

CSE221

Lab Assignment 05

Summer 2023

Submission Guidelines :

1. You can code all of them either in Python, CPP, or Java. But you should choose a specific language for all tasks.
2. For each task write separate python files like task2.py, task3.py, and so on. For problems that have subproblems, name those like task1A.py, task1B.py, and so on.
3. Add a hand written explanation of 3-4 lines for each of your solutions and submit that as a single document.
4. For each problem, take input from files called "**inputX_Y.txt**" and output at "**outputX_Y.txt**", where X is the task number and Y is the sample i/o number. For example, for problem 2 sample 1, the input file is this, "input2_1.txt". For problems that have subproblems, name the files like "input1a_1.txt", "output1a_1.txt" and so on. Same for output.
5. For each task include at least one input file (if any) in the submission folder.
6. Finally zip all the files and rename this zip file as per this
format:**LabSectionNo_ID_CSE221LabAssignmentNo_Summer2023.zip**
[Example:**LabSection01_21101XXX_CSE221LabAssignment05_Summer2023.zip**]
7. Don't copy from your friends.
8. You MUST follow all the guidelines, naming/file/zipping convention stated above.

Failure to follow instructions will result in a straight 50% mark deduction.

A useful tool for making graphs:

https://csacademy.com/app/graph_editor/

Task 01 [30 points]

We all know about the prerequisite course. For example, before taking CSE221, we have to complete CSE220. And prior to taking CSE220 we need to complete CSE111 and CSE110. The course sequence is as follows:

CSE110 → CSE111 → CSE220 → CSE221

In this problem, there are **N** courses in the curriculum. There are **M** prerequisite requirements of the form "course **A** has to be completed before course **B**".

Your task is to find an order in which you can complete the courses.

A) Solve it using the DFS approach. [15 points]

B) Solve it using the BFS approach. [15 points]

Input:

The first input line has two integers **N** ($1 \leq N \leq 1000$) and **M** ($1 \leq M \leq N^2$) - the number of courses and prerequisite requirements. The courses are numbered 1,2,3,...,N.

Next, there are M lines describing the requirements. Each line has two integers **A, B** ($1 \leq A, B \leq N$)- course A has to be completed before course B.

Output:

Print an order in which you can complete the courses. Please note, there could be multiple correct sequences. You can print any valid order that includes all the courses.

If there are no solutions, print "IMPOSSIBLE".

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

Sample Input 2	Sample Output 2
6 6 1 2 2 3 4 3 4 5 5 6 6 4	IMPOSSIBLE
Sample Input 3	Sample Output 3
8 10 1 2 1 4 2 4 2 5 2 3 4 6 4 5 6 5 5 3 7 8	1 7 2 8 4 6 5 3

Task 02[10 points]

The problem statement of this problem is the same as **Task 01**. The only difference is that in this task, you have to find the lexicographically smallest valid course sequence.

If you have two sequences, for example: A: $3 \rightarrow 1 \rightarrow 2 \rightarrow 4$ and B: $3 \rightarrow 1 \rightarrow 4 \rightarrow 2$. Then path A is lexicographically smaller than path B.

Input:

The first input line has two integers **N** ($1 \leq N \leq 1000$) and **M** ($1 \leq M \leq N^2$) - the number of courses and prerequisite requirements. The courses are numbered $1, 2, 3, \dots, N$.

Next, there are M lines describing the requirements. Each line has two integers **A, B** ($1 \leq A, B \leq N$)- course A has to be completed before course B.

Output:

Print the lexicographically smallest valid course sequence in which you can complete the courses.

If there are no solutions, print "IMPOSSIBLE".

Sample Input 1	Sample Output 1
5 3 3 1 1 2 4 5	3 1 2 4 5
Sample Input 2	Sample Output 2
6 6 1 2 2 3 4 3 4 5 5 6 6 4	IMPOSSIBLE
Sample Input 3	Sample Output 3
8 10 1 2 1 4 2 4 2 5 2 3 4 6 4 5 6 5 5 3 7 8	1 2 4 6 5 3 7 8

Task 03[10 points]

You are given a Directed Graph consisting of **N** vertices and **M** edges. You have to find the strongly connected components of the given graph.

Input:

The given map will be a directed and unweighted graph.

The first line contains two integers N and M ($1 \leq N, M \leq 100$) – the number of vertices and the total number of edges.

The next M lines will contain two integers u_i, v_i ($1 \leq u_i, v_i \leq N$) – denoting there is a road between u_i to v_i .

Output

Print all the strongly connected components of the given graph. See the output for better understanding.

Sample Input 1	Sample Output 1
5 5 1 2 2 3 2 4 3 1 4 5	1 2 3 4 5
Sample Input 2	Sample Output 2
8 9 1 2 1 6 2 3 3 4 4 5 4 6 5 7 6 2 7 5	1 2 3 4 6 5 7 8
Sample Input 3	Sample Output 3
4 3 1 2	1 2

2 3	3
2 4	4