CS 535 Design and Analysis of Algorithms **Fall Semester, 2021**

Homework 1 version 1.1

Assigned: August 30 *Due: September 15*

Remote students: please use blackboard to submit your solutions.
Notes for pseudocode usage:

1. C/Java 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. Feel free to use as procedures algorithms from the textbook; indicate page and edition.
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 $1 \ldots n$.

Problem 1 Show that there is no comparison sort whose running time is linear for at least half of the $n!$ inputs of length n. What about a fraction of $1/n$ of the inputs of length n? What about a fraction $1/(2^n)$?

Problem 2 Our SELECTION algorithms uses partitioning into groups of 5, computing the median in each group, followed by two recursive calls (with some work between them). Analyze the running time for two variants of this algorithm, where the groups are of size 3, and of size 7 respectively.

Problem 3 Assume that in a weighted $G = (V, E)$ with weights w on the edges, for every cut (as defined in blue_rule.pdf), among the edges crossing the cut only one has minimum weight. Let T be a minimum spanning tree in G. Prove that no other minimum spanning tree exists.

Give a counterexample to the converse statement; that is give an example of a weighted graph G with minimum spanning tree T and cut $(S, \bar{S})$ such that no other minimum spanning tree exists and there are two edges of minimum weight crossing $(S, \bar{S})$.

Problem 4 Given an edge-weighted undirected graph $G = (V, E, w)$, and a spanning tree T in G, define the *width* of T to be the weight of its minimum-weight edge.

Describe an efficient algorithm that, given G, finds a spanning tree T of maximum width. Present pseudocode, analyize the running time and prove correctness.

Problem 5 Let $G = (V, E, w)$ be a weighted undirected graph with edge costs w_e, for $e \in E$, and let T be a minimum spanning tree in G, given in the adjacency list representation. Add an edge e' to E of cost $w_{e'}$, obtaining a new weighted graph $G' = (V, E \cup \{e'\}, w)$. Give an efficient algorithm to compute the minimum spanning tree in G'. Present pseudocode, analyize the running time and prove correctness. Give a $O(|V|)$-algorithm for full credit.