Path Count Problem
(Total Number Of Ways Reaching To A Particular Cell In A Matrix)

Problem Statement:

Given a ***2-D MATRIX of ANY SIZE (m x n)*** and some ***MOVEMENT CONSTRAINTS***, we need to find out the ***TOTAL NUMBER of POSSIBLE PATHS/WAYS*** to reach the ***BOTTOM RIGHT CELL*** starting from the ***TOP LEFT CELL*** in the matrix under the given ***MOVEMENT CONSTRAINTS.***

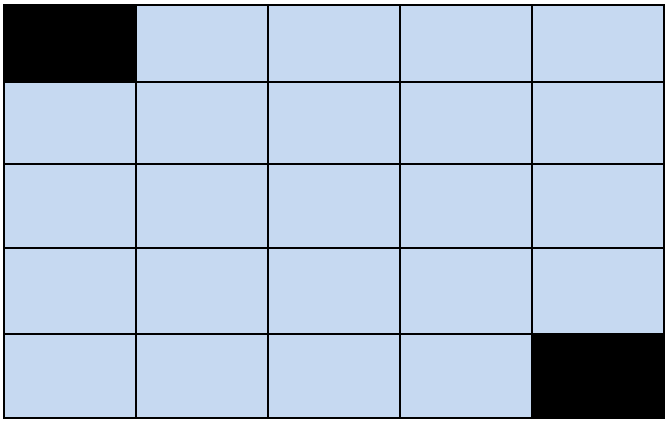
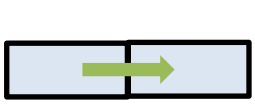


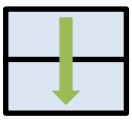
Figure: Problem Statement of Path Count Problem

Movement Constraints:

From a cell, you can

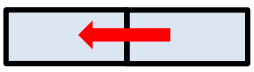
- ***move to the right***
- ***move downwards***

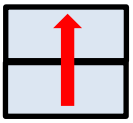


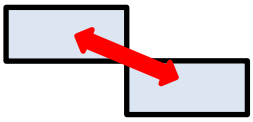


From a cell, you can't

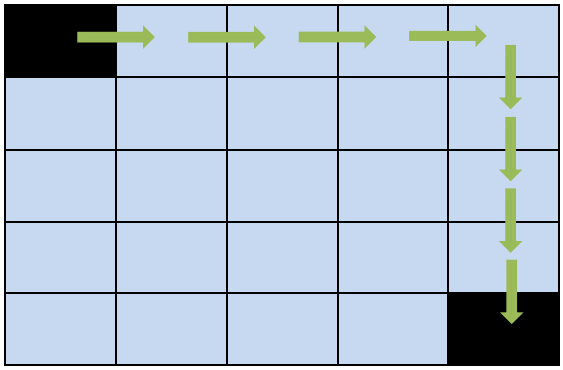
- ***move to the left***
- ***move upwards***
- ***move diagonally (neither diagonally up nor diagonally down)***

