CSC 440 - Assignment 9

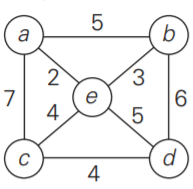
# Instructions

First, make a copy of this document (click on File → Make a copy). You will be editing your own copy of the assignment.

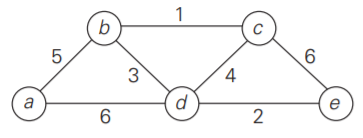
When you finish the assignment, you must download your copy and submit the downloaded file into Moodle (click on File → Download → File type; please use .doc, .docx, .pdf, or .rtf formats for your submissions.)

# Problems

1. (15 pts) Apply Prim’s algorithm to the following graph. Include in the priority queue all the vertices not already in the tree for each step, as well as their labels (nearest neighbor and distance to that neighbor).



1. (15 pts) Apply Kruskal’s algorithm to find a minimum spanning tree of the following graph. List one step for each edge examined. If the edge is not included, simply indicate that on the edge’s step.

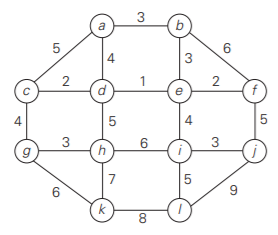


1. (15 pts) Indicate whether the following statements are true or false, and give a brief explanation of why. These explanations need not be overly technical, as long as they make the point clearly.
   1. If *e* is a minimum-weight edge in a connected weighted graph, it must be among edges of at least one minimum spanning tree of the graph.
   2. 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.
   3. If edge weights of a connected weighted graph are all distinct, the graph must have exactly one minimum spanning tree.
   4. If edge weights of a connected weighted graph are not all distinct, the graph must have more than one minimum spanning tree
2. (15 pts) Solve the following instance of the single-source shortest-paths problem with vertex *a* as the source, using Dijkstra’s algorithm. Note that this graph is not as small as the others, so for this problem it is sufficient to show the final calculated distances. Since we are starting with vertex *a*, your answer could look like:

a → b = *some number*

a → c = *some number*

...



1. (**Extra Credit** 5 pts) Dijkstra’s algorithm is not the only shortest-path algorithm in use today. Find another shortest-path algorithm **THAT IS NOT BFS** (single source - single destination, single source - all destinations, all sources - one destination, or all sources - all destinations), and write a paragraph describing how it works, as well as how it differs from Dijkstra’s algorithm.