University of Washington February 1, 2000
Department of Computer Science and Engineering
CSE 521, Winter 2000

Homework 5, **Due Wednesday, February 9, 2000**

Reading Read chapters 23-26.

Problem 1. Spanning Trees:
Page 503, Exercise 24.1-6.

Problem 2. Kruskal's Algorithm :
Page 510, Exercise 24.2-4. Remember that you can sort in linear time if the elements are from a restricted range (see chapter 9).

Problem 3. Shortest Paths:
Page 527, Exercise 25.1-7.

Problem 4. Dijkstra's Algorithm:
Page 532, Exercise 25.2-5.

Problem 5. Stacking blocks:
Suppose that there are n types of blocks, and three blocks of each type. Each type-i block is a rectangular solid with linear dimensions (x_i, y_i, z_i). A block can be oriented so that any two of its three dimensions determine the dimensions of a base and the other dimension is the height. In building a tower, one block may be placed on top of another block as long as the two dimensions of the base of the upper block are each *strictly* smaller than the corresponding base dimensions of the lower block. (Thus, for example, blocks oriented to have equal-sized bases cannot be stacked.) Design a polynomial time algorithm to determine the tallest tower that can be built. (The solution that I have in mind is $O(n^2)$, you may be able to do better than this.)

(Note: actually, you can only use two blocks of each type, due to the stacking constraint. The problem is NP-complete if there is only one block of each type. (The NP-completeness result is very challenging, if you have tons of free time, and really enjoy finding NP completeness proofs, you might want to tackle this one.)