this operation to be successful, communication between the capital and the largest city must be completely cut. This is a difficult task, since all cities in the country are connected by a computer network using the Internet Protocol, which allows messages to take any path through the network. Because of this, the network must be completely split in two parts, with the capital in one part and the largest city in the other, and with no connections between the parts.

There are large differences in the costs of sabotaging different connections, since some are much more easy to get to than others.

Write a program that, given a network specification and the costs of sabotaging each connection, determines which connections to cut in order to separate the capital and the largest city to the lowest possible cost.
Input

Input file contains several sets of input. The description of each set is given below.

The first line of each set has two integers, separated by a space: First one the number of cities n in the network, which is at most 50. The second one is the total number of connections, m, at most 500.

The following m lines specify the connections. Each line has three parts separated by spaces: The first two are the cities tied together by that connection (numbers in the range $1 - n$). Then follows the cost of cutting the connection (an integer in the range 1 to 40,000,000). Each pair of cities can appear at most once in this list.

Input is terminated by a case where values of n and m are zero. This case should not be processed. For every input set the capital is city number 1, and the largest city is number 2.

Output

For each set of input you should produce several lines of output. The description of output for each set of input is given below.

The output for each set should be the pairs of cities (i.e. numbers) between which the connection should be cut (in any order), each pair on one line with the numbers separated by a space. If there is more than one solution, any one of them will do.

Print a blank line after the output for each set of input.

Sample Input

```
5 8
1 4 30
1 3 70
5 3 20
4 3 5
4 5 15
5 2 10
3 2 25
2 4 50
5 8
1 4 30
1 3 70
5 3 20
4 3 5
4 5 15
5 2 10
3 2 25
2 4 50
0 0
```

Sample Output

```
4 1
3 4
3 5
3 2

4 1
3 4
3 5
3 2
```

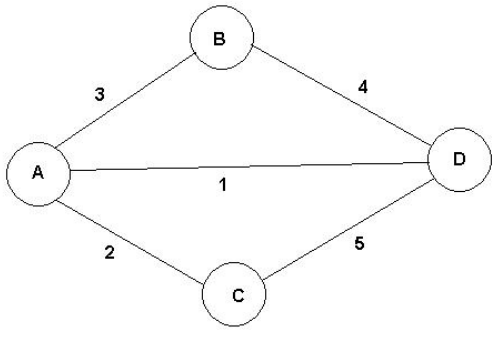
2 Dessert

2.1 Comparaison

Si ce n'est déjà fait pour le problème précédent, programmer l'algorithme probabiliste de la coupe minimum vu en cours. Reprendre l'algorithme de flot maximum déjà programmé pour le DM3 ; il peut aussi servir à trouver une coupe minimum dans un graphe en calculant les flots max entre un sommet s et chacun des autres sommets, prenant le minimum, puis en utilisant l'équivalence « max-flot = min-cut ». Comparer les vitesses d'exécution des deux algorithmes. Étudier la probabilité d'erreur en pratique de l'algorithme probabiliste de min-cut. Vous mettrez les résultats (courbes, tableaux, considérations théoriques, etc.) dans le rapport.

2.2 Un réseau déraisonnable

Exercice 2 Data flow. UVa : 10594.
In the latest Lab of IIUC, it requires to send huge amount of data from the local server to the terminal server. The lab setup is not yet ready. It requires to write a router program for the best path of data. The problem is all links of the network has a fixed capacity and cannot flow more than that amount of data. Also it takes certain amount of time to send one unit data through the link. To avoid the collision at a time only one data unit can travel i.e. at any instant more than one unit of data cannot travel parallel through the network. This may be time consuming but it certainly gives no collision. Each node has sufficient buffering capability so that data can be temporarily stored there. IIUC management wants the shortest possible time to send all the data from the local server to the final one.



Link Capacity: 10

For example, in the above network if anyone wants to send 20 unit data from A to D, he will send 10 unit data through AD link and then 10 unit

data through $AB - BD$ link which will take $10 + 70 = 80$ unit time.

Input

Each input starts with two positive integers N ($2 \leq N \leq 100$), M ($1 \leq M \leq 5000$). In next few lines the link and corresponding propagation time will be given. The links are bidirectional and there will be at most one link between two network nodes. In next line there will be two positive integers D, K where D is the amount of data to be transferred from 1st to N-th node and K is the link capacity. Input is terminated by EOF.

Output

For each dataset, print the minimum possible time in a line to send all the data. If it is not possible to send all the data, print `Impossible.`. The time can be as large as 10^{15}.

Sample input

```
4 5
1 4 1
1 3 3
3 4 4
1 2 2
2 4 5
20 10
4 4
1 3 3
3 4 4
1 2 2
2 4 5
20 100
4 4
1 3 3
3 4 4
1 2 2
2 4 5
20 1
```

Sample output

```
80
140
Impossible.
```