In lab exercises

1. Given a set $S = \{x_1, x_2, \ldots, x_n\}$ of n points on the real number line, give an algorithm for computing the smallest set of unit-length closed intervals that contain all the given points. Show correctness and run-time.

2. Let A and B be two sets containing n positive integers. Order sets A and B by an order of your choosing and let a_i and b_i be the ith elements in sorted order from A and B. Give an ordering that maximizes $\prod_{i=1}^{n} {a_i}^{b_i}$.

A graph $G = (V, E)$ can be *weighted* by assigning a weight $w(e)$ to each edge e in the graph. For the problems in today's lab, assume the weights of all edges are unique and positive. Dijkstra's Algorithm shown below in algorithm 1, can be used to solve the single source shortest path problem of given a weighted graph G and a source vertex s find the paths of lowest total weight from s to all other vertices in G reachable from s.

Algorithm 1 Dijkstra's Algorithm(G, w, s):

Given graph $G = (V, E)$ with edge weights w, and starting vertex s
Let S be a set of explored vertices
For each $u \in S$, store a distance $d(u)$
$S = \{s\}, d(s) = 0$
while $S \neq V$ **do**
  select vertex $v \notin S$ with at least one edge from S for which
    $d'(v) = \min_{e=(u,v):u \in S}(d(u) + w(e))$ is smallest
  add v to S and set $d(v) = d'(v)$
end while

---

3. Trace the execution of Dijkstra's algorithm on the sample graph shown in lab.

4. Are the shortest paths unique if the edge weights are unique? What if the edge weights are not unique? What if the edge weights are uniformly constant?

5. How is Dijkstra's algorithm greedy? Outline how you could argue that it produces an optimal result.

In the minimum spanning tree problem, we are given a connected weighted graph and we wish to choose a subset $T \subseteq E$ such that $G' = (V, T)$ is connected, but whose total edge weight $\sum_{e \in T} w(e)$ is minimized. Kruskal's algorithm iteratively builds such a subset T incrementally by repeatedly adding edges from E in increasing order of weight, provided each newly added edge does not form a cycle.

6. Trace the execution of Kruskal's algorithm on the sample graph shown in lab.

7. The subset T must be a tree. Why?

The *cut* property states: Let S be any subset of vertices that is neither empty nor equal to V. Let e be the minimum cost edge with one vertex in S and one vertex in $V - S$. Every MST contains e. Consider the following "proof" of the cut property.

Let T be a spanning tree not containing e. There must be some edge $f \in T$ with one vertex in S and the other in $V - S$, otherwise T is not a spanning tree. Since e is the cheapest edge spanning the cut, $w(e) < w(f)$ and $T - \{f\} \cup \{e\}$ is a cheaper spanning tree, contradicting the claim that T is a MST.

8. Debug this proof.