Ph.D. Qualification Examination

Algorithms (Oct. 2004)

(1) (30%) Give asymptotically tight upper bounds for $T(n)$ in each of the following recurrences.
   (a) $T(n) = T(n-2) + 1$.
   (b) $T(n) = 2T(n/2) + n \lg^2 n$.
   (c) $T(n) = 9T(n/4) + n^2$.
   (d) $T(n) = 3T(n/2) + n$.
   (e) $T(n) = T(n/2 + \sqrt{n}) + n$.

(2) (20%) Design an efficient algorithm to find a spanning tree for a connected, weighted, undirected graph $G = (V, E)$ such that the weight of the maximum-weight edge in the spanning tree is minimized.

(3) (20%) Consider a set S of $n \geq 2$ distinct numbers given in unsorted order.
   (a) Design an algorithm in $O(n)$ time to determine $x, y \in S$ such that $|x-y| \geq |w-z|$ for all $w, x \in S$.
   (b) Design an algorithm in $O(n)$ expected time to determine $x, y \in S$ such that $x + y = Z$, where Z is given, or determine that no two such numbers exist.

(4) (10%) Mergesort will sort five numbers in eight comparisons in the worst case, but, since $\lceil \lg 5! \rceil = 7$, it is possible that some other algorithm, requiring at most seven comparisons, exists. Find one.

(5) (20%) Draw a decision tree for the Mergesort of the 3-element array. Is Mergesort optimal for $n = 3$?