Purdue University
CS 381

Gopal/Hambrusch
Spring 2008

Assignment 2

Due: Friday, January 25, 2008 (before class)

1) (15 pts.) Assume you are given an array A of size n containing unsorted, distinct, positive integers. The isolation number of element $A[i]$ is defined as $I(i) = \sum_{1 \leq k \leq n} |A[i] - A[k]|$, $1 \leq i \leq n$. For example, for $A = [2, 5, 12, 3]$, we have $I[2] = 3 + 0 + 7 + 2 = 12$. Describe and analyze an $O(n \log n)$ time algorithm for computing the isolation number for every element of the array.

2) (15 pts.) CLRS: page 59: 3.4 b, d, f

3) (a) (5 pts.) Solve the recurrence $T(n) = T(n/4) + T(3n/4) + n$ using the recursion tree approach. Assume that base cases of the recurrence are constants i.e., $T(n) = c$, for $n \leq 4$ where c is a constant. You may assume that n is a power of 4.

(b) (15 pts.) We are given a sequence of n distinct numbers $a_1, \ldots, a_n$. Two elements a_i and a_j are a *friendly pair* if $i < j$ and $a_i < a_j$. We are interested in *counting* the number of friendly pairs in the sequence. Design an $O(n \log n)$ time divide-and-conquer algorithm for determining the number of friendly pairs. Analyze the running time of your algorithm by setting up a recurrence and solving it using the Master Theorem.

Comments on problems 1 and 3(b) you should consider.

- Both problems operate on distinct integers stored in an array of size n and have a straightforward $O(n^2)$ solution. For problem 1, an $O(n^2)$ solution will give partial credit. For problem 3(b), an $O(n^2)$ solution will not result in partial credit.
- The isolation number problem outputs n integers and the friendly pair problem outputs one integer. A problem generating n integers will at least have linear running time (after setting up the array of size n). This means an $O(n)$ time solution for the isolation problem is asymptotically optimal. An $O(n)$ solution could, for example, generate one output value from the previous in O(1) time.
- Consider whether the initial order of the elements in the array is relevant and sorting the array makes the problem easier. Note that for the friendly pair problem the initial indices define a friendly pair.
- For the friendly pair problem consider the logical process underlying the combining of two sorted arrays used in Mergesort.