

Analysis of Algorithms
CS 376, Fall 2020

Problem Set 6

Due **BY THE BEGINNING OF CLASS** Monday, November 16

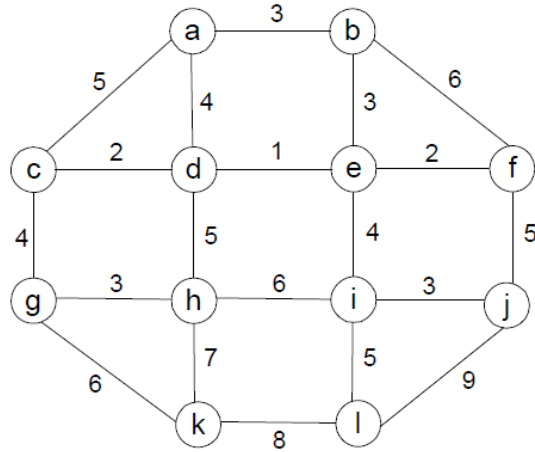
Please submit this assignment in a file named `CS376_PS6-<YourName>.pdf`.

- From CLRS, please read Ch. 22, Ch. 25 (up to pg. 699), Ch. 16 (up to and including Section 16.2), and Ch. 23.
- For this assignment and future CS376 assignments, there may be deductions for excessively inelegant solutions (e.g., algos, proofs) to exercises. Answers do not need to be optimally elegant for full credit, but when an excessively inelegant answer misses some central concepts that would result in a cleaner solution, there is room for improvement to the answer, which may entail a deduction. Please feel free to ask me questions about this and how it might apply to your specific HW solutions!
- *A general note for CS376:* When writing up your homework, please write neatly and **explain your answers clearly**, giving all details needed to make your answers easy to understand. Clear communication *is* the point, on every assignment.

In general in CS376, unless explicitly specified otherwise, answers should be accompanied by explanations; as part of this, in general, when you present an algorithm, give an accompanying correctness argument (i.e., proof). Answers without explanations may not receive full credit. Please feel free to ask me any questions about explanations that might come up!

Exercises

1. CLRS 22.1-6 (pg. 593).
2. CLRS 23.1-4 (pg. 629). (As usual, explain your answer thoroughly.)
3. Apply Kruskal's algorithm to the following graph, to find a minimum spanning tree. (Break ties, where applicable, by alphabetical ordering on the endpoints of edges.) For your answer, list all of the edges added to a Minimum Spanning Tree by Kruskal's algorithm. List them in the order in which each edge is added and, for each added edge, give a very brief (1 sentence is fine!) explanation of the reasons behind choosing that edge. You are welcome to include a picture of the MST as part of your answer, but you do not need to; listing the MST's edges in order is sufficient.



4. **Bridge Crossing Revisited!** Consider the generalization of the bridge crossing exercise earlier in the semester (Small Assignment 2, Lecture of 090220) in which there are $n > 1$ people whose bridge crossing times are $t_1, t_2, \dots, t_n$. All the other conditions of the problem are the same as before: at most two people at a time can cross the bridge (and they move with the speed of the slower of the two) and they must carry with them the only flashlight the group has.
- Design a greedy algorithm for getting the entire group across the bridge, with the goal of minimizing the total time for the group to cross. Give a very brief explanation of what makes it a greedy algorithm.
 - Does the greedy algorithm yield an optimal solution (i.e., a minimal crossing time) for every instance of this problem? If so, explain how we know it's always optimal; if not, give a concrete counterexample **with the smallest number of people** for which it is not optimal (and explain how you know that's the smallest number of people for which it's not optimal).