DP Concepts & Questions

Video 33

The future depends on what you do today.

Believe you can, and you're halfway there

MIK...

भाषण (Motivation)

Done
- 1-D based DP
- Grid based DP

Done
- String based DP
- Digit DP
- Game Strategy

We'll do :-

(·) RECURSION + MEMOIZATION (Top Down)

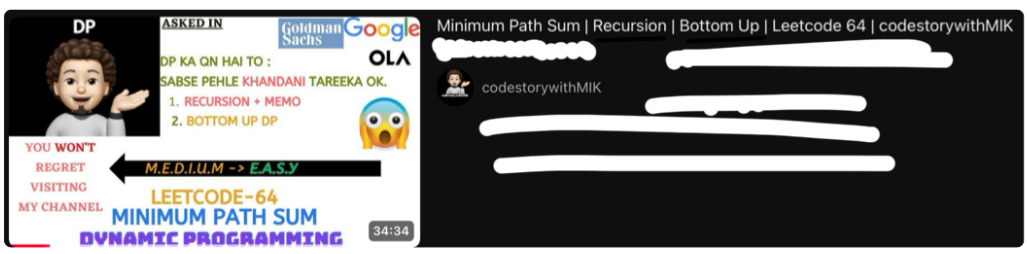
(·) Bottom UP

(·) Time & Space

DP on Grids

Solve this on your own first :-

Leetcode - 64 "Minimum Path Sum"



When you solve it, then only come to this video. ☺

1594. Maximum Non Negative Product in a Matrix

Medium | Topics | Companies | Hint

You are given a `m x n` matrix `grid`. Initially, you are located at the top-left corner `(0, 0)`, and in each step, you can only **move right or down** in the matrix.

Among all possible paths starting from the top-left corner `(0, 0)` and ending in the bottom-right corner `(m - 1, n - 1)`, find the path with the **maximum non-negative product**. The product of a path is the product of all integers in the grid cells visited along the path.

Return the *maximum non-negative product* ***modulo*** `10^9 + 7`. *If the maximum product is* ***negative****, return* `-1`.

Notice that the modulo is performed after getting the maximum product.

(0,0) →
↓

(m-1, n-1)

