COMP 271 Design and Analysis of Algorithms
2003 Spring Semester
Questions for Seventh Tutorial – Week 0f April 15, 2003.
All problems are from Question Bank 3 – Solutions available there (in revised solutions to Question Bank 3)

1. A student in class suggested the following algorithm for the chain matrix multiplication problem:

   > Suppose $n > 2$ and p_i is the smallest of $p_0, \cdots, p_n$. Break the product after A_i, and recursively apply this procedure to the product $A_1..A_i$ and $A_{i+1}..A_n$.

   Does this algorithm work (i.e, does it minimize the number of multiplications needed)? If yes, prove your conclusion. If not, give a counter-example.

2. In this question, you are required to solve the 0-1 Knapsack problem for *two* knapsacks. You are given a set of n objects. The weights of the objects are $w_1, w_2, \ldots, w_n$, and the values of the objects are $v_1, v_2, \ldots, v_n$. You are given two knapsacks each of weight capacity C. If an object is taken, it may placed in one knapsack or the other, but not both. All weights and values are positive integers. Design an $O(nC^2)$ dynamic programming algorithm that determines the maximum value of objects that can be placed into the two knapsacks. Your algorithm should also determine the contents of each knapsack. Justify the correctness and running time of your algorithm.

3. Give an $O(n^2)$ time dynamic programming algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers (i.e, each successive number in the subsequence is greater than or equal to its predecessor). For example, if the input sequence is $\langle 5, 24, 8, 17, 12, 45 \rangle$, the output should be either $\langle 5, 8, 12, 45 \rangle$ or $\langle 5, 8, 17, 45 \rangle$.

4. Run the Floyd-Warshall algorithm on the weighted, directed graph shown in the figure. Show the matrix $D^{(k)}$ that results for each iteration of the outer loop.