

# Lab05-Graph Algorithm

Algorithm and Complexity, Xiaofeng Gao, Spring 2022.

\* If there is any problem, please contact TA Jialin Lyu.

\* Name: Zhenran Xiao Student ID: 520030910281 Email: xiaozhenran@sjtu.edu.cn

- Given an undirected graph  $G = (V, E)$  in Figure 1, supposed that vertices are numbered from  $A$  to  $L$ , please complete the following tasks:

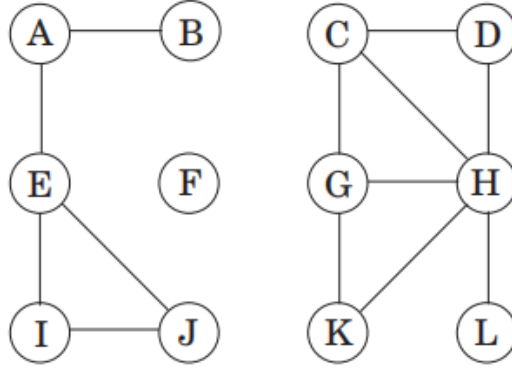


Figure 1: A sample of undirected graph.

- Please use DFS and BFS respectively to establish all the connected components. It is assumed that always start from the vertex with the smallest lexicographic order and visit its adjacent vertices in order of increasing lexicographic order.
- According to question (a), both DFS and BFS yield a tree (also possibly forest), please draft the tree and list all the tree, forward, back and cross edges (if there exists) respectively for DFS and BFS. Moreover, please list PREVISIT and POSTVISIT interval  $[pre(v), post(v)]$  for each vertex  $v$  in DFS.

## Solution.

- There are three connected components established by DFS:

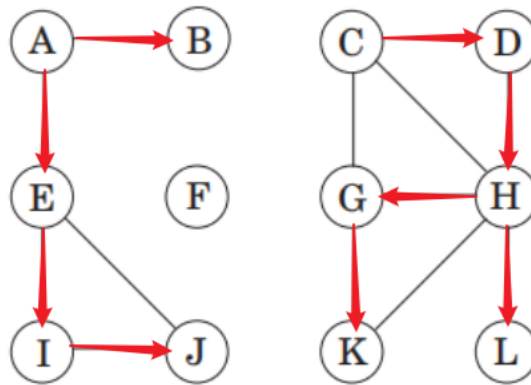


Figure 2: Connected componets established by DFS

There are three connected components established by BFS:

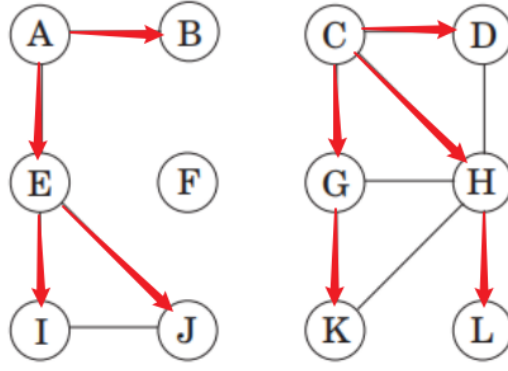


Figure 3: Connected componets established by BFS

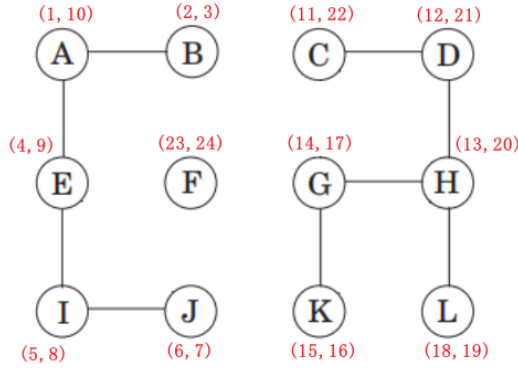


Figure 4: Forest yielded by DFS

(b) The forest yielded by DFS with root A, root C and root F.

Forward edges:  $(A,I), (A,J), (E,J), (C,H), (C,G), (C,L), (C,K), (D,G), (D,L), (D,K), (H,K)$   
 Back edges:  $(I,A), (J,A), (J,E), (H,C), (G,C), (L,C), (K,C), (G,D), (L,D), (K,D), (K,H)$   
 Cross edges:  $(E,B), (I,B), (J,B), (L,K), (L,G)$ , and edges from F to all the other vertexes.

