Spring 1995 Farach

CS 513: Exam #1

You may consult your 1 page "cheat sheet." (5 points) Any theorem used in class or algorithm presented in class can be used. Other theorems or algorithms can only be used if you prove them.

1. (25 points)
   (a) (5 pts.) After one round of `ChoosePivot` and `Partition`, as given in class, what will be the state of the following array: $5, 7, 3, 4, 2, 9, 1, 8, 7, 3, 9, 5$.
   (b) (5 pts.) Prove or give a counter example: Every graph with k connected components (k maximal connected subgraphs) must have at least $n - k$ edges.
   (c) (5 pts.) Let $T(1) = 13, T(n) = 5T(n/4) + n^{3.1}$. What is the asymptotic complexity (Θ) of $T(n)$? Prove your answer.
   (d) (5 pts.) Order the following in increasing order of asymptotic complexity. Prove your answer:
   $$n \log^2 n, n \log \log n, (n^{\log \log n})/\log n, 2^n, 3^{n/5+2}, \log_n (n \log n)^2$$
   (e) (5 pts.) Prove or give a counter example: In every Fibonacci heap obtained from an initially empty structure by a sequence of n operations (each operation is one of INSERT , DECREASE, EXTRACTMIN), every node has depth $O(n^(1/2))$.
2. (25 points) The Majority Problem is defined as follows:

   Input: An array $A[1, n]$ of numbers.

   Output: m, if $|\{i|A[i] = m\}| > n/2$, if such an m exists, a \$ otherwise.

   Give an algorithm for solving the Majority Problem. Analyze its worst-case running time. Prove its correctness. The faster the algorithm, the more points you get.
3. (25 points) The All Closest Value Problem is defined as follows:

   Input: An array $A[1, n]$ of numbers.

   Output: An array $B[2, n]$ where $0 < B[i] < i$, $0 \leq A[i] - A[B[i]] \leq A[i] - A[j]$, for all $j : 0 < j < i$.

   Give an algorithm for computing the All Closest Value Problem. Analyze its worst-case running time. Prove its correctness. The faster the algorithm, the more points you get.
4. (25 points) Show that the All Closest Value Problem has an $\Omega(n \log n)$ lower bound in the comparison model.
5. **EXTRA CREDIT** The Minimum Ordered Sum Problem is defined as follows:

   Input: Arrays $A[1, n], B[1, n]$ of numbers.

Output: A pair (i, j), $1 \leq i \leq j \leq n$, such that $A[i] + B[j]$ is minimized.

Give an algorithm for computing the MINIMUM ORDERED SUM PROBLEM. Analyze its worst-case running time. Prove its correctness. Your algorithm must run in $O(n)$ time.

Good Luck