There is a 10% penalty for turning in this homework before 2:00 PM, Monday Dec. 3. No later submissions will be allowed.

Please respect the following guidelines for writing pseudocode:

1. C instructions are fine. But do not write object-oriented additions. Do not declare or use any class. Declare only procedures (if necessary) and explain in words what each procedure does, and what is the use of each parameter.

2. One instruction per line

3. Match the brackets with a horizontal line

4. Number your lines

5. Write down if your array is indexed 0\ldots n - 1 or 1\ldots n.

6. If you use results not covered in class, present full details of pseudocode, analysis, and proofs (to the point only results from the class are used).

**Problem 1** Describe an efficient algorithm that, given an undirected graph $G$, determines a spanning tree of $G$ whose largest edge weight is minimum over all spanning trees of $G$.

Prove your algorithm is correct.

**Problem 2** 24.3-1 on page 600 in textbook.

**Problem 3** Give an example of a weighted directed graph $\vec{G}$ with negative-weight edges, but no negative-weight cycle, such that Dijkstra’s algorithm incorrectly computes the shortest-path distances from some start vertex $v$. Use the algorithm version from the handout.

A four-vertex example is possible. Draw the graph, mention the start vertex, show the result of Dijkstra’s algorithm, and point out for which vertex the result is incorrect.

**Problem 4** Let $G$ be a complete digraph with non-negative arc weights. Let the capacity of a path be the minimum arc weight along it, and let the capacity of a pair of nodes $(u, v)$ be the maximum capacity of a path from $u$ to $v$. Find a Dijkstra-like algorithm to find, for all $v \neq s$, the capacity of $(s, v)$. (Node $s$ is a fixed source.)

Present the pseudocode, analyze the running time, and prove correctness.