1. Suppose you are handing out quizzes to a row of $n$ students. There are three versions of the quiz and you must hand them out in such a way that no two adjacent students have the same quiz. Furthermore, you know ahead of time how each student will score on each test. There are three sequences of values $A_1, \ldots, A_n$, $B_1, \ldots, B_n$, and $C_1, \ldots, C_n$ where $A_i, B_i, C_i$ corresponds to the score student $i$ will receive on quiz $A$, $B$, and $C$ respectively. Provide an algorithm to find the maximum total score of the students in this row.

2. You are asked to find a subset of a sequence of positive integers $a_1, a_2, \ldots, a_n$ under the condition that adjacent values may not both appear in your subset. That is, if you take value $a_i$, you may not include $a_{i-1}$ or $a_{i+1}$. Furthermore, you want to find the subset whose total value is as large as possible. For example, the sequence 1, 8, 7, 6, 5 has a maximum subsequence of value 14.

   (a) Consider the following greedy strategy. Look at the subsequence of odd terms $a_1, a_3, \ldots$ versus even terms $a_2, a_4, \ldots$ and take whichever one has larger total value. Show that this strategy will not always yield an optimal solution.

   (b) Another greedy strategy would be the following: repeatedly take the largest term whose neighbor’s have not already been taken. Show that this strategy will not always yield an optimal solution.

   (c) Using dynamic programming, give an efficient algorithm which will correctly solve this problem.

3. Problem 6.1 from the book. A contiguous subsequence of a list $S$ is a subsequence made up of consecutive elements of $S$. For instance, if $S$ is 5, 15, -30, 10, -5, 40, 10 then 15, -30, 10 is a contiguous subsequence but 5, 15, 40 is not. Give a linear-time algorithm for the following tasks:

   (a) Input: A list of numbers, $a_1, a_2, \ldots, a_n$.

   (b) Output: The contiguous subsequence of maximum sum.

   For the preceding example, the answer would be 10, -5, 40, 10, with a sum of 55.

   (Hint: Consider contiguous subsequences ending at a position $j$ for $j \in \{1, \ldots, n\}$, how would you use this as a subproblem?)

4. Problem 6.28 from the book Often two DNA sequences are significantly different, but contain regions that are very similar and are highly conserved. Design an algorithm that takes an input two strings $x[1\ldots n]$ and $y[1\ldots m]$ and a scoring matrix $\delta$ (as defined in Exercise 6.26) and outputs substrings $x'$ and $y'$ of $x$ and $y$, respectively, that have the highest-scoring alignment over all pairs of such substrings. Your algorithm should take time $O(mn)$.

5. Problem 6.14 from the book. You are given a rectangular 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 and that the final selling price of the product is $c_i$. Assume the $a_i$, $b_i$, and $c_i$ are all 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.

   (Hint: for each $1 \leq i \leq X$ and $1 \leq j \leq Y$, what would be the best return from a cloth of shape $i \times j$?)

6. Problem 6.7 from the book. A subsequence is palindromic if it is the same whether read left to right or right to left. For instance, the sequence $A, C, G, T, G, T, C, A, A, A, T, C, G$ has many palindromic subsequences. Devise an algorithm that takes a sequence $x[1\ldots n]$ and returns the (length of the) longest palindromic subsequence. Its running time should be $O(n^2)$.

   (Hint: for each position $j \in \{1\ldots n\}$, how many options do you have for the longest palindromic subsequence?)