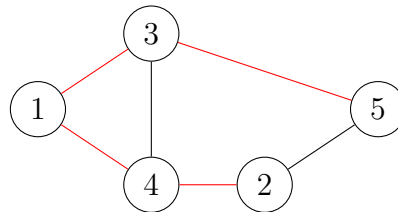


Homework 8: Spanning Trees and Shortest Paths

Due: Friday, April 3 at 11:59 pm on Canvas

Concepts: minimum spanning trees, breadth-first search, shortest paths

- (6 points) Prove that if a graph $G = (V, E)$ has unique edge weights (i.e. $w_{e_1} \neq w_{e_2}$ for any two edges $e_1, e_2 \in E$), 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^* .
- (15 points) 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.



- (6 points) 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.
- (6 points) 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?
- (3 points) 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?