CS314: Algorithms
Homework 6

Required Problems

1. Problem 1 from Chapter 6 of the text (on page 312-313)
   
   Note - this problem has a,b, and c, so don’t miss part c on the next page!

2. (Problem 1 from lecture note packet) Suppose you are given an array \( A[1..n] \) of integers.
   Describe and analyze an algorithm that finds the largest sum of elements in a contiguous subarray \( A[i..j] \).
   For example, if the array contains \((-6, 12, -7, 0, 14, -7, 5)\), the largest sum of contiguous entries is \(19 = 12 - 7 + 0 + 14\).

3. (Problem 2 from lecture notes, parts a and c) For these problems, we define a subsequence as anything that can be obtained from a sequence (or list of things) by extracting a subset of the elements but keeping them in the same order. For example, the strings C, YAIOAI, and DYNAMICPROGRAMMING are all subsequences of the string DYNAMICPROGRAMMING.

   (a) Let \( A[1..m] \) and \( B[1..n] \) be to arbitrary arrays. A common subsequence of \( A \) and \( B \) is another sequence that is a subsequence of both \( A \) and \( B \). Describe an efficient algorithm to compute the length of the longest common subsequence of \( A \) and \( B \).

   (b) Call a sequence of numbers \( X[1..n] \) oscillating if \( X[i] < X[i+1] \) for all even \( i \), and \( X[i] > X[i+1] \) for all odd \( i \). Describe an efficient algorithm to compute the length of the longest oscillating subsequence of an arbitrary array \( A \) of integers.