MA/CSSE 473 HW 13 textbook problems and hints

Problem #1 (10) 9.1.3 (Greedy job scheduling)

3. Consider the problem of scheduling n jobs of known durations $t_1, ..., t_n$ for execution by a single processor. The jobs can be executed in any order, one job at a time. You want to find a schedule that minimizes the total time spent by all the jobs in the system. (The time spent by one job in the system is the sum of the time spent by this job in waiting plus the time spent on its execution.)

Design a greedy algorithm for this problem. ▷ Does the greedy algorithm always yield an optimal solution?

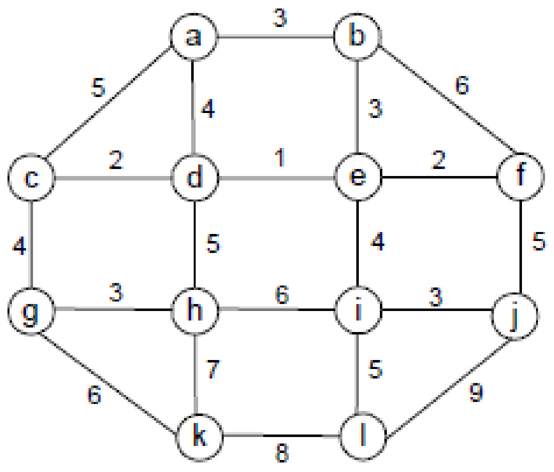
Author's hint:

3. Considering the case of two jobs might help. Of course, after forming a hypothesis, you will have to either prove the algorithm's optimality for an arbitrary input or find a specific counterexample showing that it is not the case.

Problem #2 (6) 9.1.9b [9.1.7]b (Prim example)

(Prim example) Start with node a. Whenever you have a choice because edge weights are equal, choose the vertex that is closest to the beginning of the alphabet. Then everyone should get the same answer, making it easier for us to check your work.

b. Apply Prim's algorithm to the following graph. Include in the priority queue only the fringe vertices (the vertices not in the current tree which are adjacent to at least one tree vertex).



Author's hint:

b. After the next fringe vertex is added to the tree, add all the unseen vertices adjacent to it to the priority queue of fringe vertices.

Problem #3 (5) 9.1.10 [9.1.8] (Prim prior connectivity check?)

8. The notion of a minimum spanning tree is applicable to a connected weighted graph. Do we have to check a graph's connectivity before applying Prim's algorithm or can the algorithm do it by itself?

Author's hint:

8. Applying Prim's algorithm to a weighted graph that is not connected should help in answering this question.

Problem #4 (10)) 9.1.15 [9.1.11] (change value of an item in a min-heap)

11. Outline an efficient algorithm for changing an element's value in a min-heap. What is the time efficiency of your algorithm?

Author's hint

11. Consider two cases: the key's value was decreased (this is the case needed for Prim's algorithm) and the key's value was increased.

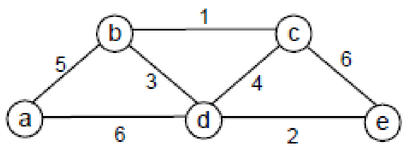
:

Problem #5 (6) 9.2.1b (Krushkal example)

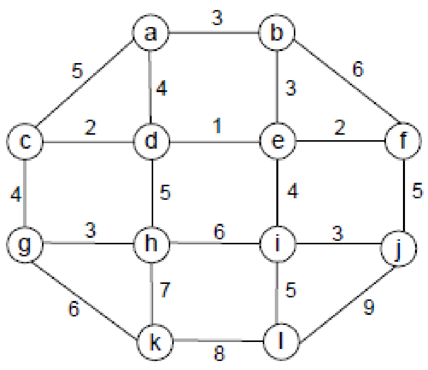
(Krushkal example) Whenever you have a choice because edge weights are equal, choose the edge whose vertices are closest to the beginning of the alphabet. Then everyone should get the same answer, making it easier for us to check your work.

1. Apply Kruskal's algorithm to find a minimum spanning tree of the following graphs.

a.



b.



Author's hint:

1. Trace the algorithm for the given graphs the same way it is done for another input in the section.

Problem #6 (8) 9.2.2 (Kruskal TF questions) **Briefly explain your answers.**

2. Indicate whether the following statements are true or false:

   a. If e is a minimum-weight edge in a connected weighted graph, it must be among edges of at least one minimum spanning tree of the graph.

   b. If e is a minimum-weight edge in a connected weighted graph, it must be among edges of each minimum spanning tree of the graph.

   c. If edge weights of a connected weighted graph are all distinct, the graph must have exactly one minimum spanning tree.

   d. If edge weights of a connected weighted graph are not all distinct, the graph must have more than one minimum spanning tree.

Author's hint:

2. Two of the four assertions are true, the other two are false.

Problem #7 (8) 9.4.1 Huffman Code construction

(a) 4 points. When there is a choice due to a tie, place the one that appears first in the problem statement's character list "on the left" in the tree. (b) 2 points. (c) 2 points.

1. a. Construct a Huffman code for the following data:

character	A	B	C	D	_
probability	0.4	0.1	0.2	0.15	0.15

   b. Encode the text `ABACABAD` using the code of question a.

   c. Decode the text whose encoding is 100010111001010 in the code of question a.

Author's hint:

1. See the example given in the section.

Problem #8 (12) 9.4.3 Huffman TF **Explain your answers**

3. Indicate whether each of the following properties are true for every Huffman code.

   a. The codewords of the two least frequent characters have the same length.

   b. The codeword's length of a more frequent character is always smaller than or equal to the codeword's length of a less frequent one.

Author's hint:

3. You may base your answers on the way Huffman's algorithm works or on the fact that Huffman codes are known to be optimal prefix codes.