Theoretical Computer Science
Qualifying Exam
Fall 1994

Instructions. Please work any 4 of the 7 problems given below.

Problem 1. We are given a set of n nuts and n bolts, of varying sizes. For each nut, there is exactly one bolt that it fits; others are too small or too large. You are allowed to select a nut and a bolt and compare them; the result of a comparison is either (a) nut too small; (b) nut too large; or (c) perfect fit. You are *not* allowed to compare nuts to other nuts directly, or bolts to other bolts.

Describe a randomized algorithm that matches the nuts to the bolts in $O(n \log n)$ expected time.

Problem 2. Give an efficient algorithm to solve the following problem:

You are given an acyclic directed graph G, in which each edge has a nonnegative cost. There is a distinguished vertex v, and another set B of vertices (distinct from v) with the property that there is a path from v to each vertex in B. Find a minimum-cost set of edges whose removal eliminates all paths from v to B.

Note that the set of *all* edges has the requisite disconnecting property so a solution always exists.

What is the running time of your algorithm, as a function of the number of vertices and the number of edges in the graph?

Problem 3. Comment on the computational complexity of computing $n!$. Assume that inputs and outputs are in binary.

Problem 4. Although the SATISFIABILITY problem is NP-complete, for restricted classes of formulas the problem can be solved in polynomial time.

(a) Give a polynomial time algorithm for 2SAT - the satisfiability problem in which each conjunct has at most 2 literals.

(b) Give a polynomial time algorithm for Horn-SAT - the satisfiability problem in which each conjunct has at most one positive literal. (Notice that in this case a clause either has all negative literals, or it can be rewritten as an implication.)

Problem 5. Prove an $\Omega(n \log n)$ lower bound for comparison-based algorithms for the element uniqueness problem: given a set of n elements, decide if they are all distinct. Assume that when elements i and j are compared, one of three outcomes is possible: "$i < j$", "$i = j$" or "$i > j$".

Problem 6. Consider the following two definitions of PP, or "probabilistic polynomial time". First, say a probabilistic Turing machine M *accepts* an input x if and only if the probability of reaching an accepting state on x is strictly greater than $1/2$.

Then, we say L is in the class PP_1 if there is a probabilistic Turing machine that accepts L in worst case time $p(n)$ for some polynomial p. We say L is in the class PP_2 if there is a probabilistic Turing machine that accepts L in expected time $p(n)$ for some polynomial p.

(a) Show that PP_1 is contained in PSPACE.

(b) Show that any recursively enumerable language is contained in PP_2.

Problem 7. Suppose that the Turing machine model is modified to allow two new operations: insert and delete. The delete operation deletes the current tape square and the insert operation inserts a new tape square at the current location. Obviously, we can define both a one tape and k-tape version of such a machine.

(a) Formalize a definition for a k-tape Turing machine with insert and delete instructions.

(b) Show that a k-tape Turing machine can with insert and delete instructions can be simulated by a standard k-tape Turing machine. How much of a loss of efficiency is there in your simulation?