

## 64. Minimum Path

1	3	1
1	5	1
4	2	1

→ den, abajo

→ input vector <vector<int>>, grid

→ dp

dp

①	3	1
1	5	1
4	2	1

grid

1	0	0
0	0	0
0	0	0

→ inicializar dp con el mismo valor que grid

↑

1	③	1
1	5	1
4	2	1

1	4	0
0	0	0
0	0	0

→ caso, no hay valor arriba

dp[i][j] solo tomaría el valor de grid[i][j] más el dp i-1, dp[i][j-1]

1	3	①
1	5	1
4	2	1

1	4	5
0	0	0
0	0	0

1	3	1
①	5	1
4	2	1

1	4	5
2	0	0
0	0	0

→ caso, no hay valor i-1

dp[i][j] solo tomaría el valor de grid[i][j] más el valor dp arriba dp[i-1][j]

1	3	1
1	⑤	1
4	2	1

1	4	5
2	7	0
0	0	0

→ caso, si hay valor arriba y abajo

comparar dp[i-1][j] con dp[i][j-1]

& dp[i-1][j] > dp[i][j-1]

dp[i][j] = grid[i][j] + dp[i-1][j]

1	3	1
1	5	⑦
4	2	1

1	4	5
2	7	6
0	0	0

caso contrario

dp[i][j] = grid[i][j] + dp[i][j-1]

1	3	1
1	5	1
4	2	1

1	4	5
2	7	6
6	0	0

1	3	1
1	5	1
4	2	①

1	4	5
2	7	6
6	8	0

→

1	3	1
1	5	1
4	2	①

1	4	5
2	7	6
6	8	⑦

→ se retorna dp[m-1][n-1]