Assignment #5 – GRAPHS

Evaluation on:

Tuesday, 26 March, 2013 (for A Batch)
~~**Thursday, 28 March, 2013**~~ (for B Batch) (will be announced later)

Submission deadline (on or before):

Tuesday, 26 March, 2013, 01:30 PM (for both Batch A and Batch B)

Policies for Submission and Evaluation

You must submit your assignment in the moodle (eduserver) course page, on or before the submission deadline.

You will have to do a modification of any of the programs given in the assignment during your evaluation. If you fail to complete this program, you will be awarded a score of 0 for that assignment.

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 F grade in the course. Detection of ANY malpractice regarding the lab course will also lead to awarding an F grade.

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 `<ROLLNO>_<NAME-WITHOUT-SPACES>.zip` (Example: `B110000CS_LAXMANRAHUL.zip`). The zip file must only contain your source codes. DO NOT add any other files except your source code, into the zip archive. The source codes must be named as `<ROLLNO>_<NAME-WITHOUT-SPACES>_<PROGRAM-NUMBER>.<extension>` (Example: `B110000CS_LAXMANRAHUL_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 `B110000CS_LAXMANRAHUL_1b.cpp`

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.

About the Assignment

This assignment is based on graphs. Implement ALL questions. Your program MUST take input from text files, which is formatted as specified with each question.

Common Input Format

The first line of the input file contains a positive integer **N**, the number of vertices of the graph. The set of vertices, **V** contains vertices that are labeled **0**, **1**, **2**, …, **N-1**.

The second line contains a non-negative integer **E**, the number of edges of the graph. (Assume undirected graphs).

Then, **E** lines follow, each one containing three space-separated integers **u**, **v** and **c** (**u**, **v** ∈ **V**). This means, there is an undirected edge that connects the vertices labeled **u** and **v**, and the cost of this edge is **c**.

Sample Input

```
5
4
0 1 1
1 2 1
2 3 1
3 4 1
```

Questions

1. **BFS**

   Write a program that performs Breadth First Search in a graph.
   Take an integer from the terminal as an extra input. Do the BFS, starting from this vertex.

   Output Format

   The output must contain exactly N integers on a single line – the sequence of vertex labels, in the order they are visited.

   Sample Output (start at 0)

   ```
   0 1 2 3 4
   ```

2. **DFS**

Write a program that performs Depth First Search in a graph.
Take an integer from the terminal as an extra input. Do the DFS, starting from this vertex.

Output Format

The output must contain exactly N integers on a single line – the sequence of vertex labels, in the order they are visited.

Sample Output (start at 0)

```
0 1 2 3 4
```

3. **DIJKSTRA**

Write a program that implements Dijkstra's algorithm.
Take an integer from the terminal as an extra input. This is the source vertex for Dijkstra's algorithm.

Output Format

The output must contain exactly N integers on a single line – the list of lengths of shortest paths from the source vertex.

Sample Output (src node = 0)

```
0 1 2 3 4
```

4. **KRUSKAL**

Write a program that implements Kruskal's algorithm.

Output Format

Output the cost of the minimum spanning tree on the first line. Below that, output the adjacency matrix of the tree.

Sample Output

```
4
0 1 0 0 0
1 0 1 0 0
0 1 0 1 0
0 0 1 0 1
0 0 0 1 0
```

Also, submit DISJOINT SETS, from the previous assignment

--- Best Wishes ---