University of Washington
Department of Computer Science and Engineering
CSE 521, Spring 2001

April 17, 2001

Homework 4, **Due Wednesday, April 25, 2001**

Reading After covering greedy algorithms, we are going to jump to the Graph Algorithms portion of the text. I am going to assume that you are familiar with the basic material from Chapter 23 (basic representations, BFS, DFS, Topological Sort, Strongly Connected Components). I will cover Minimum Spanning Trees, but will take a slightly different approach. I will assume that you have seen Minimum Spanning Tree algorithms before. We will spend a greater amount of time on path finding algorithms (chapters 25 and 26).

Problem 1. From the text:
CLR, Page 333, Exercise 17.1-1.

Problem 2. From the text:
CLR, Page 333, Exercise 17.1-2.

Problem 3. From the text:
CLR, Page 337, Exercise 17.2-4.

Problem 4. From the text:
CLR, Page 337, Exercise 17.2-6. (The challenging part of this problem is understanding the meaning of Problem 10-2.)

Problem 5. 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. (If you don't see how to do it in $O(nD)$, anything polynomial in n and D is okay).

(Challenge - i.e., for the fun of it) Give an $O(nD)$ algorithm to solve this problem when all the deadlines are at most D. Argue that your algorithm is correct. (Again, it is not critical to come up with $O(nD)$ runtime - anything that is polynomial in n and D is okay.)