19I510 Design and Analysis of Algorithms

Exercise 5

Graph Algorithms

Learning Objective

1. Apply graph traversal techniques to find the connectedness between the vertices
2. Adapt graph algorithms for tasks which has 1….n links between the data points
3. Given n tasks and their dependencies, find the ordering in which tasks can be executed. If task 1 is dependent on task 2 and task 3, then tasks 2 and 3 should be executed before task 1.

**Input Format**First line contains two integers m and n where m represents the number of tasks and n denote the number of dependencies. Tasks are numbered from 1 to n.

The next n lines contain two integers x and y representing the fact that task i should be executed before task j

**Output Format**Print the ordering of the tasks. If ordering is not possible due to cyclic dependency output "NOT POSSIBLE"

**Sample Input**

5 4

1 2

2 3

1 3

1 5

**Sample Output**  
1 4 2 5 3

1. Government of India is planning to provide WIFI connectivity for Coimbatore under the SmartCity Scheme. They have obtained the map of locations where an access point can be placed. The edges between the locations indicate reachability of the locations. If the access point is created at location i, determine whether WIFI is accessible at all the other locations.

Access point can be accessed at location k only if its distance from access point is below th (th is number of links to connect the local k to location i).

**Input Format**First line contains the number of vertices(n) and number of edges in the graph. next line contains two integers which denote the location j and threshold th.

Following m lines contains two integers v1 and v2 separated by space denoting the fact that vertices v1 and v2 are connected directly. Vertices are numbered from 1 to n.

**Output Format**Print the list of locations for which the WIFI is accessible (ordered by distance and location number)

**Sample Input**

**10 14**

**3 1**

**1 2**

**2 3**

**3 4**

**4 5**

**5 3**

**3 6**

**3 10**

**10 9**

**9 6**

**3 8**

**3 7**

**9 8**

**8 7**

**7 6**

**Sample Output**

2 4 5 6 7 8 10

1. XYZ company LAN network has n computers and m connections. Your task is to find out if system A can send a message to system B. The computers are numbered 1,2,…,n. System A is numbered as 1 and System B is numbered as n.

**Input Format**First line contains the number of computers(n) and number of connections in the graph.

Following m lines contains two integers v1 and v2 separated by space denoting the fact that computers v1 and v2 are connected directly. Vertices are numbered from 1 to n.

**Output Format**

Print "POSSIBLE" if system A can send message to system B. Else print the message "IMPOSSIBLE"

**Sample Input**

5 5  
1 2  
1 3  
1 4  
2 3  
5 4

**Sample Output**

POSSIBLE

1. There are n cities and m flight connections. Your task is to check if you can travel from any city to any other city using the available flights.

**Input Format**

The first input line has two integers n and m denoting the number of cities and flights. The cities are numbered 1,2,…,n.

After this, there are m lines describing the flights. Each line has two integers a and b indicating that there is a flight from city a to city b. All flights are one-way flights.  
  
**Output Format**  
  
Print "YES" if all routes are possible, and "NO" otherwise.

**Sample Input**

4 5  
1 2  
2 3  
3 1  
1 4  
3 4

**Sample Output**

NO

1. To help capture criminals on the run, the police are introducing a new computer system. The area covered by the police contains **N** cities and **E** bidirectional roads connecting them. The cities are labelled 1 to **N**. The police often want to catch criminals trying to get from one city to another. Inspectors, looking at a map, try to determine where to set up barricades and roadblocks.

Given three cities A, B and C, the system should be able to answer the following query, whether the criminal can travel from city A and B if roads in city C is blocked?

Write a program that implements the described system.

**Input Format**

The first line contains two integers **N** and **E** (2 ≤ N ≤ 105, 1 ≤ E ≤ 5\*105), the number of cities and roads. Each of the following **E** lines contains two distinct integers between 1 and **N** – the labels of two cities connected by a road. There will be at most one road between any pair of cities.

Last line will contain the three vertices A, B and C

**Output Format**  
  
Print "YES" if criminal can escape else print "NO"

**Sample Input**

6 7  
1 2  
1 3

2 3

2 4

4 5

4 6

5 6

1 5 2

**Sample Output**

NO