Question 1 by Dan Nguyen (z5206032)

Given a triangular grid, T, of non-negative integers with n rows where the i^{th} row has i entries. The j^{th} entry in row i is denoted $T\,(i, j)$.

A *route* is any path that starts at $T\,(1, 1)$ and travels down T through diagonal pathing i.e. the next node in the *route* from $T\,(i, j)$ is $T\,(i + 1, j - 1)$ or $T\,(i + 1, j)$.

Define *distance* to be the sum of all entries in the *route*.

Part A

$$\begin{array}{ccccc} & & 0 & & \\ & 0 & & 2 & \\ 10 & & 0 & & 3 \end{array}$$

Starting from $T\,(1, 1) = 0$, a greedy algorithm will choose $T\,(2, 2) = 2$ then choose $T\,(3, 3) = 3$. The greedy algorithm will determine the route to be $(T\,(2, 2),\ T\,(3, 3))$ and will determine the largest *distance* to be 5.

The correct *route* is $(T\,(2, 1),\ T\,(3, 1))$ which has the largest *distance* that is 10.

Part B

Define $Q(i, j)$ as the problem of finding the largest *distance* of the *route* $T\,[1..i, j]$ ending with $T\,(i, j)$.

Define $\text{opt}(i, j)$ as the solution to $Q(i, j)$.

For each $1 \leq i \leq n$ and $1 \leq j \leq i$, solve for $Q(i, j)$ using dynamic programming where the recurrence is:

$$\text{opt}(i, j) = \max\{\text{opt}(i - 1, j - 1) \mid 2 \leq j \leq i,\ \text{opt}(i - 1, j) \mid 1 \leq j < i\} + T\,(i, j)$$

The base case is $\text{opt}(1, 1) = T\,(1, 1)$. The order of solving Q is important i.e. subproblems with lesser i then lesser j are solved first.

The final answer is:

$$\text{opt}(n, i) = \max\{\text{opt}(n - 1, i - 1),\ \text{opt}(n - 1, i)\} + T\,(n, i)$$

There are $n + (n-1) + (n-2) + ... + 1 = n(n+1)/2$ subproblems which are solved in $O(n^2)$. The time complexity of solving each subproblem is constant. Therefore, the overall time complexity is $O(n^2)$ as required.