

## Assignment 03

# Dynamic Programming

Please submit your solutions in PDF format. The PDF must be typed, NOT handwritten. Solution for each problem must start on a new page. The solutions should be concise; complicated and half-witted solutions might receive low marks, even when they are correct. Solutions should be submitted on the course website.

### Problem 1: Collaborators

[2 points]

List the name of the collaborators for this assignment. If you did not collaborate with anyone, write "None" (without the quotes).

**Problem 2: Fair and Square**

[4 points]

You are given a bag with  $m$  coins of different denominations that sums up to a value, *total*. You need to divide the coins between two friends in such a way that both of them have a fair share, that means the difference between the amount these two gets should be minimum.

- (a) [2 points] What is the maximum amount a friend can get?
- (b) [2 points] What is the difference between the amount these two friends will get?

**Problem 3: Parenthesization? Nope!**

[11 points]

According to Wikipedia,  $C_n$  is the number of different ways  $n + 1$  factors can be completely parenthesized (or the number of ways of associating  $n$  applications of a binary operator). For  $n = 3$ , for example, we have the following five different parenthesizations of four factors:

- $((ab)c)d$
- $(a(bc))d$
- $(ab)(cd)$
- $a((bc)d)$
- $a(b(cd))$

We are interested in finding  $C_n$ , the  $n$ th Catalan number, which can be recursively defined as:

$$C_1 = 1$$
$$C_n = \sum_{i=1}^{n-1} C_i C_{n-i}$$

We can use the following code to find the  $n$ th Catalan Number:

```
int cat(int n)
{
    int c = 0;
    if(n == 1)
        c = 1;
    else
    {
        for(int i = 1; i < n; i++)
            c += cat(i) * cat(n - i);
    }
    return c;
}
```

For simplicity, we are only interested in the Catalan Numbers that can be stored in memory. But the algorithm can be generalized for larger values too.

- [4 points] Analyze the running time of the given code.
- [7 points] Describe the DAG to compute the first  $n$  Catalan Numbers using the recursive definition.

**Problem 4: Capsicum**

[23 points]

Arun Nahaj loves bell peppers, known locally as capsicums. That's why he planted  $n$  capsicum plants in a line at his garden. Due to his excitement, he forgot to keep enough space between the plants. After a while, the plants started growing capsicums. To be specific, the  $i$ th plant grew  $c_i$  capsicums. But the plants were not getting enough nutrition and became weak. Arun deduced that the reason might be the small distance between the consecutive plants. So he decided to uproot a few of the plants so that the others can grow properly. For each plant, Arun will either leave it as is or uproot the plant. In case he doesn't uproot the  $i$ th plant, he will uproot both the neighboring plants,  $(i - 1)$ , and  $(i + 1)$ . Since the capsicums have not matured properly, Arun wants to uproot the plants in such a way that the number of capsicums in the uprooted plants is minimized. Help him figure out which plants to uproot and which one to leave as is.