The forest yielded by BFS with root A, root C and root F.

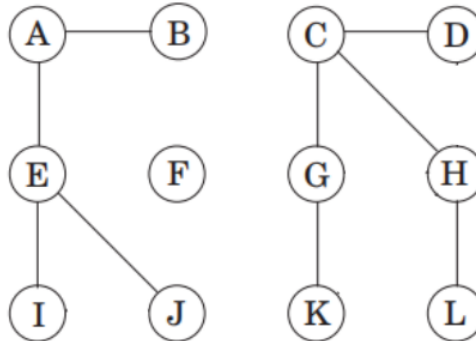


Figure 5: Forest yielded by BFS

Forward edges:  $(A,I), (A,J), (C,K), (C,L)$   
 Back edges:  $(I,A), (J,A), (K,C), (L,C)$

Cross edges:  $(E,B), (J,I), (I,B), (J,B), (G,D), (H,D), (L,K), (L,G)$ , and edges from  $F$  to all the other vertexes.

□

2. In addition to edge capacities, a flow network has vertex capacities. That is each vertex has a limit on how much flow can pass through. Show how to transform a flow network  $G = (V, E)$  with vertex capacities into an equivalent flow network  $G' = (V', E')$  without vertex capacities, such that a maximum flow in  $G'$  has the same value as a maximum flow in  $G$ . How many vertices and edges does  $G'$  have? (An illustrative example is shown in Figure 6, you may give the solution of Figure 6 first, and then give the general solution.)

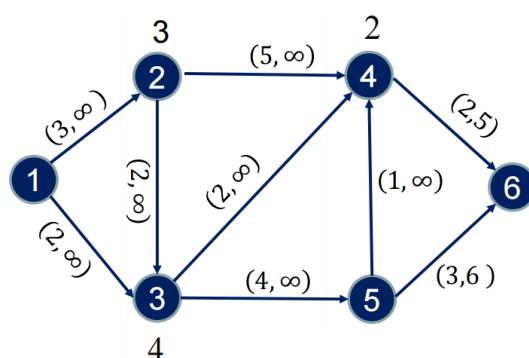


Figure 6: A sample of flow network with vertex capacities. (The number adjacent to each vertex represents the vertex capacity. The first item of each directed edge represents its value, and the second item represents its edge capacity.)

### Solution.

Transforming method: divide a vertex  $m$  with vertex capacity into two vertices  $u$  and  $v$  without vertex capacity; add an edge from  $u$  to  $v$  with edge capacity equal to the original vertex capacity; let the original edges that flow into vertex  $m$  flow into  $u$ , and the original edges that flow out from  $m$  flow out from  $v$ .

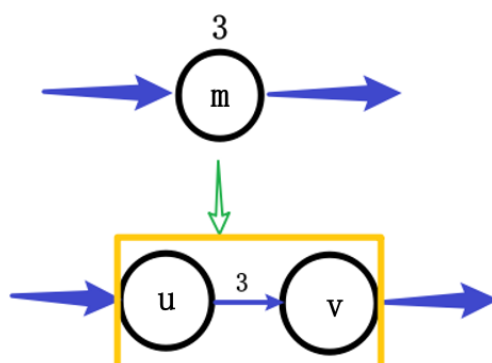


Figure 7: The image of transforming method

So the equivalent flow network of the network in Figure 6 is:

□

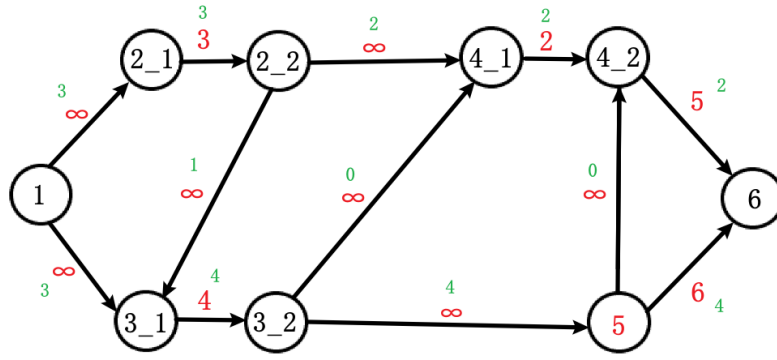


Figure 8: The equivalent flow network of the network in Figure 6

You can choose one of the following questions (Question 3 and Question 4) to answer. You will get extra bonus points if you finish both of them correctly. Otherwise, you will get the highest score among them.

