

Lab Assignment 6

CS 302 – Advanced Data Structures and File Processing

Spring 2020

Problem 1

You are given an undirected graph G which is represented as adjacency list. Its n nodes are labelled from 0 to $n - 1$. For each node, its list of neighbours is out of order. Implement an algorithm that sorts all these lists in total linear time. Note that linear-time for graphs means in $\mathcal{O}(|V| + |E|)$ time. Using counting sort on each list of neighbours requires a total of $\mathcal{O}(|V|^2 + |E|)$ time. Feel free to change the given graph as needed. However, do not change the IDs of vertices.

Problem 2

You are given a weighted directed acyclic graph G and a vertex s . Implement an algorithm that determines in linear time the distance from s to all other vertices. Note that edge weights can be negative. The output should be an int-array containing the distances from s to all vertices ordered by their IDs, i. e., the output should be $[d(s, v_0), d(s, v_1), \dots]$. If there is no path from s to some vertex v_i , set $d(s, v_i) := \text{Integer.MAX_VALUE}$.

Implementation

You are given a file *Lab6.java* and a file *Graph.java* (which you can download from canvas). The file *Lab6.java* contains a class Lab7 with the two functions `problem1` and `problem2`. Implement your solutions in the corresponding functions. **Do not make any changes outside of these two functions (e.g. by adding helper functions); such changes will be undone.** Do not output anything to the terminal. The class *Graph* in the file *Graph.java* contains the functions `relax`, `bfs`, and `dfs`. Feel free to use these functions.

The program already implemented in the file *Lab6.java* randomly generates test cases. The seed of the random number generator is set to ensure the same test cases whenever the program is executed. Note that the purpose of the tests is for you to avoid major mistakes. **Passing all given tests does not imply that your algorithm is correct, especially that it has the expected runtime.**

Submission

For your submission, upload the file *Lab6.java* with your implementation to canvas.

This is an individual assignment. Therefore, a submission is required from each student.

Deadline: Sunday, June 7, 11:59 p.m.