Ph.D. Qualification Examination

Algorithms (April 2008)

(1) (15%) Let k be a nonnegative constant. Show that the solution of the following recurrence is $T(n) = 3kn^{\log_2 3} - 2kn$ for n a power of 2.

$$T(n) = \begin{cases} k & n = 1 \\ 3T(n/2) + kn & n > 1 \end{cases}$$

(2) (15%) The sets A and B have n elements each given in the form of sorted arrays. Propose an $O(n)$-time algorithm to compute $A \cup B$ and $A \cap B$.

(3) (20%) Let L be an array of n distinct integers. Give an efficient algorithm to find a longest increasing subsequence of entries in L. For example, if $L = [5, 3, 7, 9, 4, 6, 2, 8]$, then a longest increasing subsequence is $[3, 4, 6, 8]$. What is the running time of your algorithm?

(4) (20%) Let $G = (V, E)$ be any connected undirected graph. A *bridge* of G is an edge e such that the graph obtained from G by removing e is disconnected. Give an efficient algorithm to find all the bridges of G. What is the running time of your algorithm?

(5) (15%) Given an array L of n numbers, the **distinct elements problem** is to check whether there are two equal numbers in L. Give an $O(1)$-time nondeterministic algorithm for this problem.

(6) (15%) Let π_2 be a problem for which there exists a deterministic algorithm that runs in time $O(2^{\sqrt{n}})$ (where n is the input size). Prove or disprove:

If π_1 is another problem such that π_1 is polynomially reducible to π_2, then π_1 can be solved in deterministic time $O(2^{\sqrt{n}})$ on any input of size n.