3. *Information Transmission.* During a war, the information from different regions in a country can be transmitting by the following 2 sending modes:

- **Telegram mail mode:** If city  $A$  has the conditions to transmit telegram to city  $B$ , then city  $A$  can transmit telegram to city  $B$  **in one direction** at a certain time. (The unit of time is minute, and the time required is not necessarily the same if it is transmitted in the opposite direction.) In addition, if the telegram can't be transmitted directly, city  $A$  can also transmit telegram to other cities, then other cities maybe have the conditions to transmit it to the city  $B$ ;
- **Internet transmission mode:** If both city  $A$  and city  $B$  can send telegram to each other (telegram sent from any one of the cities can be delivered to the other one, and it is allowed to transmit information passing through other cities), it can be considered that city  $A$  and city  $B$  are in the same region. In that case, the two cities  $A$  and  $B$  can transmit information to each other through the Internet immediately and without spending any extra time;

Given the required time of sending telegram between different cities in the country, please help to calculate the minimum time to send information between different cities or to tell that the message cannot be transmitted.

#### Input Specification:

There are multiple test cases. The first line of the input is an integer  $T(1 \leq T \leq 10)$ , indicating the number of test cases. Then  $T$  test cases follow.

The first line of each test case contains two integer separated by a space,  $N(1 \leq N \leq 500)$  and  $E(0 \leq E \leq N^2)$ , indicating the number of cities (numbered from 1 to  $N$ ) and the number of agreements on sending telegrams, respectively.

Then  $E$  lines follows, each line contains three integers separated by spaces,  $X, Y, M(1 \leq X, Y \leq N, 1 \leq M \leq 1000)$ , indicating that there exist an agreement to send a telegram from city  $X$  to city  $Y$ , and that such a telegram will be delivered in  $M$  minutes.

After that, there will be a line with an integer  $K(0 \leq K \leq 100)$ , the number of queries.

Finally, there will be  $K$  lines, each representing a query and containing two integers separated by a space,  $A, B(1 \leq A, B \leq N)$ . You must determine the minimum time to send information from city  $A$  to city  $B$ .

## Output Specification:

For each test case output  $K$  lines. The  $i$ -th line should contain an integer, indicating the minimum time (in minutes) of sending telegram of the  $i$ -th query. If there aren't communication conditions between the cities of the query, print `Impossible` (without quotes) in one line instead.

Sample Input:	Sample Output:
2	0
4 5	6
1 2 5	6
2 1 10	0
2 3 6	Impossible
3 4 8	10
4 3 7	Impossible
5	0
1 2	
1 3	
1 4	
4 3	
4 1	
3 3	
1 2 10	
2 3 1	
3 2 1	
3	
1 3	
3 1	
3 2	

**Remark:** The input data `Lab05-Telegram.in` (sample test cases only) and the template code `Lab05-Telegram.cpp` are attached on the course webpage. Please include your `Lab05-Telegram.cpp` file in your uploaded .rar or .zip file.

- Please briefly describe your algorithm and analyze its time complexity and space complexity.
- Try to write a C/C++ code to solve this problem, you only need to complete the `TODO` part in `Lab05-Telegram.cpp`. Your program will be judged by the online judge system, including several test cases, half for the test data which equivalent to the sample test case, and another half for other corner and huge test cases.

## Solution.

- For every query City A to City B, use Floyd-Warshall Alogrithm twice to find the shortest sending time from A to B and from B to A. If both the two path exist, the actual sending time is 0(by Internet). If only the path from A to B exists, the sending time is the path. If the path from A to B doesn't exist, it is impossible to mail between the two cities.

The time complexity:  $O(N^3)$

The space complexity:  $O(N^2)$

- Please check it in `Lab05-Telegram.cpp`.

□



```
Microsoft Visual Studio 调试控制台
0
6
6
0
Impossible
10
Impossible
0
```

Figure 9: Output of Lab05-Telegram.cpp

4. *Punch in Shanghai Metro.* Little Gyro is a subway fan. This holiday he finally has the opportunity to come to Shanghai <sup>a</sup> the capital of the subway, and want to have fun in taking subway.