Example :-

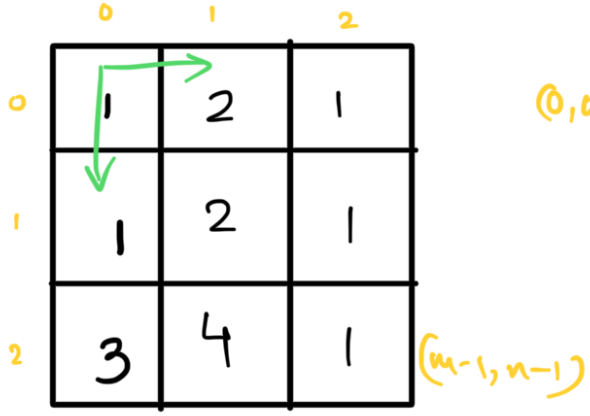
1	-2	1
1	-2	1
3	-4	1

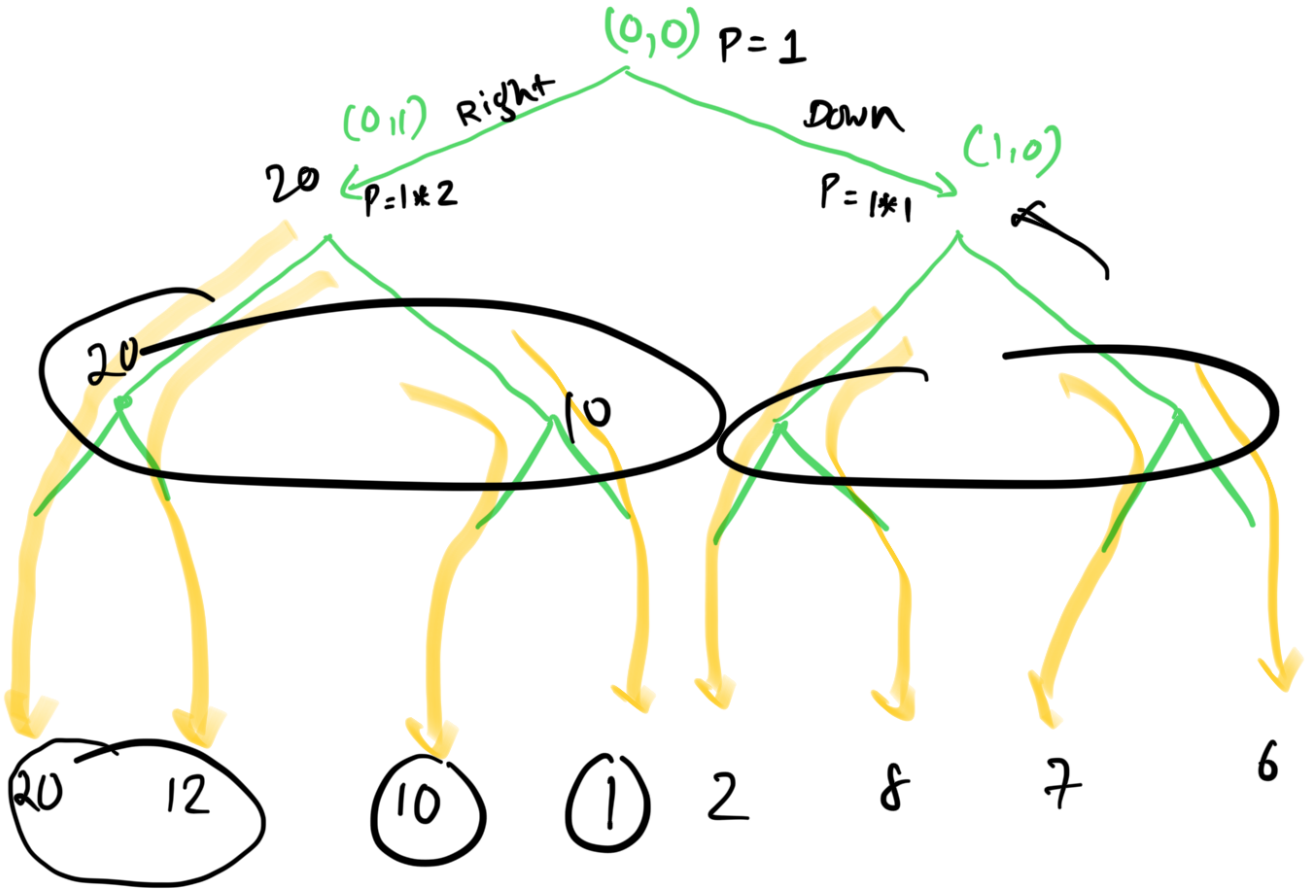
1 * 1 * (-2) * (-4) * 1 = 8

Output :- 8

Thought Process

+ve numbers.





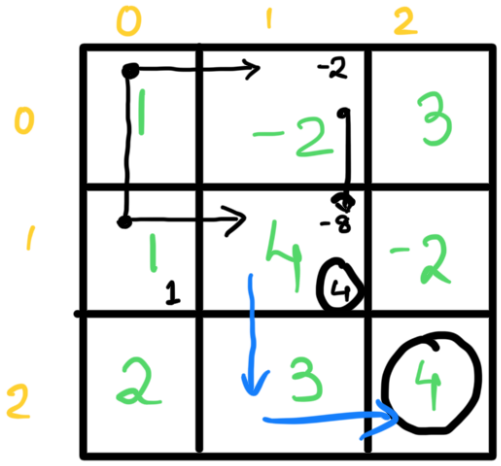

```
if (m-1 && n-1) {
    return grid[i][j];
}
```

right = grid[i][j] * solve(grid,i,j+

```
down = grid[i][j] * Solve(grid, i+1, j);
```

return max(right, down) ;

