

# 063. Unique Paths II

## 063 Unique Paths II

- Dynamic Programming+Array

### Description

Follow up for "Unique Paths":

Now consider if some obstacles are added to the grids. How many unique paths would there be?

An obstacle and empty space is marked as **1** and **0** respectively in the grid.

For example,

There is one obstacle in the middle of a 3x3 grid as illustrated below.

```
[
  [0,0,0],
  [0,1,0],
  [0,0,0]
]
```

The total number of unique paths is **2**.

**Note:**  $m$  and  $n$  will be at most 100.

### 1. Thought line

### 2. Dynamic Programming+Array

```
1 class Solution {
2 public:
3     int uniquePathsWithObstacles(vector<vector<int>>& obstacleGrid) {
4         if (obstacleGrid.empty()) return 0;
5         int m = obstacleGrid.size(), n = obstacleGrid[0].size();
6         int Possibility [m][n] = {0};
7
8         // initiate Possibility array
9         bool flagi = false, flagj = false;
10        for (int i= 0; i<=m-1; ++i){
11            if (obstacleGrid[i][0]==1 || flagi){
12                flagi = true;
13                Possibility[i][0] = 0;
14            }
15            else if (!flagi)
16                Possibility[i][0] = 1;
17        }
18
19        for (int j= 0; j<=n-1; ++j){
20            if (obstacleGrid[0][j]==1 || flagj){
21                flagj = true;
22                Possibility[0][j] = 0;
23            }
24            else if (!flagj)
25                Possibility[0][j] = 1;
26        }
27
28        // dp
29        for (int i=1; i<=m-1; ++i){
30            for (int j=1; j<=n-1; ++j){
```

```
31         if (obstacleGrid[i][j] != 1)
32             Possibility[i][j] = Possibility[i-1][j] + Possibility[i][j-1];
33     }
34 }
35 }
36 return Possibility[m-1][n-1];
37 }
38 }
39 };
```