

Problem 11.1 Longest ordered subarray - see DynProg1 folder for implementation.

Problem 11.2 Sum in triangles - see DynProg2 folder for implementation

B) analyse the runtime of your solution and compare it to the brute force approach

Brute force: at any number in the pyramid, there are 2 possible paths that can be taken. If we consider n to be the number of lines found on the pyramid, the number of elements in the pyramid is given by: the sum of numbers from 1 to n or: $\frac{n(n+1)}{2}$. Resulting in a time complexity of

$$2^{\frac{n(n+1)}{2}} = \Theta(2^n)$$

Dynamic Programming: The memoization loop iterates through every element in the pyramid, if we consider the number of lines to be n , then the number of elements is again: $\frac{n(n+1)}{2}$, resulting in a time complexity $\Theta(n^2)$. The algorithm for finding the path is only linear in time.

C) Explain why a greedy algorithm does not work for this problem:

To find the solution to this problem, we essentially need to consider all possible “paths” from the bottom of the triangle to the top in order to verify that we actually got a globally optimum solution. A greedy algorithm that would start from the top and pick the available maximum value on the level below would not consider all of the possibilities: using the sample test case as an example, the sample algorithm would pick $7 \rightarrow 8 \rightarrow 1 \rightarrow 7 \rightarrow 5 = 28$, which is not the globally optimum solution.