# Dynamic Programming

1. **Maximum Value Contiguous Subsequence**. Given a sequence of $n$ real numbers $A_1, A_2, \ldots, A_n$, determine a contiguous subsequence $A_i \ldots A_j$ for which the sum of elements in the subsequence is maximized.
2. **Making Change**. You are given n types of coin denominations of values $v_1 < v_2 < \cdots < v_n$ (all integers). Assume $v_1 = 1$, so you can always make change for any amount of money C. Give an algorithm which makes change for an amount of money C with as few coins as possible.
3. **The Integer Knapsack Problem (Duplicate Items Permitted)**. You have n types of items, where the ith item type has an integer size $s_i$ and a real value $v_i$. You are trying to fill a knapsack of total capacity C with a selection of items of maximum value. You can add multiple items of the same type to the knapsack.
4. **Box Stacking**. 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 want 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 a1 . . . an and n cities on the northern bank with x-coordinates b1 . . . bn. You want 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 ith city on the northern bank to the ith city on the southern bank.
6. **Balanced Partition**. You have a set of n integers each in the range 0 . . .K. Partition these integers into two subsets such that you minimize |S1 −S2|, where S1 and S2 denote the sums of the elements in each 1 of the two subsets.
7. **Edit Distance**. Given two text strings A of length n and B of length m, you want to transform A into B with a minimum number of operations of the following types: delete a character from A, insert a character into A, or change some character in A into a new character. The minimal number of such operations required to transform A into B is called the edit distance between A and B.