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 your examinee number on each page. To keep answers concise, it is permissible to refer to algorithms in books and cite page numbers.
1. Prove that $L = \{xa^nREV(x) \mid x \in \{a,b\}^* \land |x| \leq n \}$ is not a context-free language ($REV(x)$ is the reversal of string $x$).
2. (a) Design a deterministic finite automaton (DFA) for the language $L$, where every string in $L$ is in $\{a, b\}^*$ and has at least two occurrences of the substring $aa$. Note: the string $aaa \in L$ because there’s an occurrence starting at position 1 and another at position 2.

(b) Prove that every DFA for $L$ in part (a) has at least 5 states.
3. Which of the following decision problems are decidable? Give a brief justification for each answer.

(a) Given two regular expressions $E_1$ and $E_2$ over $\Sigma$, is $L(E_1) \cup L(E_2) = \Sigma^*$?

(b) Given two context-free grammars $G_1$ and $G_2$ with terminal alphabet $\Sigma$, is $L(G_1) \cup L(G_2) = \Sigma^*$?

(c) Given a Turing machine $T$ with input alphabet $\Sigma$, is $L(T) = \Sigma^*$?

(d) Given a context-free grammar $G$ and an integer $k$, does $L(G)$ contain a string whose length is a multiple of $k$?
4. Let $G$ be a directed graph with $n$ vertices and $m$ edges. Assume each edge has a cost associated with it and that the cost is an integer in the range $1, \ldots, k$. We are given as input an adjacency list representation of $G$. Our task is to rearrange the adjacency list for each vertex so that it is in nondecreasing order of cost.

(a) What is the worst-case total asymptotic time as a function of $m$, $n$, and $k$ if we do heapsort on each adjacency list separately?

(b) Suppose we do bucket sort separately on each list. What is the worst-case total asymptotic time as a function of $m$, $n$, and $k$?

(c) Briefly describe an algorithm that sorts all the lists in total time $O(m + n + k)$. 
5. Suppose you are given a list of \( n \) ordered pairs of integers \((x_1, y_1), \ldots, (x_n, y_n)\), where each pair \((x_i, y_i)\) represents the closed interval \([x_i, y_i]\) on the real line (assume \( x_i \leq y_i \)).

(a) Describe an efficient algorithm to find the union of all the intervals and output that union as a list of disjoint intervals in increasing order of left endpoints. For example, if the input is \((6, 8)(1, 3)(5, 9)(7, 10)\), the output should be \((1, 3)(5, 10)\).

(b) Analyze the asymptotic time complexity of your algorithm.

(c) Argue that, under a natural set of assumptions, your algorithm’s time complexity is no worse than that of any other algorithm solving this problem.
6. Let $G$ be an unweighted, undirected graph, let $T$ be a spanning tree of $G$, and let $r$ be the root of $T$. Then define $d(T) = \sum_{v \in T} d(r, v)$, where $d(r, v)$ is the number of edges on the path from $r$ to $v$ in $T$. If $d(T) \geq d(T')$ for every tree $T'$ rooted at $r$ then $T$ is called a maximum-depth tree.

(a) Let $G$ be the graph pictured on the right. Find a maximum-depth tree rooted at vertex 1 for $G$.

(b) Give an example of a graph $G$ and a depth-first search (DFS) tree $T$ for $G$ such that $T$ is not a maximum-depth tree.

(c) Prove that every maximum-depth tree of an arbitrary graph $G$ is a DFS tree. Hint: Show that a maximum-depth tree has no forward or cross edges. You may assume that the converse of Theorem 23.9 in Cormen, Leiserson, and Rivest is true. The theorem says that every edge of a DFS tree is either a tree edge or a back edge.
7. Consider four decision problems \( A, B, C, \) and \( D \), where \( A \) is known to be in class \( \mathcal{P} \), \( B \) is known to be in class \( \mathcal{NP} \), \( C \) is known to be \( \mathcal{NP} \)-complete, and \( D \) is known \textit{not} to be in \( \mathcal{P} \).

(a) Draw a table with rows corresponding to \( A, B, C, \) and \( D \), and columns corresponding to the classes \( \mathcal{P}, \mathcal{NP}, \) and \( \mathcal{NP} \)-complete (NPC). Each table entry should be \textbf{yes}, \textbf{no}, or \textit{?}, depending on whether the given problem is definitely known to be in the class, definitely known not to be in the class, or its status with respect to the class is unknown (according to the information given above).

For each of the statements below, write \textbf{impossible} next to it if it's impossible. Otherwise make a list of the \textit{?} items from your table of part (a) that are changed to a \textbf{yes} or \textbf{no} by the statement. For example, \((B, \mathcal{P}, \textbf{yes})\) would indicate that the statement implies \( B \in \mathcal{P} \) (assuming this was not already known). Write \textbf{none} if there are no changes.

\( X \leq_{\mathcal{P}} Y \) means that \( X \) reduces to \( Y \) in polynomial time. \( \mathcal{NP} \)-completeness is defined with respect to \( \leq_{\mathcal{P}} \) as in the Cormen, Leiserson, and Rivest text.

(b) \( D \leq_{\mathcal{P}} B \)

(c) \( D \leq_{\mathcal{P}} A \)

(d) \( C \leq_{\mathcal{P}} D \)

(e) \( A \leq_{\mathcal{P}} B \)

(f) \( C \leq_{\mathcal{P}} A \)

(g) \( B \leq_{\mathcal{P}} A \)