Homework 8

Sam Daitzman // DSA Spring 2020

Exercise 1: Single Minimum Spanning Tree

Prove that if a graph $G = (V, E)$ has unique edge weights (i.e. $we1 \neq we2$ for any two edges $e1, e2 \in E$), then there is a single minimum spanning tree. In other words, there is only one optimal solution. **Hint**: Use a proof by contradiction and suppose that there are two optimal spanning trees T_1* and T_2* .

Exercise 2: Breadth-First Search

Suppose that you are given an unweighted graph $G = (V, E)$ and a node $i \in V$. One way to find the shortest path from i to all other nodes $j \in V$ is to run breadth-first search from i. In particular, for every node v added to the queue when processing u in BFS, we mark u as v's parent. This induces a tree of shortest paths from i, as shown below in red for $i = 1$. In this case, the shortest path from 1 to 5 is 1-3-5.

Exercise 2a: BFS Proof by Contradiction

Use a proof by contradiction to show that BFS keeps track of the shortest path from i to all other nodes. **Hint**: Consider the closest node v to the source i whose BFS path is not a shortest path.

Exercise 2b: Weighted to Unweighted

Given a weighted graph $G = (V, E)$ in which each edge weight $w_e \geq 0$ is an integer, explain how to convert G into an unweighted graph $G' = (V', E')$ such that finding the shortest path from i to j in G' gives you the shortest path in G. How many vertices and edges does this new graph have?

Exercise 2c: BFS Shortest Path

Using your construction above, explain how to find the shortest path from i to all other nodes j in a weighted graph $G = (V, E)$ using

BFS. What is the runtime of your overall algorithm? How does it compare to the runtime of Dijkstra's algorithm?