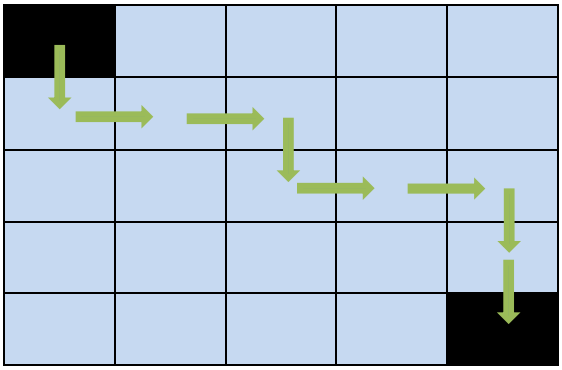


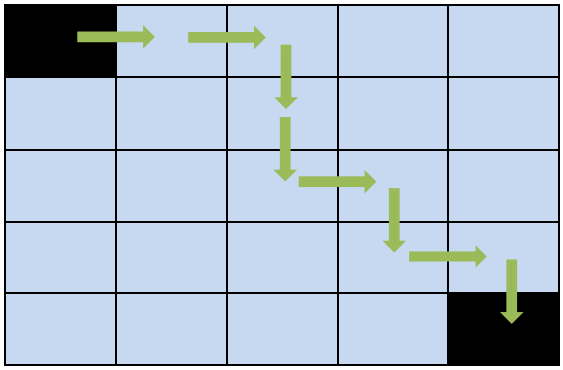


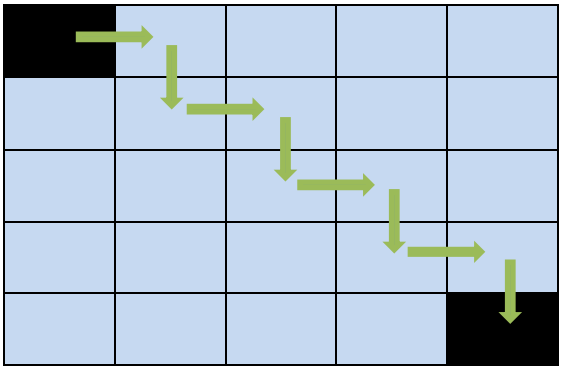


Allowed/Valid Path/Way Solutions:





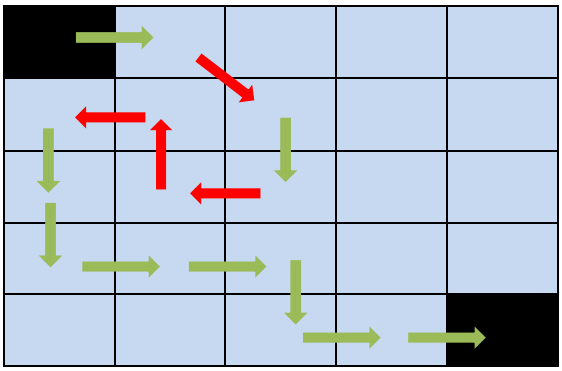




.....
.....

.....
.....

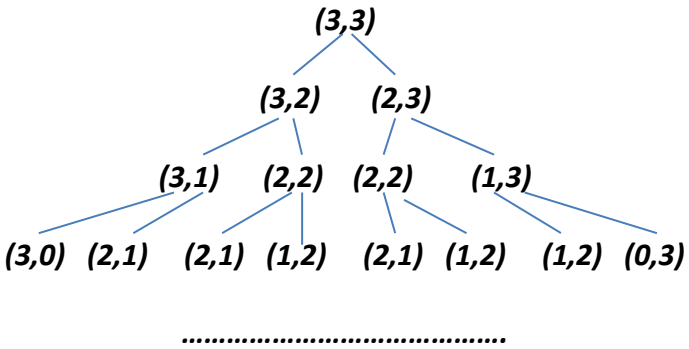
Invalid/Not Allowed Path/Way Solutions:



Implementing the Path Count Problem using Dynamic Programming

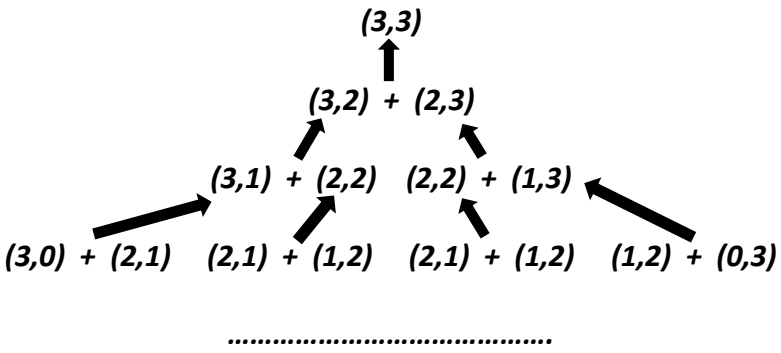
Overlapping Sub-problems Property:

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3)
(2,0)	(2,1)	(2,2)	(2,3)
(3,0)	(3,1)	(3,2)	(3,3)



Optimal Substructure Property:

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3)
(2,0)	(2,1)	(2,2)	(2,3)
(3,0)	(3,1)	(3,2)	(3,3)



Worked Out Example of Path Count Problem Using Dynamic Programming

	0	1	2	3	4	5	6	7
0	1	1	1	1	1	1	1	1
1	1	2	3	4	5	6	7	8
2	1	3	6	10	15	21	28	36
3	1	4	10	20	35	56	84	120
4	1	5	15	35	70	126	210	330
5	1	6	21	56	126	252	462	792
6	1	7	28	84	210	462	924	1716