Midterm Examination on Algorithms

Teacher: Biing-Feng Wang **April 17, 2007**

Problem 1: (10%, Growth of Functions)

(1) (4%) What's the definition of O-notation?

(2) (6%) Let $f(n)$, $g(n)$, and $h(n)$ be asymptotically nonnegative functions such that $f(n) = O(g(n))$ and $g(n) = O(h(n))$. Using the definition of O-notation, prove that $f(n) = O(h(n))$.

Problem 2: (15%, Recurrences)

(1) (8%) Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 4T(\lfloor n/2 \rfloor) + n$ for $n \geq 2$ and $T(1) = 1$

(2) (7%) Use the substitution method to verify your answer in (1).

Problem 3: (10%, Binary Heaps)

(1) (6%) Let $A_1, A_2, ..., A_k$ be k sorted lists. Let n be the total number of elements in all the lists. Give an efficient algorithm to merge all the lists into one sorted list.

(2) (4%) What's the time complexity of your algorithm? Explanation is necessary.

Problem 4: (10%, Sorting) Prove that the average-case running time of quicksort is $O(n \log n)$.

Problem 5: (10%, Sorting in linear time) Let $X_1, X_2, ..., X_m$ be m arrays. Each X_i contains an arbitrary sequence of integers within the range $[1, n^3]$, where n is the total number of elements in all the lists. Give an $O(n)$-time algorithm to sort each X_i. For example, given $m = 2$, $n = 8$, $X_1 = (3, 6, 9, 27, 1)$, and $X_2 = (4, 139, 6)$, the output is (1, 3, 6, 9, 27) and (4, 6, 139).

Problem 6: (15%, Greedy algorithms)

(1) (7%) Suppose you are given two sets A and B, each containing n positive integers. You can reorder each set however you like. After reordering, let a_i be the ith element in A and b_i be the ith element in B. Then, you will receive a payoff of $\Pi_{i=1}^{n} a_i^{b_i}$. Give an efficient algorithm that will maximize your payoff. What is the time complexity?

(2) (8%) Prove that your algorithm is correct.

Problem 7: (10%, Dynamic programming)

(1) (7%) Give an efficient algorithm to compute a LCS of two given sequences $X = (x_1, x_2, ..., x_m)$ and $Y = (y_1, y_2, ..., y_m)$.

(2) (3%) What is the running time of your algorithm? Justify your answer.

Problem 8: (10%, Design of Algorithms)

(7%) Suppose you have one machine and a set of n jobs a_1, a_2, ..., a_n to process on that machine. Each job a_j has a processing time t_j, a profit p_j, and a deadline d_j. The machine can process only one job at a time, and job a_j must run uninterruptedly for t_j consecutive time units. If job a_j is completed by its deadline d_j, you receive a profit p_j, but if it is completed after its deadline, you receive a profit of 0. Give an algorithm to find the schedule that obtains the maximum amount of profit, assuming that all processing times are integers between 1 and n.

(3%) What's the running time of your algorithm? Justify your answer.

Problem 9: (10%, Homework) This problem is to verify whether you had done homework by yourself. Please answer either of the following. (If you answer both, only the one getting lesser score will be counted.)

(a) An array $A[1..n]$ contains all the integers from 0 to n except one. In this problem, we cannot access an entire integer in A with a single operation. The elements of A are represented in binary, and the only operation we can use to access them is "fetch the jth bit of $A[i]$," which takes constant time. Give an efficient algorithm to determine the missing integer.

(b) Let $X[1..n]$ and $Y[1..n]$ be two arrays, each containing n numbers already in sorted order. Give an efficient algorithm to find the median of all $2n$ elements in arrays X and Y.

Bonus: (10 %, self-taught) Describe the procedure Tree-Delete(T, z) in Chapter 12.