**CS 2302 - Data Structures**

**Fall 2019**

**Project 6**

**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](https://docs.python.org/3.6/library/unittest.html)
* Creating a separate file where you call your implementations using hard-coded graphs

**Extra Credit (8 points total - each problem is worth 2 points):**

**Solve the following LeetCode Problems**

1. [**https://leetcode.com/problems/course-schedule/description/**](https://leetcode.com/problems/course-schedule/description/)
2. [**https://leetcode.com/problems/course-schedule-ii/description/**](https://leetcode.com/problems/course-schedule-ii/description/)
3. [**https://leetcode.com/problems/minimum-height-trees/description/**](https://leetcode.com/problems/minimum-height-trees/description/)
4. [**https://leetcode.com/problems/reconstruct-itinerary/description/**](https://leetcode.com/problems/reconstruct-itinerary/description/)

**What you need to do**

**Part 1 - Due Monday, December 2, 2019**

Final due date (everything finished - code).

**Part 2- Due Thursday, December 5, 2019**

Final due date (report)

**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 Decomposition** | Operations are broken down into loosely coupled, highly cohesive methods | Operations are broken down into methods, but they are not loosely coupled/highly cohesive | Most of the logic is 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 |
| **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. |