Examination "Breadth"

Theory

September 23, 2002 - 3 hours

SOLVE EACH PROBLEM ON A SEPARATE PAGE. WRITE ONLY ON ONE SIDE OF A PAGE. PUT DOWN YOUR CODE ON EACH PAGE.

1. The concept of *completeness* is essential in time or space bounded computations.
   (a) Define the notion of EXP-completeness for the class EXP defined as $\cup_k DTIME(2^{n^k})$.
   (b) What are the differences between EXP-completeness and NP-completeness? What are the similarities?

2. Consider a language $L = \{w \in \{a,b\}^* :$ the number of a's in w is twice the number of b's$\}$.
   (a) Is L regular? Prove your answer.
   (b) Is L context-free? Prove your answer.

3. Let T_1 and T_2 be two binary search trees with n elements each. Describe an algorithm, provide pseudocode and analyze space and time complexity for the problem of outputting in increasing order a sequence consisting of all the elements stored in T_1 and T_2. For full credit the running time of your algorithm should be linear in the size of the input. You may assume that all the elements in T_1 and T_2 are distinct.

4. (a) Describe a binary heap representation for priority queues.
   (b) Explain how DELETE_MAX() operation works and what its cost is. In your description you can use, without defining, standard heap operations such as heapify-up or heapify-down.
   (c) Show a binary heap for the set of letters {F, O, U, N, D, A, T, I, L, E, X, M} assuming that we consider priority queues with MAX element in the root. You do not need to construct this heap with any standard operations such as the bottom-up or top-down constructions - just show a correct heap. The standard alphabetical order of the characters in the English alphabet is assumed.
   (d) Show the same heap after DELETE_MAX(). Use the standard algorithm that utilizes HEAPIFY (called also HEAPIFY_DOWN, PERCOLATE_DOWN) function.

5. (a) Give definitions for
   i. an equivalence relation on a set
   ii. the equivalence classes defined by an equivalence relation.

   (b) A partition of a set S is a collection of subsets $\{S_i : i \in I\}$ for some index set I such that $S_i \cap S_j = \emptyset$ if $i \neq j$ and $U_{i \in I} S_i = S$. Prove that the set of equivalence classes of an equivalence relation is a partition.

SELECT ONE OF PROBLEMS 6 or 7. SOLVE ONLY ONE OF THESE. WRITE ON PAGE ONE YOUR SELECTION. DO NOT INCLUDE SOLUTIONS OF MORE THAN ONE PROBLEM, ONLY THE PROBLEM LISTED ON PAGE ONE WILL COUNT.

6. A *coloring* of a graph $G = (V, E)$ is a labeling of the vertices of G so that no two adjacent vertices have the same color. If k is an integer, then we say that a graph G is *k-colorable* if it has a coloring with at most k distinct labels.

   (a) Give pseudcode for a polynomial time algorithm that determines whether a graph is 2-colorable. Analyze the algorithm's time complexity.

   (b) Prove that if every vertex in a graph has degree two, then the graph is 3-colorable.

   (c) Using the preceding part, describe an algorithm that determines whether a given graph is 3-colorable. Give pseudocode and analyze the complexity. Your algorithm should run in polynomial time when every vertex in the graph has degree two, but may run in exponential time otherwise.

7. A clause in propositional logic is *Horn* if at most one of the literals that appear in it as disjuncts is positive. A propositional theory is *Horn* if all its formulas are Horn clauses.

   A propositional theory is *consistent* if there is a truth assignment to propositional variables in the theory for which all formulas of the theory are true.

   (a) Prove that if D is a consistent propositional Horn theory and A is a set of atoms occurring in D, then

$$D \cup \{\neg a \colon a \in A,\ D \not\models a\}$$

   is consistent.

   (b) Is the above statement true for arbitrary propositional theories? Prove or disprove.