Assignment 11

Done By: Nayan Man Singh Pradhan

**Problem 11.1**

**Longest ordered subarray**

A .*cpp* file named “LOS.cpp” is created. Execute make to run.

**Problem 11.2**

**Sum in triangles**

**a.**

A .*cpp* file named “triangle.cpp” is created. Execute make to run. I have added a screenshot from the *.cpp* file that explains how the dynamic program runs.

A picture containing table

Description automatically generated

**b.**

My program:

A screenshot of a cell phone

Description automatically generated

I use two *“for”* loops in my program that iterates through all the nodes/elements (n) of the triangle. **Therefore, the runtime of my solution will be approximately .**

The brute force approach will create different arrays for all possible paths and pick the path that gives the largest sum. While this approach may be comparable with the dynamic programming approach for very small values of n, as n -> 10000 the number of arrays will increase drastically as every node (except the nodes in the bottom row) will have two solutions, causing the runtime to increase drastically too. The runtime of the brute force solution , where k is the number of rows. **Therefore, the runtime of the brute force solution is approximately , where k is the number of rows.**

**c.**

The greedy algorithm does not work for this problem as the greedy algorithm always picks the locally optimal solution. The greedy problem would start off by picking the largest number in the triangle from the top giving 7, 8, 1, 7, and finally 5, which equals to 28 (which is not the largest sum). The dynamic programming approach solves all the sub-problems and pricks the globally optimal solution, which makes it suitable for this problem, while the greedy algorithm picks the locally optimal solution, which might not lead to the correct solution, making the greedy unsuitable for this problem.