Indian Institute of Technology Roorkee

Department of Computer Science and Engineering

CSN-261: Data Structures Laboratory (Autumn 2019-2020)

Lab Assignment-8 (L8) **Date: October 9, 2019** **Duration: 2 Weeks**

General Instructions:

1. Every Lab Assignment will be performed by the students individually. No group formation is required and the evaluations will be done every week for the students individually.

Submission and Evaluation Instructions:

1. **Submit your** zipped folder (**<filename>.zip** or **<filename>.tar.gz**) through your account in Moodle through the submission link for this Lab Assignment in Moodle course site: https://moodle.iitr.ac.in/course/view.php?id=46.
2. **Hard deadline for Final submission in Moodle: October 23, 2019 (1:00 pm Indian Time).** For any submission after Final Deadline, 20% marks will be deducted (irrespective of it is delayed by a few seconds or a few days). The key to success is starting early. You can always take a break, if you finish early.
3. The submitted zipped folder (**<filename>.zip** or **<filename>.tar.gz**) must contain the following:
   (a) The source code files in a folder
   (b) A report file (**<filename>.DOC** or **<filename>.PDF**) should contain the details like:
      i. Title page with details of the student
      ii. Problem statements
      iii. Algorithms and data structures used in the implementation
      iv. Snapshots of running the codes for each problem
4. The submission by each student will be checked with others' submission to identify any copy case (using such detection software). If we detect that the code submitted by a student is a copy (partially or fully) of other's code, then the total marks obtained by one student will be divided by the total number of students sharing the same code.

Instructions for L8:

1. Objective of this Lab Assignment is to make the students familiar with different data structures while coding the programs in the Java language to solve some real-life problems.
2. The students are expected to have a basic knowledge of data structures and the Java programming language.
3. The student will have to demonstrate and explain the coding done for this Lab Assignment in the next laboratory class to be held on **October 23, 2019** for evaluation.

Problem Statement 1:
Implement Dijkstra's algorithm in Java to find all shortest paths between all pair of vertices in a weighted graph. Modify this algorithm to find all shortest paths between two nodes, if more than one occurs. Following this, compute betweenness centrality measure of each node.

Betweenness Centrality of a node/vertex, w is given as, $BC(w) = \sum_{u,v \in V} \frac{\sigma_{uv}(w)}{\sigma_{uv}}$, where, σ_{uv} is the number of all shortest paths between u and v; and $\sigma_{uv}(w)$ is the number of all shortest paths between u and v through w. (https://en.wikipedia.org/wiki/Betweenness_centrality)

Data structure that may be used: List, Set, Map, etc.
Input: A GML (Graph Modeling Language) file as a graph input.
Output: Betweenness Centrality of each node.
Note: Use JGraphT class in java (https://jgrapht.org) for this problem.

Test Case:
Input: P1.gml Adjacency Matrix:

V9	9	0	0	5	8	0	0	3	5	0
V8	1	4	3	0	0	5	0	0	0	
V7	5	2	0	0	0	5	0	0		
V6	7	8	0	0	0	0	0			
V5	0	6	0	0	8	0				
V4	8	0	0	0	0					
V3	0	8	5	0						
V2	0	0	0							
V1	4	0								
V0	0									
	V0	V1	V2	V3	V4	V5	V6	V7	V8	V9

Output:

w	BC(w)	w	BC(w)
V0	7.8333	V5	9.0000
V1	1.3333	V6	0.0000
V2	6.7500	V7	1.7500
V3	2.6667	V8	7.0000
V4	0.0000	V9	6.2500

Problem Statement 2:
Create a project/program in Java called Unscramble Word. Given a string of 'N' characters print all the words present in a dictionary of length 'M' such that 3 < M <= N.

Use dictionary present in Linux @ /usr/share/dict/words.

Implement this code in java and the student may use inbuilt data structures such as Maps, Sets, etc. (For fast execution, use of Trie is suggested).

Input: A String
Output: All unscrambled words of given string present in the dictionary categorized by length of word. Also print the total number of words of each length.

Test Case:
Input: "great"
Output:

Length: 5	greta, grate, great, retag, targe	Count: 5
Length: 4	ager, gate, gear, geta, grat, rage, rate, tare, tear	Count: 9