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

```
class Solution {
public:
    int uniquePathsWithObstacles(vector<vector<int>>& obstacleGrid) {
        if (obstacleGrid.empty()) return 0;
        int m = obstacleGrid.size(), n = obstacleGrid[0].size();
        int Possibility [m][n] = {0};

        // initiate Possibility array
        bool flagi = false, flagj = false;
        for (int i= 0; i<=m-1; ++i){
            if (obstacleGrid[i][0]==1 || flagi){
                flagi = true;
                Possibility[i][0] = 0;
            }
            else if(!flagi)
                Possibility[i][0] = 1;
        }

        for (int j= 0; j<=n-1; ++j){
            if (obstacleGrid[0][j]==1 || flagj){
                flagj = true;
                Possibility[0][j] = 0;
            }
            else if (!flagj)
```

```
            Possibility[0][j] = 1;
        }

        // dp
        for (int i=1; i<=m-1; ++i){
            for (int j=1; j<=n-1; ++j){
                if (obstacleGrid[i][j]!=1)
                    Possibility[i][j] = Possibility[i-1][j]+Possibility[i][j-1];
            }

        }
        return Possibility[m-1][n-1];

    }
};
```