The pricing rule of Shanghai Metro is that the starting price is 3 yuan, and the shortest distance between the departure station and the arrival station is increased by 1 yuan per  $K$  kilometers.

For example, let  $K = 10$ , Hongqiao Railway Station is 20 kilometers away from People's Square Station, so that the fare between them is  $3 + 2 = 5$  yuan.

In order to get the greatest satisfaction, Little Gyro decides to take the subway in the following way: If Little Gyro gets on at a certain station (set it as subway station  $A$ ), for all arrival stations with the same fare, Little Gyro will only exit at the station with the farthest distance or terminal, take a punch photo outside the station, and then take subway to the next station in the same way.

Given the departure station (Little Gyro would also take a punch card photo before setting out), while taking the subway, Little Gyro suddenly becomes curious about how many punch photos at most and at which subway stations could he take during the whole journey.

**Hint:** Subway refers to the urban traffic rail system which mainly runs underground. Generally speaking, the subway is composed of several stations and many different lines running in both directions.

When two or more lines pass through the same station, you can transfer and change the line you take. For example, both Line 1 and Line 2 of the Shanghai Metro pass through the People's Square Station, then when you arrive at the People's Square by Line 1, you can transfer to Line 2 to go to various stations of Line 2. In that case, you needn't to exit the station for transfer (Outbound transfer is not considered), so that you can't get a punch photo at the same time. As a result, there are no restrictions on transfer while Little Gyro taking the subway.

### Input Specification:

There are multiple test cases. The first line of the input is an integer  $T$  ( $1 \leq T \leq 10$ ), indicating the number of test cases. Then  $T$  test cases follow.

The first line of each test case contains three integers  $N, M, K$  ( $1 \leq N \leq 200, 1 \leq M \leq 1500, 1 \leq K \leq 10^6$ ), indicating the number of stations, the number of lines of Shanghai Metro, and increase the fare by 1 yuan per  $K$  kilometers, respectively.

The following  $M$  lines describe the information of Shanghai Metro lines, with the following format:

$< Station1 > < Space > < Distance > < Space > < Station2 > < Space > < Distance > < Space > < Station3 > \cdots < Station(X - 1) > < Space > < Distance > < Space > < StationX >$



Figure 10: The map of Shanghai Metro.

Where  $\langle StationX \rangle$  is a number from 1 to  $N$ .  $\langle Distance \rangle$  between two station numbers  $\langle Station(X-1) \rangle, \langle StationX \rangle$  refers to the distance of two stations on the line. The distance between two stations is a positive integer not greater than  $10^6$ . The stations on each line are unique.

**Hint:** It maybe exists multiple directly connected lines between two subway stations, and the distances between them are not necessarily equal.

After that, there will be a line with an integer  $Q(0 \leq Q \leq 200)$ , the number of queries of the departure subway stations.

Finally, there will be  $Q$  lines, each representing a query and containing one integer  $X_i(1 \leq X_i \leq N)$ , indicating the departure subway station that Little Gyro choose.

### Output Specification:

For each query output two lines, the first line consists one integer, indicating the maximum number of punch photos that Little Gyro could take. The second line consists several numbers, indicating the station number (within the ascending order) that Little Gyro could punch in.

Please, **DO NOT** output extra spaces at the end of each line, or your program may be considered incorrect!

Sample Input:	Sample Output:
2	5
6 2 6	1 2 4 5 6
5 6 2 6 6	5
1 6 2 4 3 1 4	1 2 4 5 6
4	6
5	1 2 3 4 5 6
4	5
3	1 2 4 5 6
2	2
6 2 6	1 3
1 1 2 1 3	3
4 2 5 2 6	4 5 6
4	2
1	4 6
5	3
4	1 2 3
2	

**Remark:** The input data `Lab05-SHMetro.in` (sample test cases only) and the template code `Lab05-SHMetro.cpp` are attached on the course webpage. Please include your `Lab05-SHMetro.cpp` file in your uploaded .rar or .zip file.

- Please briefly describe your algorithm and analyze its time complexity and space complexity.
- Try to write a C/C++ code to solve this problem, you only need to complete the `TODO` part in `Lab05-SHMetro.cpp`. Your program will be judged by the online judge system, including several test cases, half for the test data which equivalent to the sample test case, and another half for other corner and huge test cases.

**Remark:** You need to include your .pdf and .tex files in your uploaded .rar or .zip file.