COT5405: ANALYSIS OF ALGORITHMS

MidTerm Exam I

Date: Sep 27, 2007, Thursday
Time: 1:55pm – 3:55pm
Professor: Alper Üngör (Office CSE 430)

This is a closed book exam. No collaborations are allowed. Your solutions should be concise, but complete, and handwritten clearly. Use only the space provided in this booklet, including the even numbered pages. Feel free to give reference to the algorithms, definitions and concepts discussed in class rather than describing them in detail. There are no bonus points.

GOOD LUCK!

Your name: ______________________________

	Credit	Max
Problem 1		15
Problem 2		15
Problem 3		20
Problem 4		20
Problem 5		20
Total		100

1. [15 points = 3 + 3 + 3 + 3 + 3] TRUE/FALSE QUESTIONS (NO NEED FOR JUSTIFICATION)

(a) TRUE/FALSE
Let $T(n) = T(\alpha n) + T((1-\alpha)n) + cn$, where α is a constant in the range $0 < \alpha < 1$, c is a positive constant, and $T(1)$ is $O(1)$. Then, $T(n)$ is $O(n \log n)$.

(b) TRUE/FALSE
Given an open-addressed hash table with load factor α, the number of probes in an unscuccessful search is at most $1/(1-\alpha)$.

(c) TRUE/FALSE
$f(n)$ is $O(g(n))$ if and only if $g(n)$ is $\Omega(f(n))$.

(d) TRUE/FALSE
Given a set of n numbers, one can output in sorted order the k elements following the median in sorted order in $O(n + k \log k)$ time.

(e) TRUE/FALSE
Radix sort requires an *in-place* auxiliary sort in order to work properly.

2. [15 points] Interval (Red-Black) Trees
Let $A = \{[1,4], [3,14], [7,9], [5,12], [8,23], [14,16]\}$ be a set of intervals. Recall the augmented data structure presented in class to maintain intervals using red-black trees. Construct an interval (red-black) tree of A, verifying its properties. Is this tree unique?

3. [20 points = 6 + 6 + 8] HEAPS

Recall the array-based implementation of a max-heap, where each insertion involves a bottom up traversal (with potential exchanges) in the heap tree. Suppose we construct a max-heap by successively inserting n distinct items into an initially empty heap.

(a) How many different trees can you get for $n = 3$ items, e.g., {1,2,3}? Draw the trees.

(b) How many different trees can you get for $n = 4$ items? Draw the trees.

(c) Is it true that if you pick a random sequence of n items then each of the possible trees is equally likely? Justify your answer.

4. [20 points = 5 + 5 + 10] DYNAMIC PROGRAMMING
We are given the prices of a stock for n consecutive days. Let p_i represent the price of a stock on day i. Let $\Delta_{i,j} = p_j - p_i$ denote the amount of money an investor makes (or loses) by buying this stock on day i and selling it on a *later* day j.

(a) Describe a linear time algorithm to compute the maximum possible profit an investor could make by a single buy-and-sell transaction, i.e., $max_{1 \le i < j \le n}\{\Delta_{i,j}\}$.

(b) Let $F(t) = max_{1 \le i < j \le t}\{\Delta_{i,j}\}$ denote the maximum profit an investor could make in the first t days by a single buy-and-sell transaction. Similarly, let $L(t) = max_{t \le i < j \le n}\{\Delta_{i,j}\}$ denote the maximum profit an investor could make in the last t days by a single buy-and-sell transaction. Adapt your algorithm in part (a) for computing $F(t)$ and $L(t)$ for all $t = 1, \dots, n$ in linear time.

(c) Describe a linear time algorithm to compute the maximum possible profit an investor could make by two buy-and-sell transactions. Specifically, your algorithm should find i, j, k, l that maximizes $\Delta_{i,j} + \Delta_{k,l}$ such that $i < j < k < l$.

5. [20 points = 3 + 3 + 2 + 6 + 6] LOWER/UPPER BOUNDS

Consider the problem of finding the three largest numbers (in decreasing order) from a collection of 8 distinct numbers using comparisons.

Let u be an upper bound on the number of comparisons used by an algorithm (say Alg A) that solves this problem. Let l be a lower bound on the number of comparisons required to solve this problem by any algorithm. (Think of u and l simply as numbers, like "10" or "20", since there is no input size parameter such as n in the problem).

(a) Is it possible that $u < l$? What would you conclude about Alg A and its optimality?

(b) Is it possible that $u > l$? What would you conclude about Alg A and its optimality?

(c) Is it possible that $u = l$? What would you conclude about Alg A and its optimality?

(d) Descibe an algorithm with a good upper bound u.

(e) Find a good lower bound l on the number of comparisons. (Hint: Consider the # of leaves in a decision tree.)

[For parts (d) and (e), your bounds are certainly good if $u - l \leq 3$. Otherwise, partial credit will be given.]