Dynamic Programming: AlgoLab 6

You have to use only dynamic programming method to solve following programs.
Hint: http://people.csail.mit.edu/bdean/6.046/dp

Lab 7

1. Write a C or a C++ program to find i^{th} Fibonacci number.
2. **Longest Increasing Subsequence:** Given a sequence of n real numbers $A_1 ... A_n$, determine a subsequence (not necessarily contiguous) of maximum length in which the values in the subsequence form a strictly increasing sequence.
3. **Maximum Value Contiguous Subsequence:** Given a sequence of n real numbers $A_1, ..., A_n$, Give a linear time algorithm to determine a contiguous subsequence $A_i, ..., Aj$ for which the sum of elements in the subsequence is maximized. For instance, if sequence is
   $5, 15, -30, 10, -5, 40, 10.$
   then $15, -30, 10$ is a contiguous subsequence but $5, 15, 40$ is not.
4. You are given a set of n types of rectangular 3-D boxes, where the ith box has height h_i, width w_i and depth d_i (all real numbers). You have to write a program to create a stack of boxes which is as tall as possible, but you can only stack a box on top of another box if the dimensions of the 2-D base of the lower box are each strictly larger than those of the 2-D base of the higher box. Of course, you can rotate a box so that any side functions as its base. It is also allowable to use multiple instances of the same type of box.
5. **Building Bridges.** Consider a 2-D map with a horizontal river passing through its center. There are n cities on the southern bank with x-coordinates $a_1 ... a_n$ and n cities on the northern bank with x-coordinates $b_1 ... b_n$. You have to write a program in C or C++ to connect as many north-south pairs of cities as possible with bridges such that no two bridges cross. When connecting cities, you are only allowed to connect the i^{th} city on the northern bank to the i^{th} city on the southern bank.