National Institute of Technology, Calicut

Department of Computer Science and Engineering

CS2094 – Data Structures Lab

Assignment-5 (Main Batch)

Policies for Submission and Evaluation

You must submit your assignment in the moodle (Eduserver) course page, on or before the submission deadline. Also, ensure that your programs in the assignment must compile and execute without errors in Athena server. During evaluation your uploaded programs will be checked in Athena server only. Failure to execute programs in the assignment without compilation errors may lead to zero marks for that program.

Your submission will also be tested for plagiarism, by automated tools. In case your code fails to pass the test, you will be straightaway awarded zero marks for this assignment and considered by the examiner for awarding F grade in the course. Detection of ANY malpractice regarding the lab course will also lead to awarding an F grade.

Naming Conventions for Submission

Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar or .tar.gz). The name of this file must be ASSG<NUMBER>_<ROLLNO>_<FIRSTNAME>.zip (For example: ASSG5_BxxyyyyCS_LAXMAN.zip). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive.

The source codes must be named as ASSG<NUMBER>_<ROLLNO>_<FIRSTNAME>_<PROGRAM-NUMBER>.<extension> (For example: ASSG5_BxxyyyyCS_LAXMAN_1.c). If there is a part *a* and a part *b* for a particular question, then name the source files for each part separately as in ASSG5_BxxyyyyCS_LAXMAN_1b.c.

If you do not conform to the above naming conventions, your submission might not be recognized by some automated tools, and hence will lead to a score of 0 for the submission. So, make sure that you follow the naming conventions.

Standard of Conduct

Violations of academic integrity will be severely penalized.

Each student is expected to adhere to high standards of ethical conduct, especially those related to cheating and plagiarism. Any submitted work MUST BE an individual effort. Any academic dishonesty will result in zero marks in the corresponding exam or evaluation and will be reported to the department council for record keeping and for permission to assign F grade in the course. The department policy on academic integrity can be found at: http://cse.nitc.ac.in/sites/default/files/Academic-Integrity.pdf.

Assignment Questions

1. Consider a boolean matrix containing only 0 and 1. Any cell containing 1 is called a filled cell. Two cells are said to be *connected* if they are adjacent to each other horizontally, vertically, or diagonally.

If one or more filled cells are also connected, they form a *region*. Note that each cell in a region is connected to at least one other cell in the region but is not necessarily directly connected to all the other cells in the region.

Given an matrix, find and print the number of cells in the ***largest region*** in the matrix.
Note that there may be more than one region in the matrix.

Input Format

- The first line contains two integers *n and m* denoting the number of rows and number of columns respectively.
- Each of the *n* subsequent lines contains *m* space-separated integers

Output Format

- Print the number of cells in the largest *region* in the given matrix.

Sample Input

4 5

1 1 0 0 0

0 1 1 0 1

0 0 1 0 0

1 0 0 1 0

Sample Output

6

Explanation

The diagram below depicts three regions of the matrix; for each region, the component cells forming the region are marked with an X:

X X 0 0 0	1 1 0 0 0	1 1 0 0 0
0 **X X** 0 1	0 1 1 0 **X**	0 1 1 0 1
0 0 **X** 0 0	0 0 1 0 0	0 0 1 0 0
1 0 0 **X** 0	1 0 0 1 0	**X** 0 0 1 0

The first region has **6 cells** and the second and third region has one cell each.

2. Write a C program to implement following search in a directed graph. Vertex ordering in a graph follows the natural number sequence starting from 0.

1. Breadth First Search (BFS)
2. Depth First Search (DFS)

Input Format

First line contains two integers denoting the no of vertices(n) and edges(m).

Next m lines denote the pair of vertices representing edge. Next line contains the source vertex.

Output Format

Print the BFS and DFS traversal in two different lines

Sample Input

4 6

0 1

0 2

1 2

2 0

2 3

3 3

0

Sample Output

0 1 2 3

0 1 2 3

Note (For 3)-

In a graph with n vertices, the vertices are labeled from 0 to n-1. Use adjacency lists to store the graphs, with the vertices sorted in ascending order. The adjacency list of each node is a singly linked list that contains its adjacent nodes sorted in ascending order from left to right. The nodes in this list contain two fields, namely, the label of the adjacent node and the weight of the edge, if provided. Unless specified otherwise, the adjacency lists must be processed iteratively from left to right.

3. Write programs that compute the minimum spanning tree of a connected undirected graph using the following algorithms:

a. Kruskal's algorithm

b. Prim's algorithm

Input format:

- The first line of the input contains a positive integer n, the number of vertices in the graph, in the range 1 to 1000.
- The subsequent n lines contain the labels of the nodes adjacent to the respective nodes, sorted in ascending order from left to right.
- The subsequent n lines contain the weights of the edges corresponding to the adjacency list. The edge weights are real numbers in the range [-10000, 10000]. Further, no two edges have the same weight.

Output format:

- Print the sum of the edge weights of the minimum spanning tree as the output.

Sample Input

12

8 9

2 3 4

1 3 5 6

1 2 4

1 3 5 7 8 9

2 4 6

2 5 7 10 11

4 6 8

0 4 7 9

0 4 8

6 11

6 10

27 41

10 11 17

10 7 33 44

11 7 26

17 26 5 8 15 16

33 5 21

44 21 31 18 29

8 31 20

27 15 20 13

41 16 13

18 23

29 23

Sample Output

164

4. Given a directed graph G= (V, E), edge weights $w_e \geq 0$, source $s \in V$, find the weight of shortest path from s to all other vertex.

Input Format

- First line is the number of nodes (V) in a graph. Vertices are labelled from 0 to V-1.
- Second line is the number of edges (E) in a graph.
- Next E lines each line consist of three space separated integers x y z, where x and y denote the two nodes between which the directed edge exist and z is the weight of the edge.
- Next line consists of one integer s, where s denotes the source node.

Output Format

- Print the destination node and weight associated with it. **If there is no path print "INF"**

Sample Input

4

4

0 1 1

0 3 4

2 1 2

3 2 5

0

Sample Output

0 0

1 1

2 9

3 4