COMP3121 Homework Q2

Arth Sanskar Patel
z5228942

July 30, 2020

1 Answer

In this question we are given a grid of squares with each having a number associated with it, which represents the elevation of the terrain in that square. The over grid has dimensions R x C. We need to find the path with least possible moves from lower elevation to higher elevation from the top left corner (1, R) to the bottom right corner (C, 1) by only moving to the right or down from the current square we are in. This is similar to the tut 4 problem 12. The difference being that we need to count the number of times we go from a lower elevation to higher. The algorithm will work in this manner:

First we make a second matrix of size R x C and initialize all values in it to 0. Each cell in this matrix will store the count of elevation changes (only from lower to higher) which the path will take to reach that particular cell. To reach (C,1), it can only be achieved from the cells (C-1,1) or (C, 2). So we solve the sub problem of counting the minimum number of elevation changes needed to reach the cells (C-1,1) and (C,2). Lets call these number as EC_1 and EC_2 respectively. We now compare the elevations of (C-1,1) and (C,2) to the elevation of (C,1). For example let the elevation of (C,1) be 4, (C-1, 1) be 2 and (C,2) be 5. So moving from (C-1,1) to (C,1) will be 1 elevation change as $2 < 4$ There fore total elevation changes from source to (C,1) via (C-1,1) will be $EC_1 + 1$. Similarly when we move from (C,2) to (C,1) we see there is no elevation change as 5 ¿ 4 and hence the total elevation change in this case will be EC_2. We now compare $EC_1 + 1$ and EC_2 and chose the minimum of them to get the least possible number of elevation changes needed to reach (C,1).

Using the algorithm recursively will lets us get the minimum number of elevation changes need to reach any cell. For any cell (m,n) we will have:

$$mEC(m,n) = min\Big(\Big(mEC(m-1,n) + EC((m-1,n),(m,n))\Big), \Big(mEC(m,n+1) + EC((m,n+1),(m,n))\Big)\Big)$$

Here we are simply are calling the Minimum Elevation Change (mEC) function for the cells which are immediately above or to the left of our target cell and adding the Elevation Change (EC) to them and picking the minimum out of the two. The example of the algorithm is on next page. Please go through it to understand more.

Lets take an example grip of size 3 x 3 and see how the algorithm works. Here we need to go from (3,1) to (1,3). We

3	5	3	5
2	2	1	4
1	4	2	2
	1	2	3

can only move to (1,3) from (1,2) or (2,3). Calling the minimum elevation cost function will eventually take us to the cell (1,1) to start from. So lets fill the empty matrix from there. The base case or mCE(3,1) will be 0 as its the origin point. mCE(2,1) will be the mCE(3,1) + EC((2,1), (3,1)). As (3,1) is already higher than (2,1), it will be 0. Similarly for (3,2) the answer will be 0. Now from (2,1) you can go to (1,1) or (2,2). Going to (1,1) will cost us 1 EC as $4 > 2$. The total would be 1 as well because to reach 2, EC was 0. So we change the value of (1,1) to 1. Going to (2,2) can be done from (2,1) as well as (3,2). Since EC of both (2,1) and (3,2) is 0 and as $2 > 1$ and $3 > 1$ then the EC of (2,2) is 0. Now from (3,2) you can go to (3,3) or (2,2). As we already know the EC for (2,2) lets take for (3,3). As EC for (3,2) is 0 and $5 > 3$ then the total EC for (3,3) will be 1. Now from (1,1) we can go to (1,2) using 0 EC as $4 > 2$. The EC of (1,1) is 1 so the total EC for (1,2) will be 1. If we go to (1,2) from (2,2), the answer will still be 1 as $1 < 2$. Similarly doing forward for all the values will give us the best EC of 1 to reach (1,3) from (3,1). The final matrix will be as follows

3	0	0	1
2	0	0	1
1	1	1	1
	1	2	3

As the for the whole algorithm, we have to go through each cell once to find the best possible answer, the time complexity for this will be of $O(R * C)$. Reference for this question was taken from the Tut 4 question 12.