CS 2302 - Data Structures
Fall 2018
Project 6 - Option A

Overview

Implement the following graph algorithms:

- Kruskal's algorithm
- Topological sort

Feel free to use the instructor's implementation of the Disjoint Set Forest and Graph data structures. Do at least one of the following to test your implementations:

- Unit Tests
- Creating a separate file where you call your implementations using hard-coded graphs

Extra Credit (5 points total):

Solve the following LeetCode Problems

1. **https://leetcode.com/problems/minimum-height-trees/description/**
2. **https://leetcode.com/problems/reconstruct-itinerary/description/**

What you need to do

Part 1 - Due Tuesday, December 3, 2018

Implement the program described above and upload your code to GitHub.

Rubric

Criteria	**Proficient**	**Neutral**	**Unsatisfactory**
Correctness	The code compiles, runs, and solves the problem.	The code compiles, runs, but does not solve the problem (partial implementation).	The code does not compile/run, or little progress was made.
Space and Time complexity	Appropriate for the problem.	Can be greatly improved.	Space and time complexity not analyzed
Problem	Operations are	Operations are	Most of the logic is

Decomposition	broken down into loosely coupled, highly cohesive methods	broken down into methods, but they are not loosely coupled/highly cohesive	inside a couple of big methods
Style	Variables and methods have meaningful/appropriate names	Only a subset of the variables and methods have meaningful/appropriate names	Few or none of the variables and methods have meaningful/appropriate names
Robustness	Program handles erroneous or unexpected input gracefully	Program handles some erroneous or unexpected input gracefully	Program does not handle erroneous or unexpected input gracefully
Documentation	Non-obvious code segments are well documented	Some non-obvious code segments are documented	Few or none non-obvious segments are documented
Code Review	Useful feedback was provided to team members. Feedback received from team members was used to improve the code.	Feedback was provided to team members, but it was not very useful. Feedback received from team mates was partially used to improve the code	Little to no feedback was provided to team mates. Received feedback was not used to improve the code.
Report	Covers all required material in a concise and clear way with proper grammar and spelling.	Covers a subset of the required material in a concise and clear way with proper grammar and spelling.	Does not cover enough material and/or the material is not presented in a concise and clear way with proper grammar and spelling.