Homework#9 Dynamic programming

Textbook:

7.5. Use Algorithm LCS to find the length of a longest common subsequence of the two strings $A =$ "xzyzzyx" and $B =$ "zxyyzxz". Give one longest common subsequence.

7.11. Consider applying Algorithm MATCHAIN on the following five matrices:

$$M_1 : 2 \times 3, \quad M_2 : 3 \times 6, \quad M_3 : 6 \times 4, \quad M_4 : 4 \times 2, \quad M_5 : 2 \times 7.$$

(a) Find the minimum number of scalar multiplications needed to multiply the five matrices, (that is $C[1,5]$).

(b) Give a parenthesized expression for the order in which this optimal number of multiplications is achieved.

7.26. In order to lower the prohibitive running time of the knapsack problem, which is $\Theta(nC)$, we may divide C and all the s_i's by a large number K and take the floor. That is, we may transform the given instance into a new instance with capacity $\lfloor C/K \rfloor$ and item sizes $\lfloor s_i/K \rfloor, 1 \le i \le n$. Now, we apply the algorithm for the knapsack discussed in Sec. 7.6. This technique is called *scaling and rounding* (see Sec. 15.6). What will be the running time of the algorithm when applied to the new instance? Give a counterexample to show that scaling and rounding does not always result in an optimal solution to the *original* instance.

7.30. Consider the *money change* problem. We have a currency system that has n coins with values $v_1, v_2, \ldots, v_n$, where $v_1 = 1$, and we want to pay change of value y in such a way that the total number of coins is minimized. More formally, we want to minimize the quantity

$$\sum_{i=1}^{n} x_i$$

subject to the constraint

$$\sum_{i=1}^{n} x_i v_i = y.$$

Here, $x_1, x_2, \ldots, x_n$ are nonnegative integers (so x_i may be zero).

(a) Give a dynamic programming algorithm to solve this problem.

(b) What are the time and space complexities of your algorithm?

(c) Can you see the resemblance of this problem to the version of the knapsack problem discussed in Exercise 7.27? Explain.

7.34. Let $G = (V, E)$ be a directed acyclic graph (dag) with n vertices. Let s and t be two vertices in V such that the indegree of s is 0 and the outdegree of t is 0. Give a dynamic programming algorithm to compute a longest path in G from s to t. What is the time complexity of your algorithm?

SubsetSum. Give an O(nt) algorithm for the following task.
Input: A list of n positive integers $a_1, a_2, \ldots, a_n$; a positive integer t.
Question: Does some subset of the a_i's add up to t? (You can use each a_i at most once.)

PickCards. Consider the following game. A "dealer" produces a sequence $s_1 \cdots s_n$ of "cards," face up, where each card s_i has a value v_i. Then two players take turns picking a card from the sequence, but can only pick the first or the last card of the (remaining) sequence. The goal is to collect cards of largest total value. (For example, you can think of the cards as bills of different denominations.) Assume n is even.

(a) Show a sequence of cards such that it is not optimal for the first player to start by picking up the available card of larger value. That is, the natural *greedy* strategy is suboptimal.

(b) Give an $O(n^2)$ algorithm to compute an optimal strategy for the first player. Given the initial sequence, your algorithm should precompute in $O(n^2)$ time some information, and then the first player should be able to make each move optimally in O(1) time by looking up the precomputed information.