FOUNDATIONS OF COMPUTING

Instructions: This exam consists of 7 problems, each of equal weight. Please write answers on the exam. If you would like to attach additional pages, include examinee number on each page. To keep answers concise, it is permissible to refer to algorithms in books and cite page numbers.
1. a. Show that each of the following decision problems is in the class $NP$.

   (i). Given a weighted graph $G$ with integer edge weights, an integer $k$, and vertices $u$ and $v$ of $G$, does $G$ have a simple path from $u$ to $v$ of weight at most $k$?

   (ii). Given a graph $G$, and an integer $k$, does $G$ have a simple path of length (number of edges) at least $k$?

   (iii). Given a weighted graph $G$ with integer edge weights and an integer $k$, does $G$ have a spanning tree of weight at most $k$?

   (iv). Given a weighted graph $G$ with integer edge weights and an integer $k$, does $G$ have a spanning path (simple path that includes all vertices) of weight at most $k$?

b. If $P \neq NP$, which of the decision problems above cannot be in the class $P$?
2. For each of the following descriptions involving languages $L_1$ and $L_2$, either give specific languages $L_1$ and $L_2$ that satisfy the description or state a theorem (or theorems) that tells you such examples are impossible.

   a. $L_1$ and $L_2$ are context-free but not regular and $L_1 \cap L_2$ is regular.

   b. $L_1$ and $L_2$ are context-free and $L_1 \cap L_2$ is not context-free.

   c. $L_1$ and $L_2$ are context-free but not regular and $L_1 \cup L_2$ is regular.

   d. $L_1$ and $L_2$ are context-free and $L_1 \cup L_2$ is not context-free.

   e. $L_1$ is context-free but not regular, $L_2$ is regular, and $L_1 \cap L_2$ is context-free but not regular.
3. You are given an array $T[1..n]$ of $n$ records of students in some course, sorted by their grades in nondecreasing order. For students with the same grade, the records are arranged in alphabetical order by their last names. Grades are floating point numbers. In answering the questions below, if it is convenient you may make assumptions about $n$ such as “$n$ is a power of 2” or “$n$ is one less than a power of 2.”

   a. Design an algorithm which, given $g$ and $t$, returns TRUE if $t$ or more students in the array have the grade $g$ and FALSE otherwise. The goal is to minimize the total number of comparisons between keys of records in the array.

   b. Do an exact worst-case analysis (not asymptotic) of the total number of key comparisons performed by your algorithm in part (a).
4. Let \( L_n = \{ a^i b^i \mid 0 \leq i \leq n \} \).
   
a. What is \( s(n) \), the minimum number of states for any deterministic finite automaton for \( L_n \)?
   
b. Prove that \( s(n) \geq \) your answer to part (a).
   
c. Prove that \( s(n) \leq \) your answer to part (a).
5. Describe an algorithm that, given \( n \) integers in the range \( 1, \ldots, k \), preprocesses its input and then answers any query about how many of the \( n \) integers fall into an interval \( [a, b] \) in \( O(1) \) time. Your algorithm should use \( O(n + k) \) preprocessing time.
6.

(a) Some binary search tree schemes (such as AVL trees and red-black trees) guarantee worst-case running time of $O(\log n)$ for insertions, deletions and lookups in a tree of $n$ nodes, while other schemes (such as Splay trees) guarantee amortized logarithmic cost per operation. Explain clearly what this means. Compare the performance of these schemes to the worst case running times for insertion, deletion, and lookup in ordinary binary search trees.

(b) Draw a complete binary search tree that contains one copy of each of the following keys and call the resulting tree $T$:

\[
B, \ D, \ F, \ G, \ I, \ J, \ L, \ N,
\]
\[
P, \ R, \ S, \ U, \ V, \ W, \ Z.
\]

(c) Do ONE of the following parts (i), (ii), (iii). Circle the one you choose:

(i) Consider the tree $T$ drawn in part (b) to be a splay tree. Show and explain all the steps involved in inserting the key C into $T$.

(ii) Consider the tree $T$ drawn in part (b) to be a red-black tree. Assume that all the leaves are red, their parents are black, their grandparents are red, and the root is black. Show and explain all the steps involved in inserting the key H into $T$.

(iii) Consider the tree $T$ drawn in part (b) to be an AVL tree. Show and explain all the steps involved in first deleting the key G from $T$ and then inserting the key C into the resulting tree.
7. A source in a directed graph $G$ is a vertex of $G$ from which all vertices of $G$ can be reached by a directed path. Design an algorithm that, given the adjacency list representation of a directed graph $G = (V, E)$, returns a source of $G$, if one exists and otherwise returns FALSE. The running time of your algorithm must be asymptotically optimal. Partial credit will be given for algorithms which are correct, but not optimal.