9101 Assignment 3
Haojin Guo
z5216214

**Q2.**

2. You are given a 2D map consisting of an $C \times R$ grid of squares; in each square there is a number representing the elevation of the terrain at that square. Find a path going from square $(1, R)$ which is the top left corner of the map to square $(C, 1)$ in the lower right corner which from every square goes only to the square immediately below or to the square immediately to the right so that the number of moves from lower elevation to higher elevation along such a path is as small as possible. (20 pts)

**Solution,**
1)
This solution is based on dynamic programming.
Let $dp[i][j]$ indicates the minimum number of moves from lower elevation to higher elevation at $(i, j)$.
Let $T[i][j]$ represents the elevation of the terrain at each square $(i, j)$.
2)
Base case:
initialize $dp[0][k]$ $and$ $dp[i][0]$
where, $k \in [0, R]$ $and$ $i \in [0, C]$
3) Recursion:
Assuming that we have solved all the subproblems $dp[k][m]$ $for\ k < I, m < j$,
Here, we have 4 cases according to the fact that we must go through $T[i][j-1]$ or $T[i-1][j]$ so as to $T[i][j]$.

a) CASE 1:
If $T[i][j] > T[i-1][j]$ $and\ T[i][j] > T[i][j-1]$, which means both directions will lead the number of moves from lower elevation to higher elevation plus by 1.
Hence,
$$dp[i][j] = \min(dp[i-1][j] + 1, dp[i][j-1] + 1)$$
b) CASE 2:
When, $T[i][j] > T[i-1][j]$ $and\ T[i][j] \leq T[i][j-1]$, only go down the moves will increase.
$$dp[i][j] = \min(dp[i-1][j] + 1, dp[i][j-1])$$
c) CASE 3:
When, $T[i][j] \leq T[i-1][j]$ $and\ T[i][j] > T[i][j-1]$, only go right the moves will increase.
$$dp[i][j] = \min(dp[i-1][j], dp[i][j-1] + 1)$$

d) CASE 4:
When, $T[i][j] \leq T[i-1][j]$ $and\ T[i][j] \leq T[i][j-1]$, both directions will

not lead the moves increased.

$$dp[i][j] = \min(dp[i-1][j], dp[i][j-1])$$

Therefore,

$$dp[i][j] = \begin{cases} \min(dp[i-1][j]+1, dp[i][j-1]+1) & \text{if } T[i][j] > T[i-1][j] \;\; and\; T[i][j] > T[i][j-1] \\ \min(dp[i-1][j]+1, dp[i][j-1]) & if\;\; T[i][j] > T[i-1][j] \;\; and\; T[i][j] \leq T[i][j-1] \\ \min(dp[i-1][j], dp[i][j-1]+1) & if\;\; T[i][j] \leq T[i-1][j] \;\; and\; T[i][j] > T[i][j-1] \\ \min(dp[i-1][j], dp[i][j-1]) & if\; T[i][j] \leq T[i-1][j] \;\; and\; T[i][j] \leq T[i][j-1] \end{cases}$$

4)

The result is $dp[C][1]$.

5) Time complexity:

The total time complexity is O(R*C), because it is need to traverse a two-dimensional array.