Joseph Lee

Module 3 Basic Programming Write-up

In the main, a matrix called dp[][], which stores the longest path at each respective pathsize from the input matrix, is initialized. In the main, it also read the input and initialize dp[][] with size of rows and columns in each []. After that there is a for loop that goes through each elevation tuple and set the outcome to be the max value between the previous result and from the current elevation. The helper function takes matrix dp[][], row x, column y and pathsize. Inside the longestPath, I create a pathlength that has initial values equal to 1 (since the minimum path length is always 1). After that it compare to see if the current elevation is greater than the surrounding ones. If the current elevation is greater than any of its surrounding, it finds the maximum value between all the smaller elevation's longest path + 1 and put the max value into the matrix dp[][] at the respective position. The outcome will be stored into List called results. The time complexity will be O(r*c) as each position in the input matrix and each elevation is processed.