Piyush Kumar

**Analysis of Algorithms, Fall 2010**

Assignment #5
Due: April 8th

Each of the problems should be solved on a separate sheet of paper to facilitate grading. Please don't wait until the last minute to look at the problems.

Problem 1 An array $T$ contains $n$ elements. You want to find the $m$ smallest, where $m$ is much smaller than $n$. Give an efficient algorithm to solve this problem. (Note that $m$=1 becomes the problem of finding the minimum element)

Problem 2

Solve the following recurrence relations:

- $T(n) = 2T(n/3) + 1$
- $T(n) = T(n-1) + 2$
- $T(n) = T(\sqrt{n}) + 1$

Problem 3 What is the sum of the $n$-th roots of unity? What is their product if $n$ is odd? If $n$ is even?

Problem 4 Suppose that you want to multiply the two polynomials $(x + 2)$ and $(x^2 + 1)$ using the FFT. Choose an appropriate power of two, find the FFT of the two sequences, multiply the results componentwise, and compute the inverse FFT to get the final result.

Problem 5 Show that any array of integers $x[1 \ldots n]$ can be sorted in $O(n + M)$ time, where $M = \max_i x_i - \min_i x_i$. For small $M$, this is linear time: why doesn't the $\Omega(n \log n)$ lower bound apply in this case?

Problem 6 If you have solved problem 3 and 4 of this homework, solve Chapter 5, Exercise 4 from the text book.

Problem 7 Give an efficient algorithm for finding the largest independent set for a given tree $T$ with root node $r$. (Perhaps try a linear time solution?)