z5242692

Chenqu Zhao

COMP9101 (T2-2020)

Homework 4 – Q2

Use a 2D array Q to store the minimum number of moves from lower elevation to higher elevation for each square in the 2D map. Therefore, we have the following subproblem: find the minimum number of moves f from lower elevation to higher elevation when reaching square (x, y). Assume A and B are coordinates of two adjacent squares, and B is the next step of A. Define a sub-function $gap(A, B) = \begin{cases} f(A) & \text{if the elevation of A} \geq \text{the elevation of B} \\ f(A) + 1 & \text{if the elevation of A} < \text{the elevation of B} \end{cases}$

Since each step of the path goes either below or right to the current square, we can set up the state transfer equation as follow:

$$f(x, y) = min\{gap((x-1, y), (x, y)), gap((x, y+1), (x, y))\}$$

The base case $f(1, R) = 0$. Traverse the 2D map row by row to calculate the minimum number of moves from lower elevation to higher elevation using the transfer equation and fill in the 2D array Q with these values.

After filling up the 2D array Q, we can apply back-tracking technique to generate an exact path. Initialize a list P with a single element square $(C, 1)$. Start from the end point square $(C, 1)$, move to the left or the top step by step if the value f of that square is not larger than the value f of the current square. Append the square to list P. Keep moving until reaching the start point square $(1, R)$.

The **reverse list** of list P is one of the paths we want.