Q2

Zijian Yue

Z5188675

The problem can be solved by solving the following subproblem.

Subproblem: When moving from a square to the other 2 squares always choose the square with the minimal elevation numbers.

Recursion function finds the minimum value of the elevation numbers:

MinEle(c,r) = min{MinEle(c+1,r) + 1 ,MinEle(c,r-1)}, if A[c][r] < A[c+1][r] && A[c][r] >= A[c][r-1]

= min{MinEle(c+1,r) ,MinEle(c,r-1) + 1}, if A[c][r] <= A[c+1][r] && A[c][r] > A[c][r-1]

= min{MinEle(c+1,r) + 1 ,MinEle(c,r-1)+1}, if A[c][r] <= A[c+1][r] && A[c][r] <= A[c][r-1]

BaseCase: MinEle(c,r) = 0 if c == C and r == 1,

MinEle(c,r) = INF if c > C or r < 1 ----> if (c,r) is out of boundary

Recursion function that finds the path according to the MinEle

From(i) = = args min{MinEle(c+1,r) + 1 ,MinEle(c,r-1)}, if A[c][r] < A[c+1][r] && A[c][r] >= A[c][r-1]

= args min{MinEle(c+1,r) ,MinEle(c,r-1) + 1}, if A[c][r] <= A[c+1][r] && A[c][r] > A[c][r-1]

= args min{MinEle(c+1,r) + 1 ,MinEle(c,r-1)+1}, if A[c][r] <= A[c+1][r] && A[c][r] <= A[c][r-1]

The args from the From(i) function will return the pair(c,r) which will obtain smallest elevation numbers from(1,R) to (1,C)

The TimeComplexity is O((R*C)^2), because there are about R*C/2 subproblems,and each of these subproblems are calling 2 subproblems which makes the overall time complexity ((R*C)/2)^2.