Given a 2d matrix with cost in each cell, find minimum cost in traversing from top-left corner to a given cell. Each cell of the matrix represents a cost to traverse through that cell.

<https://www.geeksforgeeks.org/min-cost-path-dp-6/>

You can only traverse down, right and diagonally lower cells from a given cell, i.e., from a given cell (i, j), cells (i+1, j), (i, j+1), and (i+1, j+1) can be traversed.

Min cost from 0,0 -> 2,2

{{1, 2, 3}

{4, 8, 2}

{1, 5, 3}} min cost = 1+2+2+3

So the idea is that to reach (m,n), the possible ways are (m, n-1), (m-1, n-1), (m-1, n)

So the min cost of reach the cell (m, n) will be min of above 3 cells + cost(m, n)

If(m ==0 && n==0 ) //first cell

Return cost[0][0]

If(m<0 || n<0) //cell beyond reach

Return INT\_MAX

Cost(m, n) = min(cost(m, n-1), cost (m-1, n-1), cost (m-1, n)).

Discussions:

<https://www.geeksforgeeks.org/find-the-maximum-cost-path-from-the-bottom-left-corner-to-the-top-right-corner/>

<https://www.geeksforgeeks.org/minimum-cost-path-left-right-bottom-moves-allowed/>

Kosaraju’s algorithm:

To check if graph is strongly connected.

A graph is said to be strongly connected if there is a edge from u->v for all vertices.

Algo for finding if a graph is strongly connected for an undirected graph is simple: Use BFS/DFS from any vertex. If all vertices are visited, its connected. For directed graph its not possible bcoz , consider below scenario:

0 -> 1 -> 2 -> 3

Here all nodes are visited but they are not connected.

Hence, use below method to determine strongly connected graph for directed graph.

1. all vertices with non-zero degree belong to single strongly connected component.

2. out degree of every vertex is equal to in degree. i.e. number of edges starting from a node should be equal to number of edges entering the node.

Kosaraju algo using below method:

1. Run DFS for a graph from any node. All nodes should be visited.
2. Transpose graph.
3. Run DFS for a graph from same node. All nodes should be visited.