1. A ***vertex cover*** of a graph is a subset S of the vertices such that every vertex v either belongs to S or has a neighbor in S. In other words, the vertices in S cover all the edges. Finding the minimum size of a vertex cover is NP-hard, but in trees it can be found using dynamic programming.

   Given a tree T and non-negative weight $w(v)$ for each vertex v, describe an algorithm computing the minimum weight of a vertex cover of T.

2. Suppose you are given an unparenthesized mathematical expression containing n numbers, where the only operators are $+$ and $-$; for example:

   $$1+3-2-5+1-6+7$$

   You can change the value of the expression by adding parentheses in different positions. For example:

   $$1+3-2-5+1-6+7=-1$$
   $$(1+3-(2-5))+(1-6)+7=9$$
   $$(1+(3-2))-(5+1)-(6+7)=-17$$

   Design an algorithm that, given a list of integers separated by $+$ and $-$ signs, determines the maximum possible value the expression can take by adding parentheses.

   You can only insert parentheses immediately before and immediately after numbers; in particular, you are not allowed to insert implicit multiplication as in $1+3(-2)(-5)+1-6+7=33$.

3. Fix an arbitrary sequence $c_1 < c_2 < \cdots < c_k$ of coin values, all in cents. We have an infinite number of coins of each denomination. Describe a dynamic programming algorithm to determine, given an arbitrary non-negative integer x, the least number of coins whose total value is x. For simplicity, you may assume that $c_1 = 1$.

   To think about later after learning "greedy algorithms":

   (a) Describe a greedy algorithm to make change consisting of quarters, dimes, nickels, and pennies. Prove that your algorithm yields an optimal solution.

   (b) Suppose that the available coins have the values $c^0, c^1, \ldots, c^k$ for some integers $c > 1$ and $k \geq 1$. Show that the greedy algorithm always yields an optimal solution.

   (c) Describe a set of 4 coin values for which the greedy algorithm does ***not*** yield an optimal solution.