

COMPSCI330 Design and Analysis of Algorithms

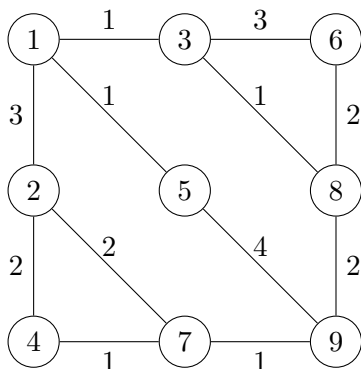
Assignment 5

Due Date: Wednesday, March 18, 2020

Guidelines

- **Describing Algorithms** If you are asked to provide an algorithm, you should clearly define each step of the procedure, establish its correctness, and then analyze its overall running time. There is no need to write pseudo-code; an unambiguous description of your algorithm in plain text will suffice. If the running time of your algorithm is worse than the suggested running time, you might receive partial credits.
- **Typesetting and Submission** Please submit the problems to GradeScope. You will be asked to **label your solution for individual problems**. Failing to label your solution can cost you 5% of the total points (3 points out of 60 for this homework).
- \LaTeX is preferred, but answers typed with other software and converted to pdf is also accepted. Please make sure you submit to the correct problem, and your file can be opened by standard pdf reader. **Handwritten answers or pdf files that cannot be opened will not be graded.**
- **Timing** Please start early. The problems are difficult and they can take hours to solve. The time you spend on finding the proof can be much longer than the time to write. If you need more time for your homework please use this form and submit a STINF.
- **Collaboration Policy** Please check this page for the collaboration policy. You are **not** allowed to discuss homework problems in groups of more than 3 students. **Failure to adhere to these guidelines will be promptly reported to the relevant authority without exception.**

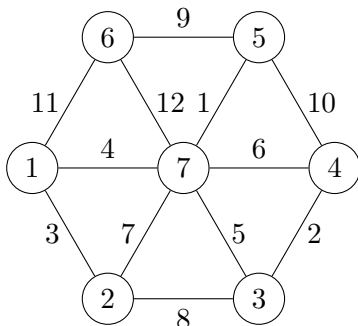
Problem 1 (Dijkstra). (10 points) We are going to perform Dijkstra's algorithm on the following graph



This is an undirected graph, the numbers close to the edges are the edge weights. The numbers on the vertices are the indices of the vertices. We will perform Dijkstra using vertex 1 as the starting vertex.

- (a) (5 points) Give the ordering in which the vertices are visited in Dijkstra's algorithm. (When there are multiple options, the algorithm chooses the vertex with smaller index.)
- (b) (5 points) After vertex 7 is visited by Dijkstra's algorithm, what are the distance values for all the vertices (if a vertex cannot be reached yet its distance value is $+\infty$)?

Problem 2 (Prim and Kruskal). (20 points) We will run Prim's algorithm and Kruskal's algorithm for the following graph:



In this graph, the numbers in circles are indices for vertices, and the numbers next to edges are their lengths.

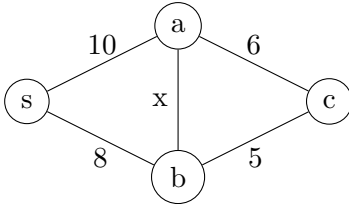
- (a) (5 points) Suppose we run Prim's algorithm with starting vertex 7, list the edges added by Prim's algorithm in the order that they are added by the algorithm.
- (b) (5 points) Now we will run Kruskal's algorithm, list the edges added by Kruskal's algorithm in the order that they are added by the algorithm.

Problem 3 (Graph Examples). (20 points)

- (a) (10 points) We talked about cycle property for minimum spanning trees, which says the longest edge of a cycle can be safely removed without impacting the length of the minimum spanning trees. The contrary of this would say "the shortest edge of a cycle always belong to some minimum spanning tree", however this is not true. Construct a graph with 5 vertices and a cycle that contains

all 5 vertices, such that none of the edges of this cycle belong to any minimum spanning tree in the graph. (You just need to list the edges and their weights, specify the cycle, and give one minimum spanning tree for your graph.)

(b) (10 points) Consider the following undirected graph, where the edge (a, b) has weight x , which could be a negative number. Suppose we run Dijkstra's algorithm starting from vertex s , for what range of x can the algorithm correctly find the shortest paths from s to every other vertex? For what range of x does the graph have a negative cycle? (For both questions, specify the range using inequalities, like $x > 10$ or $-100 < x \leq 100$. No need for justification.)



Problem 4 (Electric Network). (20 points) D energy is a electric company that is serving the triangle area. The company has a few power plants, and it wants to connect all the houses in the area to one of the generators. The potential connections are given as a graph with n vertices and m edges. The first k out of these n vertices are power plants, the remaining $n - k$ vertices are houses that needs to be connected. Every edge represents a possible link to build, and the cost is given as the weight of the edge (you can assume the weights are nonnegative). The goal is to select a subset of edges such that every house is connected to at least one of the power plants (a house can be connected indirectly via a path that passes through other houses/power plants). You are hired by D energy to design an algorithm for solving this problem. You can assume that it is always possible to connect every house to at least one of the generators. Design the algorithm, analyze its running time and prove its correctness. Your algorithm should run in $O(m + n \log n)$ time.