SPRING 2005: **COT 5993** INTRO. TO ALGORITHMS

[HOMEWORK 3; DUE MAR 29 AT START OF CLASS]

How to write algorithmic solutions: An ideal algorithmic solutions must show **Basic Idea**, **Algorithm**, **Proof of Correctness**, and **Time and Space Complexity Analysis**.

Reminder: ADD A SIGNED STATEMENT THAT YOU HAVE ADHERED TO THE COLLABORATION POLICY FOR THIS CLASS AND THAT WHAT YOU ARE PRESENTING IS YOUR OWN WORK.

Problems

22. (**Regular**) You are given a set of n elements. However, the input has a large number of duplicates, and the number of unique elements in the list are only k (not a constant). Design a simple comparison-based sorting algorithm that sorts using $o(n \log n)$ comparisons. Analyze its time complexity in terms of n and k. **Hint:** Use red-black trees.
23. (**Regular**) Solve problem 9.3-1, page 192.
24. (**Exercise**) Solve problem 9.3-3, page 192. No details are needed. Couple of sentences should be sufficient.
25. (**Regular**) Solve 9.3-6, page 192. You may assume that k is a power of 2.
26. (**Exercise**) Solve 12.2-1, page 259.
27. (**Extra Credit**) Solve 12.2-8, page 260.
28. (**Exercise**) Solve 13.3-2, page 287. Handdrawn trees are acceptable.
29. (**Regular**) Solve 14.1-5, page 307.
30. (**Regular**) Solve 14.2-3, page 310.
31. (**Exercise**) Solve 16.3-2, page 392.
32. (**Exercise**) Solve 15.4-1, page 355.
33. (**Exercise**) Read and understand the application described in Sections 15.1, 15.2, or 15.5.
34. (**Regular**) In class, we discussed greedy and dynamic programming algorithms to solve the *activity selection problem.* Consider the following modification to the problem. Assume that you are going to be paid a bonus of v_i dollars, if you scheduled activity a_i. First show that a greedy algorithm will fail to find the optimal solution. Then design an algorithm to find a set of non-overlapping activities that would leave you with the largest possible bonus.