CSC-421 Applied Algorithms and Structures
Spring 2021-22

Instructor: Iyad Kanj
Office: CDM 832
Phone: (312) 362-5558
Email: ikanj@cs.depaul.edu
Office Hours (Office/Zoom): Monday 4:40-5:40 & Wednesday 1:00-3:00
Course Website: https://d2l.depaul.edu/

Assignment 3
(Due May 16)

1. (25 points) Pascal's triangle looks as follows:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
...
```

   The first entry in a row is 1 and the last entry is 1 (except for the first row which contains only 1), and every other entry in Pascal's triangle is equal to the sum of the following two entries: the entry that is in the previous row and the same column, and the entry that is in the previous row and previous column.

   (a) (10 points) Give a recursive definition (relation) for the entry $C[i, j]$ at row i and column j of Pascal's triangle. Make sure that you distinguish the base case(s).

   (b) (5 points) Give a recursive algorithm to compute $C[i, j], i \geq j \geq 1$. Illustrate by drawing a diagram (tree) the steps that your algorithm performs to compute $C[6, 4]$. Does your algorithm perform overlapping computations?

   (c) (10 points) Use dynamic programming to design an $O(n^2)$ time algorithm that computes the first n rows in Pascal's triangle.

2. (25 points) In a previous life, you worked as a cashier in the lost Antarctican colony, spending the better part of your day giving change to your customers. Because paper is a very rare and valuable resource in Antarctica, cashiers were required by law to use the fewest bills possible whenever they gave change.

   (a) (5 points) Suppose that the currency of the colony was available in the following denominations: 1, 4, and 6. Consider an algorithm that repeatedly takes the largest bill that does not exceed the target amount. For example, to make 11 using this algorithm, we first take a 6 bill, then a 4 bill, and finally a 1 bill. Give an example where this greedy algorithm uses more bills than the minimum possible.

   (b) (10 points) Describe and analyze a recursive algorithm that computes, given an integer n and an arbitrary system of k denominations $\langle d_1 = 1, \ldots, d_k \rangle$, the minimum number of bills needed to make the amount n.

   (c) (10 points) Describe a dynamic programming algorithm that computes, given an integer n and an arbitrary system of k denominations $\langle d_1 = 1, \ldots, d_k \rangle$, the minimum number of bills needed to make amount n.

3. (25 points) Suppose that you are given an array $A[1..n]$ of numbers, which may be positive, negative, or zero, and which are not necessarily integers. Give an $O(n)$-time algorithm that finds the largest sum of elements in a contiguous subarray $A[i..j]$ of A. For example, given the array $[-6, 12, -7, 0, 14, -7, 5]$ as input, your algorithm should return 19, which is the content of $A[2..5]$.

4. (25 points) A subsequence of a sequence 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 C, DAMN, YAIOAI, and DYNAMICPROGRAMMING are all subsequences of the string DYNAMICPROGRAMMING.

   (a) (15 points) Let $A[1..m]$ and $B[1..n]$ be two arbitrary arrays. A common subsequence of A and B is another sequence that is a subsequence of both A and B. A *longest common subsequence* of A and B is a subsequence of A and B of maximum length.

For example, if $A = \langle CTGCGTGTC \rangle$ and $B = \langle GTCGTGGC \rangle$, then the length of the longest common subsequence of A and B is 6, and the sequence $\langle TCGTGC \rangle$ is such a longest common subsequence of A and B.

Describe an $O(nm)$-time algorithm to compute the length of the longest common subsequence of two given sequences A and B. (**Hint.** You may modify the Edit Distance algorithm so that you consider only the two operations: Insert and Delete. What is the connection between this variant of Edit Distance and the problem of computing a longest common subsequence?)

(b) (10 points) A *palindrome* is any sequence that is exactly the same as its reversal, like I, or DEED, or RACECAR, or AMANAPLANACATACANALPANAMA. Use part (a) above to give an $O(n^2)$-time algorithm to find the length of the longest subsequence of a given sequence of length n that is also a palindrome.

5. **Suggested Programming Problems on LeetCode; not to be submitted.** Below are the titles of coding problems (related to the materials we covered so far) that I suggest that you do in LeetCode.com (under the tab "Problems"). You can test your submissions and view the solutions in LeetCode.

   (i) Longest Palindromic Substring

   (ii) Maximum Product Subarray

   (iii) Maximal Square