Homework #3
Introduction to Algorithms/Algorithms 1
600.363/463
Spring 2014

Due on: Tuesday, Feb 18th, 5pm
Late submissions: will NOT be accepted
Format: Please start each problem on a new page.
Where to submit: On blackboard, under student assessment
Please type your answers; handwritten assignments will not be accepted.
To get full credit, your answers must be explained clearly,
with enough details and rigorous proofs.

February 11, 2014

1 Problem 1 (20 points)

Given an array A of n numbers, call index i, with $1 \leq i \leq n$, a *strong* index if for all j for which $1 \leq j < i$, we have $A[i] > A[j]$. For example, for the list $1, 5, 2, 3, 6$, indices 1, 2 and 5 are strong indices, while indices 3 and 4 are not. A set of n distinct numbers is randomly permuted into the array $A[1..n]$. Show that the expected number of strong indices in array A is $O(\log n)$. Hint: a review of CLRS appendix C and material on the harmonic numbers will be helpful.

2 Problem 2 (20 points)

2.1 (10 points)

Resolve the following recurrences. Use the master theorem, if applicable. In all examples assume that $T(1) = 1$. To simplify your analysis, you may assume that $n = a^k$ for some a, k.

1. $T(n) = 3T(n/2) + 1$

2. $T(n) = T(n/2) + 2\sqrt{n}$
3. $T(n) = 16T(n/16) + n^{\frac{3}{2}}$
4. $T(n) = 28T(n/3) + n^3$
5. $T(n) = nT(n/2)$
6. $T(n) = 2T(n-1) + 1$
7. $T(n) = 8T(n/2) + n^3$
8. $T(n) = T(n/2) + n \log n$

2.2 (10 points)

A sequence $a_1, a_2, \ldots, a_n$ has a *dominant element* if more than half of the elements in the sequence are the same. For example, 3 is a dominant element in the sequence $7, 3, 3, 3, 1, 3, 3, 4, 5, 3$. On the other hand, the sequence $5, 4, 1, 1, 2, 3, 2, 3, 6$ has no dominant element. Give a divide and conquer algorithm that runs in time $O(n \log n)$ and finds and returns a dominant element in a sequence of n numbers or returns `None` if no such element exists. Prove the correctness of your algorithm and prove that its runtime is $O(n \log n)$. (Note: there exists an $O(n)$ algorithm to solve this problem that doesn't make use of divide and conquer– if you figure it out, you may prove its correctness and runtime instead.)

3 Optional Exercises

Solve the following problems and exercises from CLRS: 4-3, 4-1, 7-3.