Homework problems:

6.1 (due Oct 19, 2006) Let $G = (V, E)$ be a digraph given in the adjacency-list representation (i.e., for each vertex $v \in V$ we have a (linked) list of out-neighbors of $v$).

1. Write pseudocode for a procedure which outputs an adjacency-list representation of the reverse digraph (i.e., $G$ with each edge reversed). The procedure should run in time $O(|V| + |E|)$.

2. Write pseudocode for a procedure which outputs the adjacency-list representation of $G$ in which the out-neighbors of each vertex are listed in the increasing order. The procedure should run in time $O(|V| + |E|)$.

3. Write pseudocode for a procedure which checks if $G$ is undirected (i.e., the reverse of every $e \in E$ is also in $E$). The procedure should run in time $O(|V| + |E|)$.

6.2 (due Oct 19, 2006) Let $N$ be an integer. The company W produces weights of weights 1, \ldots, $N$. To place an order the customer has to send a list of integers $a_1, \ldots, a_N$. The company W will then send the customer $a_i$ weights of weight $i$ for each $i \in \{1, \ldots, N\}$. A package sent through UPS can weigh at most $N$. The company now faces the following problem: how to pack the order into the smallest number of packages (so that no package weighs more than $N$)?

Their new employee (a graduate of U of R) suggested the following greedy algorithm: Take an empty package and repeatedly add the heaviest item which fits. When no item fits then seal the package, get a new empty package, and repeat.

1. Show (prove) that the greedy algorithm works for $N = 6$.

2. Show (find a counterexample) that the greedy algorithm does not work for $N = 10$.

6.3 (due Oct 19, 2006) Write a dynamic programming algorithm which for a given number $n$ finds the smallest number of squares which sum to $n$ (for example for $n = 7$ we need 4 squares ($7 = 2^2 + 1^2 + 1^2 + 1^2$), whereas for $n = 13$ we only need 2 squares ($13 = 3^2 + 2^2$)). Implement your algorithm and find all numbers from $\{1, 2, \ldots, 100\}$ which need 4 squares. Use “The On-Line Encyclopedia of Integer Sequences” to find a formula for the numbers which need 4 squares.

6.4 (due Oct 19, 2006) We are given a sequence of $n$ positive numbers $a_1, \ldots, a_n$. Give an algorithm which finds the increasing subsequence of $a_1, \ldots, a_n$ with the maximal sum. (For example on input $1, 101, 2, 3, 100, 4, 5$ your algorithm should output $1, 2, 3, 100$.)

Bonus problem:

6.5 (due Oct 19, 2006) Determine for which values of $N$ does the greedy algorithm in Problem 6.2 work.