ECS 122A: Algorithm Design and Analysis — Handout 5
UC Davis — Phil Rogaway — April 28, 2000

Problem Set 5—Due Friday, May 5

Problem 1.

(a) Let a and b be distinct real numbers. Write a pair of assignment statements (that is, x = <something>; y = <something>) equivalent to:

```
if (a<b) {x=a; y=b;}
    else {x=b; y=a;}
```

The right hand side of your assignment statements (what we called "<something>" above) may use only the arithmetic operators +, -, *, / and functions sqrt, exp, ln. Don't assume anything about the representation of real numbers.

(b) Generalize part (a): show how to sort n distinct real numbers, for any n, in zero comparisons, without assuming anything about the internal representation of the numbers.

(c) Does what you've just done contradict the $\lceil \lg n! \rceil$ lower bound we did for sorting n numbers? Explain. *Comment:* Reflect seriously on this question; a satisfying answer is *not* obvious.

Problem 2. The greedy algorithm for the THIEF'S PROBLEM is as follows: starting with the emptyset of items S to steal, repeatedly add to S the item i not yet in S with largest v_i/w_i but subject to constraint that adding i to S will not make S have weight exceeding B. (Notation is as in class: v_i is the value of item i; w_i is the weight of item i; and B is the total weight that can be carried.) Prove that the greedy algorithm does *not* correctly solve this problem.

Problem 3. Exercise 12.3–3 from page 232 of your text.

Problem 4. Problem 16–4 (a) from page 326 of your text.

I'm going to start you on your way—here's how my solution begins.

Arrange the employees in an ordered tree and number them $1, 2, \ldots, n$, beginning at a left-most leaf and marching left-to-right and then up until you reach the root of the tree (so that the president is employee n). Each employee has a conviviality rating $c(i)$. For any employee i let $E(i)$ denote the set of employees of i. If i has no employees, $E(i) = \emptyset$. By our numbering convention, note that $j \in E(i) \Rightarrow j < i$. This will be needed for our algorithm.

Below we assume that $c(i) \geq 0$ since employees with negative conviviality ratings can be eliminated from consideration.

Let us say that a set of employees $T \subseteq \{1, \ldots, n\}$ is *permissible* if $i \in T$ implies $E(i) \cap T = \emptyset$. We want to find a set $T \subseteq \{1, \ldots, n\}$ which is permissible and, among all permissible subsets of $\{1, \ldots, n\}$, T should be one which maximizes $C = \sum_{i \in T} c(i)$. To find this T we are going to over-achieve and find various other things as well. This always happens in dynamic programming problems.

...