1. A *subsequence* is anything obtained from a sequence by extracting a subset of elements, but keeping them in the same order; the elements of the subsequence need not be contiguous in the original sequence. For example, the strings DOG, DAMN, ROAM, YAM, RAMMING, CRAMMING, and DYNAMICPROGRAMMING are all subsequences of the sequence DYNAMICPROGRAMMING.

   Let $X[1...m]$ and $Y[1...n]$ be two arrays. A common subsequence of X and Y is another sequence that is a subsequence of both X and Y. Describe an efficient algorithm to compute the length of the longest common subsequence of X and Y.

2. Let $X[1..m]$ and $Y[1..n]$ be two arrays. A common *supersequence* of X and Y is another sequence to which X and Y are both subsequences. Describe an efficient algorithm to compute the length of the shortest common supersequence of X and Y.

3. Given a graph $G = (V, E)$, a vertex cover of G is a subset $S \subseteq V$ of vertices such that, for each edge $e = uv$ in G, u or v is in S. That is, the vertices in S cover all the edges. It is known that finding the minimum size vertex cover is NP-Hard in general graphs but it can be solved in trees using dynamic programming.

   This is the goal of this problem. Given a tree $T = (V, E)$ and a non-negative weight $w(v)$ for each vertex $v \in V$, give an algorithm that computes the minimum weight vertex cover of T. It is helpful to root the tree.

4. You are given a rectangle piece of cloth with dimensions $X \times Y$, where X and Y are positive integers, and a list of n products that can be made using the cloth. For each product $i \in [1..n]$ you know that a rectangle of cloth of dimensions $a_i \times b_i$ is needed that the final selling price of the product is c_i. Assume the a_i, b_i, and c_i are positive integers. You have a machine that can cut any rectangular piece of cloth into two pieces either horizontally or vertically. Design an algorithm that determines the best return on the $X \times Y$ piece of cloth; that is, a strategy for cutting the cloth so that the products made from the resulting pieces give the maximum sum of selling prices. You are free to make as many copies of a given product as you wish, or none if desired.

5. Consider the problem of covering point by intervals. Specifically, assume that you are given a set P of n points on the real line and a set of m intervals F each specified by two points on the real line. Describe an algorithm to find the minimum-weight set of intervals that covers all the points of P.

   (As a bonus, find a simple greedy algorithm for the unweighted case and prove its correctness.)