University of Washington
Department of Computer Science and Engineering
CSE 521, Spring 1998

April 21, 1998

Homework 4, **Due April 29, 1998**

Problem 1. Weighted activity selection :

Suppose that the activity selection problem in the text is augmented by giving each activity i a weight w_i and the problem is to maximize the weight of the activities selected. Give a dynamic programming algorithm to solve this problem.

Problem 2. From the text :

Page 337, Exercise 17.2-4.

Problem 3. From the text :

Page 337, Exercise 17.2-6.

Problem 4. Two processor scheduling:

Given tasks $\{T_1, \ldots, T_n\}$, each with a processing requirement t_i and a deadline d_i (where t_i and d_i are positive integers), determine if there is a schedule where tasks are completed at or before their deadlines. For this problem, assume that up to two tasks may be executed at once, and that tasks may not be preempted.

The problem is NP-complete. However, if the deadlines are bounded by an integer D, then a dynamic programming algorithm can be used (with run time dependent on D).

Give an $O(nD)$ algorithm to solve this problem when all the deadlines are equal to D.

Give an $O(nD)$ algorithm to solve this problem when all the deadlines are at most D. Argue that your algorithm is correct. (It is not critical to come up with $O(nD)$ runtime - anything that is polynomial in n and D is okay.)

Problem 5. Another scheduling problem:

Page 354, Problem 17-3, part A.

Problem 6. Dense bipartite subgraph:

The problem is: Given a bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$, find subsets $U' \subset U$ and $V' \subset V$ with $|U'| = |V'| = \frac{n}{2}$ that maximizes the number of edges between U' and V'.

One approach to this problem is to use a greedy algorithm: Let U' be the $\frac{n}{2}$ vertices of U with maximum degree, and then let V' be the $\frac{n}{2}$ vertices of V with maximum degree with respect to U'.

Let $GREED(G)$ be the solution (meaning the set of edges between U' and V') found by this algorithm, and let $OPT(G)$ be the solution which maximizes the number of edges in the subgraph.

Prove that $|GREED(G)| \geq \frac{1}{2} |OPT(G)|$.

Prove that this is the best possible bound for the greedy algorithm, by showing that for every $\epsilon > 0$ there exists a graph G such that $|GREED(G)| < (\frac{1}{2} + \epsilon) |OPT(G)|$.