Given: 2D map consisting of an R X C grid of squares; Starting point S: (1,R), Ending point E: (C,1).
Aim: To find a path as small as possible between two points.

Solution:

In order to get the smallest path. We should solve the following sub-problem: reversely track the path from ending point to starting point by choosing the previous point of ending point and use P to represent it. The points can be choose will only from either immediately left adjacent point PL or immediately top adjacent point PT until we arrive the starting point, depends on which distance is smaller, pick the smaller point as a point on the goal path and set this point as P, then keep iterating until we reach the starting point.

Base case: A pathList which store the point we will go and then a total distance TD. Now initial the pathList=[]; TD = 0; point P = point E(C,1). Now find the value V1 of Distance(from (C-1,1) to P) and the value V2 of Distance(from (C,1+1) to P). Choose the smaller one between V1 and V2. TD = TD + smaller value between V1 and V2. Add the distance that have smaller point to the pathList. Update the point P, P = the point that has less distance to P. Opt(i,j) = Opt(i,j) + Distance(from (C-1,1) to P)

Sub problems: Solve the problem by moving point P as much close to Starting point as possible by implementing the method used as base case until we reach. The smallest path will be the reverse ordered pathList.