# ADS Homework 12

**Name -** Tewodros Adane
**Email -** tadane@jacobs-university.de

## Problem 12.1 – Longest ordered subarray

- The implementation and usage of an algorithm that uses dynamic programming to find the Largest ordered subarray is in the file “*LongestOrderedSubarray.cpp*”.

## Problem 12.2 – Sum in triangles

a) The implementation and usage of an algorithm that uses dynamic programming to find the Maximum sum on the path from the first line to the last line is in the file “*SumInTriangles.cpp*”.

b) Time complexity(Dynamic Programming):

For a triangle formed with **n** lines, each line will have **n** elements and the total number of elements contained is the triangle:

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$

When building the ‘`lis`’ vector the algorithm passes through each element in the triangle. Therefore, the time complexity is:

$$O\left(\frac{n(n+1)}{2}\right)$$

$$O(n^2)$$

Time complexity(Brute force):

Brute force approach for this problem is to generate all possible paths, sum the cost of each path and select the maximum cost path. For a triangle with **n** lines, there are $2^n$ possible path from the first line to the last line. Therefore, the time complexity of the brute force approach is:

$$O(2^n)$$

c) Proof by contradiction:

A greedy algorithm for this problem would be an algorithm that would start at the first line and at each step always chooses maximum of the left and right elements below it.

Given the following triangle :

```
        3
    2       1
4       0       9
```

The greedy algorithm would first choose 2 and then 4. And the path it returns is:

`3 -> 2 -> 4` and `cost = 9`

However, there is a path with larger cost than this:

`3 -> 1 -> 9` and `cost = 13`

Therefore, a greedy algorithm does not work for this problem.

□