CPSC-320 Intermediate Algorithm Design and Analysis Winter 2000

Homework # 3

Due Monday, June 9th at 6pm.

NAME:_____

Signature:_____

STD. NUM: _____

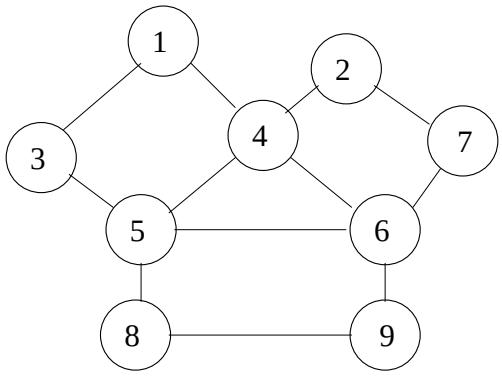
General guidelines for homeworks:

Before starting on this homework, review the homework guidelines provided on the first day of class (also on the web under "Course Description"). Remember that it is encouraged to discuss the problems with others in the class, but all write-ups are to be done on your own. **Homework grades will be based not only on getting the "correct answer," but also on good writing style and clear presentation of your solution.** It is your responsibility to make sure that the graders can easily follow your line of reasoning.

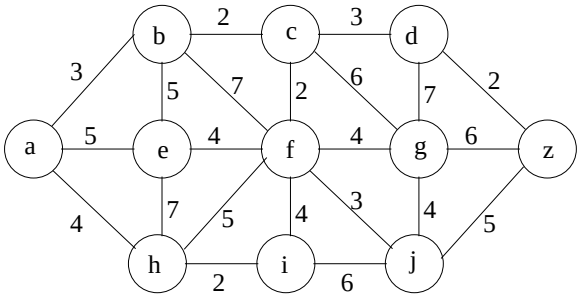
Try every problem. Even if you can't solve the problem, you will receive partial credit for explaining why you got stuck on a promising line of attack. More importantly, you will get valuable feedback that will help you learn the material.

Please acknowledge the people with whom you discussed the problems and what sources you used to help you solve the problem (e.g. books from the library). This won't affect your grade but is important as academic honesty.

1. Prove that the following graph is not bipartite. Hint: argue by contradiction.



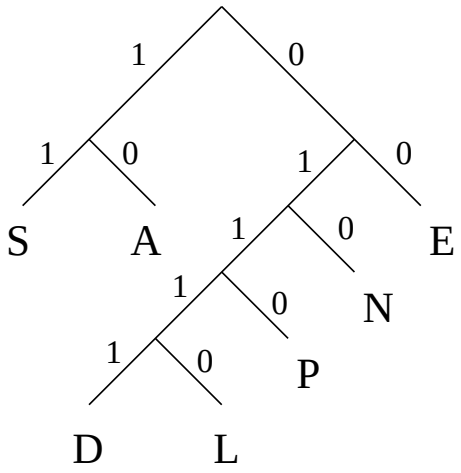
2. Find the length of a shortest and path and a shortest path between (i) a and f and (ii) a and z.



3. Decode the following bit strings using the Huffman code given in the figure.

   (a) 011000010

   (b) 01110100110

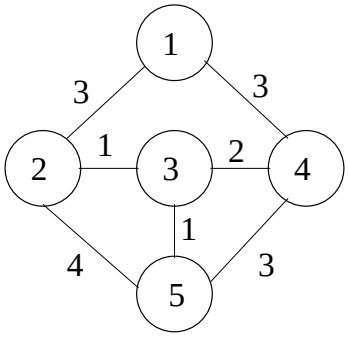


4. Construct an optimal Huffman code tree for the set of letters in the table.

Letter	**Frequency**
a	5
b	6
c	6
d	11
e	20

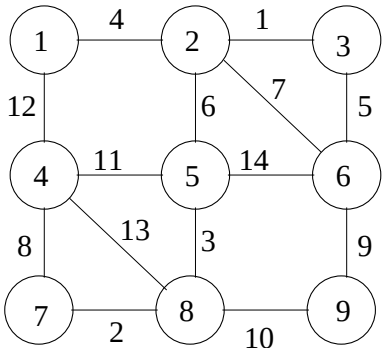
5. Starting at node 1,

   (a) Describe how Prim's and Kruskal's algorithms compute the MSP for the following graph.



(b) How can one modify Kruskal's algorithm to obtain a algorithm that clusters nodes into k clusters. That is if nodes represent data and the edge weights represent dissimilarity measures, the algorithm should group similar items together automatically. Do it with $k - 3$ for the graph of part (a).

6. Starting at node 1, describe how Prim's and Kruskal's algorithms compute the MSP for the following graph.



7. Design an efficient algorithm to find a spanning tree for a connected, weighted, undirected graph $G = (V, E)$ such that the weight of the maximum weight edge in the spanning tree is minimized. Hint: is the tree that minimises this measure the MSP? Can you argue this by contradiction.