2009년 후기 알고리즘 자격시험 문제

1. (7 pts x 7 = 49 pts) For each of the following statements, give T (True) or F (False) and briefly justify your answer. Incorrect justification will earn no credit.

(1) Give a connected undirected graph $G = (V, E)$ with nonnegative edge weights w, the shortest-path tree from any source vertex $s \in V$ is a minimum spanning tree of G.

(2) Every problem in NP can be solved in exponential time.

(3) For any two positive functions f and g, if $g(n) = O(n)$, then $f(g(n)) = O(f(n))$.

(4) The solution to the recurrence $T(n) = T(n/3) + T(n/6) + n^{\sqrt{\log n}}$ is $T(n) = \Theta(n^{\sqrt{\log n}})$ (assume $T(n) = 1$ for n smaller than some constant c)

(5) There exists a comparison-based sorting algorithm that can sort any 6-element array using at most 9 comparisons.

(6) The best case running time for Insertion Sort to sort an n element array is O(n).

(7) Computing the median of n elements takes $\Omega(n \log n)$ time for any algorithm working in the comparison-based model.

2. (5 pts x 6 = 30 pts) Given 2 decision problems L_1 and L_2 in NP and $L_1 \leq_p L_2$, for each of the following statements, give one of T(true), F(false), or O(open question), and briefly justify your answer.

   (1) If $L_1 \in$ P, then $L_2 \in$ P.

   (2) If $L_2 \in$ P, then $L_1 \in$ P.

   (3) If $L_1 \in$ NPC, then $L_2 \in$ NPC.

   (4) If $L_2 \in$ NPC, then $L_1 \in$ NPC.

   (5) If $L_2 \leq_p L_1$, then L_1 and L_2 are NP-complete.

   (6) Suppose L_2 is solvable in O(n). Then L_1 is also solvable in O(n).

3. (21 pts) Suppose that n customers arrive simultaneously and start waiting to be served in a restaurant. (Assume that the restaurant was empty prior to their arrival.) The preparation time required to fulfill the order placed by customer i is $p_i (>0)$ minutes. The waiting time w_i of customer i is the preparation time p_i + the sum of the preparation times for the customers served before customer i. (For example, if there are 3 customers with preparation times $p_1 = 2$, $p_2 = 4$, $p_3 = 3$, and the customers are served in the order (3,1,2), then the waiting times are $w_1 = 5$, $w_2 = 9$, $w_3 = 3$.) The goal is to find an order to process the customers that minimizes the average waiting time $\frac{1}{n}\sum_{i=1}^{n} w\_i$. Give an efficient algorithm to compute an optimal order to serve the customers. Analyze the running time and space requirements of your algorithm and show the correctness of your algorithm.