Problem 11.1

Longest\_Ordered\_Subarray.cpp

Problem 11.2

a)

Triangle\_Sum.cpp

b)

In my approach, every node on every line is resulted by either one comparison (if node is on the two end) or two comparisons. Therefore, the upper bound time complexity can be described as 2 times n, where n is the total number of nodes in the triangle.

Thus, if n is the level of the triangle, T(n) = O(2\*(1+n)\*n/2) = O(n^2+n) = O(n^2)

The brute force approach is to compare every possible combination. Because the triangle sum structure is binary-wise, the total number of combinations will be a 2 based number, depending on number of levels.

Thus, if n is the level of the triangle, T(n) = O(2^(n-1)) = O(2^n)

To sum up, the running time of my approach is quadratic whereas the brute force approach is exponential.

c)

If the greedy algorithm sticks with picking the greatest number in the next level, then the algorithm does not work for this problem. Because this triangle is binary-based, every node has only two choices of adding from the next level, there is no guarantee for these two choices to have the greatest number in that level.

Therefore, greedy algorithm can only produce the local optimal solution, but not a global one.