January 2021 CSE204: Data Structures and Algorithms I Sessional

Offline on Graphs and Graph Traversals

Mr Goodman was a famous computer scientist. He invented a new programming language that is superior to all other languages of all time. Using his invented language, one can do the unimaginable in both good or bad ways. Badman wanted to steal his work and use it against humanity. To protect society from the evil of Badman, Mr Goodman had broken down the invention into several pieces and hid it in different parts of Mamaland. Hearing the news, Badman attacked Mamaland and caused significant destruction but failed to recover the pieces.

A long time has passed since the attack. Mamaland is abandoned now. Lots of roads are damaged and unusable. A group of friends from the CSE department of TEUB has found a secret **chotha** of Mr Goodman in their library. They have come to know about the hidden pieces of Mr Goodman's invention and want to recover them. They have planned to go to different parts of Mamaland and start collecting the pieces.

While collecting the pieces, they will work following their ID in ascending order. That means, if there are three (3) friends with ID 0, 1, and 2, then first friend 0 will collect as many pieces as s/he can, then friend 1 followed by friend 2.

In this offline, your task is to find out if their mission is going to be successful or not.

Input/Output:

You will take input from the console and give output to an output file.

Input Format:

The first line has four space-separated integers **C, R, L, F,** denoting the number of **C**ities and **R**oads in Mamaland, the total number of **L**ocations where the pieces are hidden, and the number of **F**riends, respectively.

Each of the following **R** lines contains two space-separated integers C_1 and C_2, denoting two cities of Mamaland connected by a road. All the roads are two-way roads.

Each of the following **L** lines contains two space-separated integers C_x and P_x, denoting the city (C_x), where Mr Goodman hid P_x pieces.

Each of the following **F** lines contains two space-separated integers C_y and F_y, denoting the city (C_y) from where a friend (F_y) will start collecting the pieces.

Output Format:

In the first line of output, print whether the mission is successful or not. Print "Mission Accomplished" or "Mission Impossible" based on the scenario.

In the next line, print the total number of collected pieces and the total number of pieces.

In the next **F** lines, print the number of pieces collected by each friend.

See the sample I/O for further clarification.

Constraints:

$1 < C \leq 1000$
$1 < R \leq C * (C - 1) / 2$
$1 < L \leq C$
$1 < F \leq 100$
$0 \leq C_1, C_2, C_x, C_y < C$
$0 \leq F_y < F$

Solve the problem using **both BFS and DFS** algorithms **separately**.

Special Instructions:

Write ***readable, re-usable, well-structured, quality*** code. This includes but is not limited to writing appropriate functions for implementation of the required algorithms, meaningful naming of the variables, suitable comments where required, proper indentation etc.

Please **DO NOT COPY** solutions from anywhere (your friends, seniors, internet etc.). Implement the algorithms with your style of coding. Any form of plagiarism (irrespective of source or destination), will result in getting -100% marks. You have to protect your code.

Also, be informed that for the repeated offence of plagiarism, the departmental policies suggest stricter measures.

Submission Guideline:

1. Create a directory with your 7 digit student id as its name
2. Put the source files only into the directory created in 1
3. Zip the directory
4. Upload the zip into moodle

For example, if your student id is 1805123, create a directory named 1805123. Put your source files(.c, .cpp, .java, .py, .h, .hpp etc) only into 1805123. Zip 1805123 into 1805123.zip and upload the 1805123.zip into moodle.

Failure to follow the above-mentioned submission guideline will result in up to 10% penalty.

Submission Deadline:

April 23, 2021 11:55 PM

This is a hard deadline and there shall be no extensions for any reason whatsoever.

The sample I/O:

Input	Output
10 8 4 3 0 4 0 1 2 0 2 1 1 3 6 5 5 7 9 8 0 5 3 6 5 3 9 2 4 0 7 2 2 1	Mission Impossible 14 out of 16 pieces are collected 0 collected 11 pieces 1 collected 0 pieces 2 collected 3 pieces
10 8 4 3 0 4 0 1 2 0 2 1 1 3 6 5 5 7 9 8 0 5 3 6 5 3 9 2 4 0 7 2 8 1	Mission Accomplished 16 out of 16 pieces are collected 0 collected 11 pieces 1 collected 2 pieces 2 collected 3 pieces