

SuccessDetails

Runtime: 61 ms, faster than 39.96% of Java online submissions for Longest Increasing Subsequence.

Memory Usage: 39 MB, less than 41.66% of Java online submissions for Longest Increasing Subsequence.

Next challenges:

Increasing Triplet Subsequence

Russian Doll Envelopes

Maximum Length of Pair Chain

Number of Longest Increasing Subsequence

Minimum ASCII Delete Sum for Two Strings

Minimum Number of Removals to Make Mountain Array

Find the Longest Valid Obstacle Course at Each Position

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
12/12/2021 02:25	Accepted	61 ms	39 MB	java
12/12/2021 02:04	Wrong Answer	N/A	N/A	java

JavaAutocomplete

```
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
for(int i = 1; i < dp.length; i++){
    int longM = 0;
    for(int j = 0; j < i; j++){
        if(nums[j] < nums[i] && dp[j] > longM){
            longM = dp[j];
        }
        dp[i] = longM + 1;
        if(dp[i] > longest){
            longest = dp[i];
        }
    }
    return longest;
}
```

Your previous code was restored from your local storage. Reset to default

TestcaseRun Code ResultDebugger

AcceptedRuntime: 0 ms

Your input[10,9,2,5,3,7,101,18]

Output4Diff

Expected4

SuccessDetails

Runtime: 0 ms, faster than 100.00% of Java online submissions for Unique Paths II.

Memory Usage: 38.3 MB, less than 66.10% of Java online submissions for Unique Paths II.

Next challenges:

Unique Paths

Unique Paths III

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
12/12/2021 12:49	Accepted	0 ms	38.3 MB	java
12/12/2021 12:47	Compile Error	N/A	N/A	java
12/12/2021 12:47	Compile Error	N/A	N/A	java
12/11/2021 11:21	Accepted	1 ms	38.9 MB	java
12/11/2021 10:18	Wrong Answer	N/A	N/A	java
12/11/2021 12:44	Accepted	0 ms	37.7 MB	java

JavaAutocomplete

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
class Solution {
    public int uniquePathsWithObstacles(int[][] obstacleGrid) {
        int m = obstacleGrid.length;
        int n = obstacleGrid[0].length;
        int[][] rpta = new int[m][n];
        rpta[0][0] = obstacleGrid[0][0] == 0 ? 1 : 0;
        if (n <= 100 && m >= 1) {
            if (rpta[0][0] == 0) return 0;
            for (int i = 0; i < m; i++) {
                for (int j = 0; j < n; j++) {
                    if (obstacleGrid[i][j] == 1)
                        rpta[i][j] = 0;
                    else if (i == 0 || j == 0) {
                        if (j > 0)
                            rpta[i][j] = rpta[i][j-1];
                        else if (i > 0)
                            rpta[i][j] = rpta[i-1][j];
                    } else
                        rpta[i][j] = rpta[i-1][j] + rpta[i][j-1];
                }
            }
        }
    }
}
```

Your previous code was restored from your local storage. Reset to default

TestcaseRun Code ResultDebugger

AcceptedRuntime: 0 ms

Your input[[0,0,0],[0,1,0],[0,0,0]]

Output2Diff

Expected2

SuccessDetails

Runtime: 4 ms, faster than 90.36% of Java online submissions for Maximal Square.

Memory Usage: 42.5 MB, less than 46.60% of Java online submissions for Maximal Square.

Next challenges:

Maximal Rectangle

Largest Plus Sign

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
12/12/2021 14:00	Accepted	4 ms	42.5 MB	java
12/12/2021 13:50	Wrong Answer	N/A	N/A	java

JavaAutocomplete

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
class Solution {
    public int maximalSquare(char[][] matrix) {
        int mayon = 0;
        if (matrix != null && matrix.length > 0) {
            int f = matrix.length; //file
            int c = matrix[0].length; //column
            int rpta[][] = new int[f][c];
            for (int i = 0; i < f; i++) {
                for (int j = 0; j < c; j++) {
                    if (i == 0 || j == 0)
                        rpta[i][j] = matrix[i][j] == '1' ? 1 : 0;
                    else if (matrix[i][j] == '1')
                        rpta[i][j] = 1;
                    else {
                        rpta[i][j] = Math.min(rpta[i][j-1], Math.min(rpta[i-1][j], rpta[i-1][j-1]))+1;
                    }
                    if (mayon < rpta[i][j]) mayon = rpta[i][j];
                }
            }
        }
    }
}
```

Your previous code was restored from your local storage. Reset to default

TestcaseRun Code ResultDebugger

AcceptedRuntime: 0 ms

Your input[[["1","0","1","0","0","0"],["1","0","1","1","1","1"],["1","1","1","1","1","1"],["1","0","0","1","1","0"]]]

Output4Diff

Expected4