HW2 (due Thursday 10/12/06 at the beginning of class)

1. (Half a page) In the worst-case linear time Select algorithm described in class the input elements were divided into groups of 5. Prove that the algorithm does not run in linear time if the input elements are divided into groups of 3.

2. (Half a page) Let $X[1..n]$ and $Y[1..n]$ be two arrays, each containing $n$ numbers already in sorted order. Give an $O(\log n)$-time algorithm to find the median of all the $2n$ elements in arrays $X$ and $Y$.

3. (Half a page) In mergesort we merge two sorted lists into a single sorted list. This problem is about generalizing this to $k$ lists. Give an $O(n \log k)$-time algorithm to merge $k$ sorted lists into a single sorted list, where $n$ is the total number of elements in all the input lists.

4. (Half a page) The Federal Govt plans a perfectly straight East-West expressway across a state with $n$ cities (you know their coordinates, which are given unsorted) through a point of the State Govt's choice. The State plans to connect these $n$ cities to this expressway by constructing perfectly straight North-South feeder roads. Though the Feds will pay for the expressway, the State must pay for the feeder roads, and you can assume that the cost is linear in terms of the length of the feeder roads. Devise an algorithm to determine the location ($y$-coordinate) of the expressway to minimize the State's cost, and justify its correctness (why choosing that point would minimize the State’s cost) and complexity. Will your answer (complexity) change if you had the coordinates pre-sorted in some way (of your choice)?

5. (Half a page) We all know that Binary trees can be traversed in one of three ways (pre-order, in-order, post-order). Given one or more ordered traversals, what are the minimum requirements for constructing a tree? That is, is it possible to construct the tree uniquely from any of the above traversals alone, or some minimum combination of the aforesaid traversals are required? Justify your answer.

6. (Two pages) Insert 7,8,9,1,2,3,4,5,6 in that sequence into a Binary-search tree. Then delete the nodes in the same sequence, i.e. 7,8,9,1,2,3,4,5,6. Do the same (inserts and deletes) for a red-black tree. Mark the red nodes with double circle (with a little bit of shade if possible) and the black nodes as single circle with no shading. The numbers (elements of the data structures) should be put inside the circles.

7. (Half a page) Consider an ordinary binary min-heap data structure with $n$ elements that supports the instructions INSERT and DELETE-MIN in $O(\log n)$ worst-case time. Give a potential function $\Phi$ such that the amortized cost of INSERT is $O(\log n)$ and the amortized cost of DELETE-MIN is $O(1)$, and show that it works.

8*. (One Page) In the Union-by-rank and Find-Set with path compression algorithm describe in class, prove that every node has rank at most $\lfloor \log n \rfloor$. 