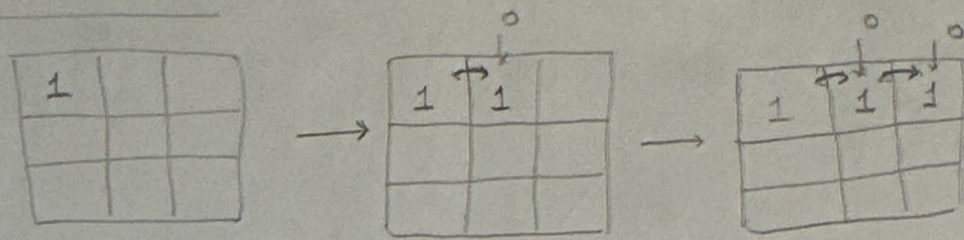
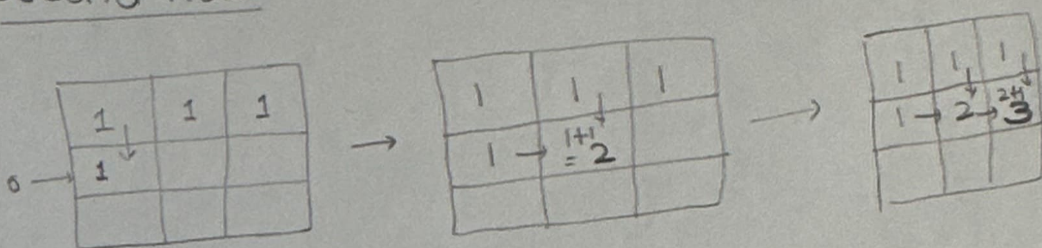
Consider 3 x 3 Matrix

Initially →

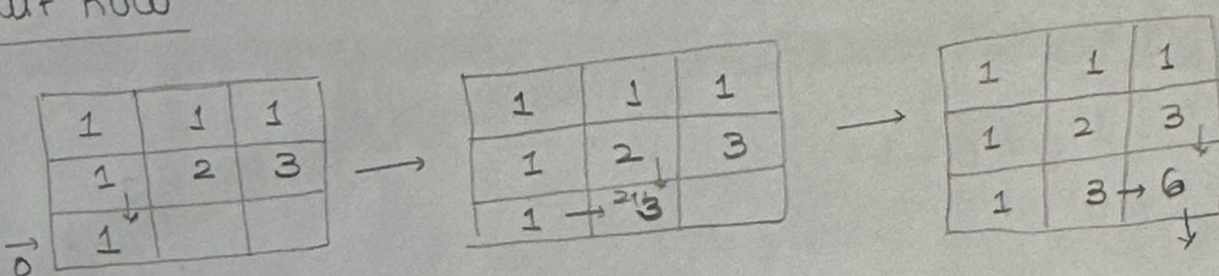


• Each cell shows, no of ways to reach (0,0) from that cell If we Only move Up & Left

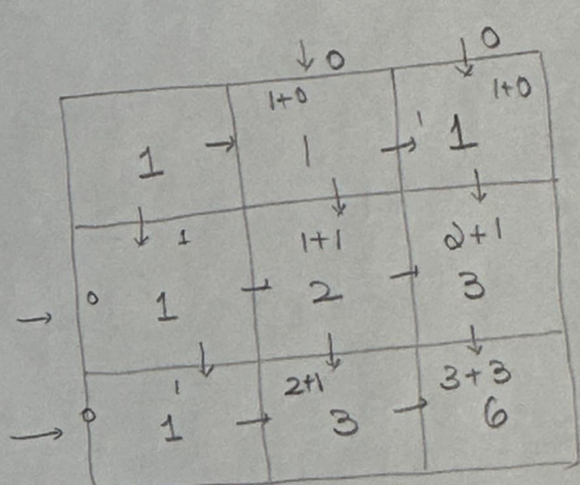
Second Row



Last Row



This says that there are 6 ways, starting from (n x n) to reach (0 x 0)



Space Optimization

* Two Observations to make:

  1. We Only Look at top & Left Value
  2. The first row & first col will always be 1

So Initial dp =

1	1	1

We dynamically update the dp, So value at Each position will be latest Value

So starting from row 1, col 1 → we know,

	j	
1	1	1

* to see up we see j value
* to see left we see $j-1$ Value

reference dp

1	1	1
1	we are here	
1		

because $j-1$ would have been calculated in prev step.

so $dp[j]$ = up + left = $dp[j] + dp[j-1] = 2$

for row 1, col 2

			j
dp	1	2	1

$dp[j] = dp[j] + dp[j-1]$

$= 2 + 1 = 3$

reference dp

1	1	1
1	2	

Next for row 2, col 2

j

dp =

1	2	3

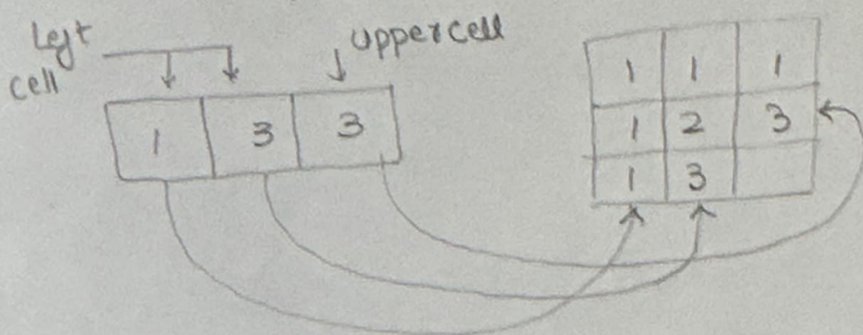
reference dp

1	1	1
1	2	3
1		

$dp[j] = dp[j] + dp[j-1]$

$= 2 + 1 = 3$

so current if we map 1D to 2D dp



Next for row 2, col 2

dp =

1	3	3

$dp[j] = dp[j] + dp[j-1] = 3 + 3 = 6$

dp =

1	3	6

→ 6 ways to reach top