COMP 122 – Algorithms and Analysis

Practice Midterm

Tuesday, October 12, 1999

Problem 1. (10pt): Consider the code for Build-Heap in Chapter 7.3 on page 145 of the textbook, that operates on a heap stored in an array $A[1 \cdots n]$. Show how this procedure can be implemented as a recursive divide-and-conquer procedure Build-Heap(A, i), where $A[i]$ is the root of the sub-heap to be built. (i.e. Write the pseudo code for the recursive version of Build-Heap.) To build the entire heap, we would call Build-Heap$(A, 1)$. Give a recurrence that describes the worst-case running time of this procedure and solve the recurrence.

Problem 2. (10pt): Give asymptotically tight upper (big O) bounds for $T(n)$ in each of the following recurrences. Justify your solutions by naming the particular case of the Master Theorem, by iterating the recurrence, or by using the substitution method.

(a) $T(n) = T(n-2) + 1$

(b) $T(n) = 2T(n/2) + nlg^2n$

(c) $T(n) = 9T(n/4) + n^2$

(d) $T(n) = 3T(n/2) + n$

(e) $T(n) = T(n/2 + \sqrt{n}) + n$

Problem 3. (10pt): Consider a set S of $n \geq 2$ distinct numbers given in unsorted order. In $O(n)$ worst-case time, determine $x, y \in S$ such that

$$|x - y| \leq \frac{1}{n-1}(\max_{z \in S} z - \min_{z \in S} z)$$

i.e. determine any two numbers that are at least as close together as the average distance between consecutive numbers in the sorted order. Basically, the input to your algorithm is in an array A. The average distance among all input elements in A is $\frac{1}{n-1}(\max_{z \in S} z - \min_{z \in S} z)$, where $n = length[A]$. You need to design an $O(n)$ time algorithm that finds any two numbers that are at least as close together as the average distance among all elements in A. (HINT: Use divide-and-conquer.) In as few words as possible, describe your algorithm and justify its running time. To keep your answers brief, use algorithms from lectures and the textbook as subroutines, if appropriate.

Problem 4. (10pt): Answer the following questions.
(a) Prove that $(n+1)^2 = O(n^2)$ by giving the constants n_0 and c so $(n+1)^2 \leq cn^2$ for all $n \geq n_0$.
(b) Are $\Omega(nlgn)$ comparisons necessary to sort any sequence of n distinct integers? Why or why not? Justify your answer.

Problem 5. (10pt): List the following functions in increasing asymptotic order. If two functions have equal asymptotic growth rate, then indicate this. (Remember that $lg\ n = log_2\ n$)

$n^2(lglgn)^2$ $\qquad \sqrt{n}$ $\qquad (log_4\ n)^3$ $\qquad n^2lgn$ $\qquad 4^{lgn}$ $\qquad lg(n^5)$