Ph.D. Qualification Examination

Algorithms (April 2012)

(1) (20%) Solve the following recurrences. Assume that $T(c) = 1$ for a constant c.
  (a) $T(n) = T(\frac{n}{2}) + \log n$
  (b) $T(n) = T(\sqrt{n}) + \log n$

(2) (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.

(3) (15%) Let $X[1..n]$ and $Y[1..n]$ be two arrays, each containing n numbers already in sorted array. Give an algorithm to find the median of all $2n$ elements in arrays X and Y. Analyze the time complexity of your algorithm.

(4) (15%) Suppose that there are $n = 2^k$ teams in an elimination tournament, in which there are $\frac{n}{2}$ games in the first round, with the $\frac{n}{2} = 2^{k-1}$ winners playing in the second round, and so on.
  (a) Develop a recurrence equation for the number of rounds in the tournament.
  (b) How many rounds are there in the tournament when there are 64 teams?
  (c) Solve the recurrence equation of part (a).

(5) (15%) Use the dynamic programming algorithm to find a longest common subsequence of the following sequences: (using a table to show your result)

C C G G G T T A C C A
G G A G T T C A

(6) (10%) A certain problem can be solved by an algorithm whose running time is in $O(n^{\log_2 n})$. Which of the following assertions is true?
  (a) The problem is tractable.
  (b) The problem is intractable.
  (c) None of the above.

(7) (10%) Show how to sort n integers in the range 0 to $n^3 - 1$ in $O(n)$ time.