FALL 1998: **COMP 7/8713: Advanced Algorithms**

Due Sep 23, in class

Every homework will have three kinds of problems. Turn in the solutions for problems marked **R** (regular) or **Ex** (exercises). Those marked **R** will be graded; those marked **Ex** will not be graded. Problems marked **EC** are extra credit problems and may be turned in any time before December 9. Only fully correct solutions will get credit. Discuss the solution with me before you write it up.

Collaboration Policy

Solving an algorithmic problem is a creative process. When presented with a new problem, it is your task to "take it apart" and reach your own understanding. This is a painstaking and time consuming process. Getting help from others destroys this process. However, discussing with others after you have spent some time with a problem can help the process and bring other aspects of the problem to light. You may discuss homework problems with other students in your class, after you have given it sufficient thought. However, when the time comes to write up your solution, it must be your own work, and it must be in your own words. Acquiring or copying solutions from other people is a violation of academic regulations. Of course, I encourage you to come and talk to me about any problem (after you have spent some amount of time and thought on them). If your solution was obtained from some other source (or with help from someone else), please mention your source. These restrictions do not apply to the *Exercises* and the *Extra Credit* problems. In fact, I strongly encourage you to work together on those problems.

Pay careful attention to the final written solution. Reread your written solutions. Algorithms, like programs, need to be commented. Before presenting a complicated algorithm, explain in English your basic idea. A well-written solution shows clarity of thought and is likely to receive better grades. Every algorithm needs a proof of correctness. An example is not a proof of correctness. Every algorithm needs to be analyzed.

Problems

1. (**R**) Exercise 15.3-5, page 295.
2. (**R**) Exercise 15.3-6, page 295.
3. (**EC**) Exercise 15-1, page 295.
4. (**R**) Exercis 16.3-5, page 319.
5. (**EC**) Exercis 16.3-6, page 319.
6. (**Ex**) Exercise 16.4-2, page 324.
7. (**R**) Prove using the decision tree model that $\omega(\log n)$ is a lower bound for the number of comparisons needed to do a search for an item in a sorted list of n items.
8. (**EC**) Prove a lower bound of $\lceil \frac{3n}{2} \rceil - 2$ for the number of comparisons needed for computing the minimum and maximum of n arbitrary numbers.
9. (**EC**) Design an efficient algorithm to compute the largest and second largest of n numbers. Analyze (precisely) the worst-case number of comparisons. Prove a lower bound (note that constants are very vital in this analysis).