Grid contains -ve nos also :-



12 = -96

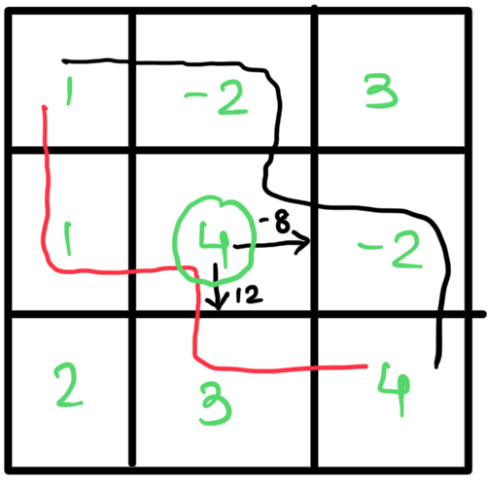
Right

4 → -2*4 = -8

12*4 = 48

Down

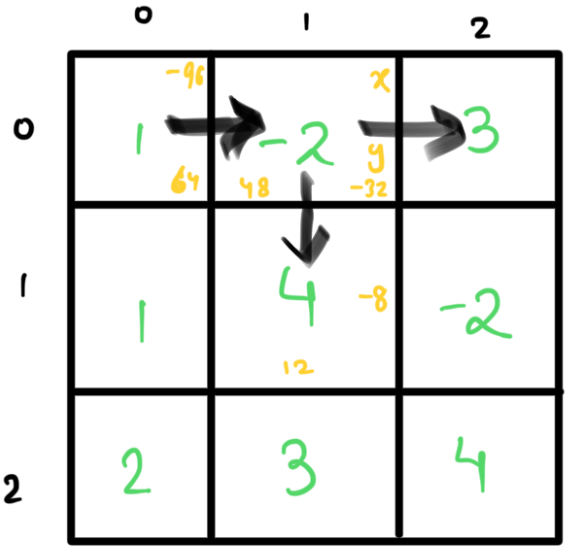
3*4 = 12



$-32 * -2 * 1$

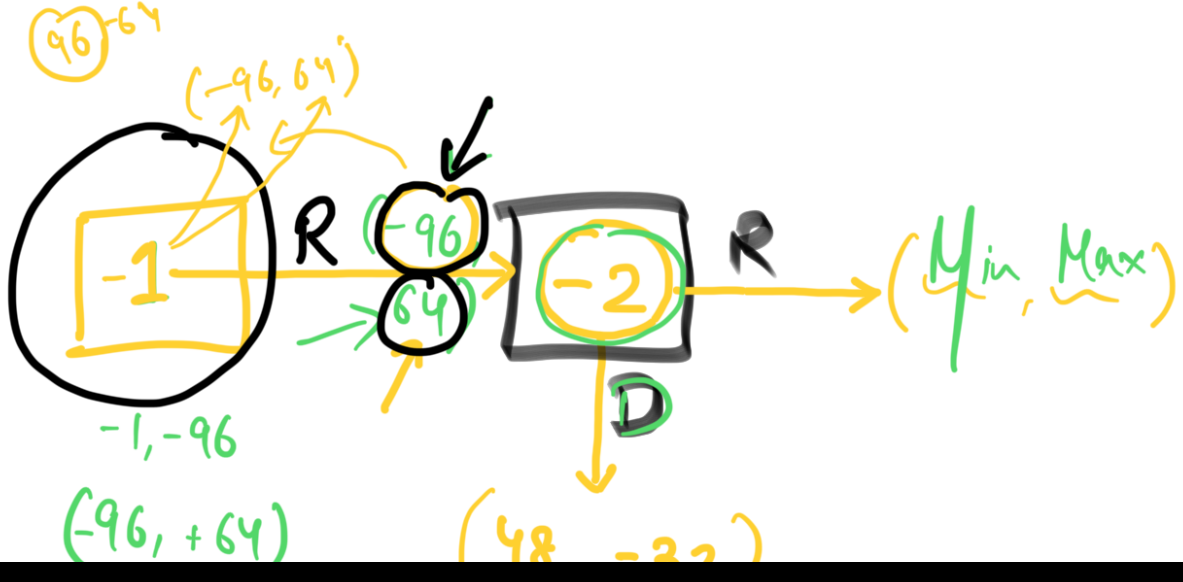
$= +64$

$12 \leftarrow \max(-8, 12)$ R, D ✗



$1 * -96 = -96$

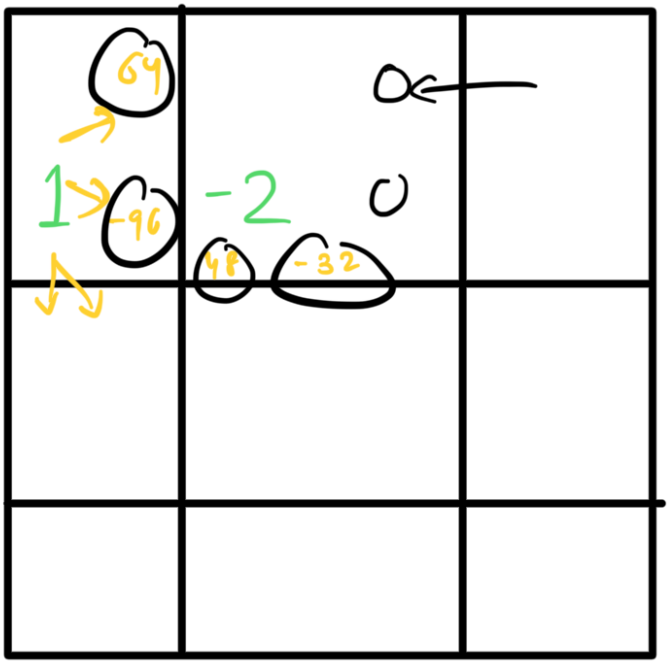
$1 * 64 = 64$



Every cell must have two information from

Right → (rightMax, rightMin)

Down → (downMax, downMin)



-96 , 64

maxVal = 64

minVal = -96

downMax = 48

downMin = -32

grid[i][j]

maxVal = max(maxVal, -2 * downMax, -2*downMin);

minVal = min(minVal, -2*downMax, -2*downMin);

rightMax = x

rightMin = y

maxVal = max(maxVal, -2 * rightMax, -2 * rightMin);

minVal = min(minVal, -2*rightmax, -2* ridmin);

return {maxVal, minVal};

if(maxVal < 0) -1

maxVal=

-1 * 2 * 1 = -2

-1 * 3 * 1 = -3

(•) vector<vector<pair<ll, ll>>> t

Bottom Up

Solve (i, j) ;

t[i][j] = pair<ll, ll>

State Definition :- (0,0) ⟶ (m-1, n-1)

t[i][j] = { max_product / min_product from (0,0) to (i,j) , } ;

1	2	3
1	-4	-2
2	3	4

	0	1	2
0	{1, 1}	{2, 2}	{6, 6}
1	{1, 1}	?	
2	{2, 2}		

t

dp[0][0] = ???

Base Case:- 0th row

0th col