



```

1 public:
2     int
3     uniquePathsWithObstacles(vector<vector<int>>& obstacleGrid) {
4         int m = obstacleGrid.size();
5         if (m == 0) {
6             return 0;
7         }
8         int n =
9         obstacleGrid[0].size();
10        if (n == 0) {
11            return 0;
12        }
13        vector<vector<int>> cache(m,
14        vector<int>(n, -1));
15        return dfs(obstacleGrid,
16        cache, 0, 0, m, n);
17    }
18
19    int dfs(vector<vector<int>>&
20    obstacleGrid, vector<vector<int>>&
21    cache, int i, int j, int m, int n){
22        if (obstacleGrid[i][j] == 1)
23        {
24            return 0;
25        }
26        if (i + 1 == m && j + 1 ==
27        n) {
28            return 1;
29        }
30        if (cache[i][j] != -1) {
31            return cache[i][j];
32        }
33        int paths = 0;
34        if (i + 1 < m) {
35            paths +=
36            dfs(obstacleGrid, cache, i + 1, j,
37            m, n);
38        }
39        if (j + 1 < n) {
40            paths +=
41            dfs(obstacleGrid, cache, i, j + 1,
42            m, n);
43        }
44    }
45 }

```