

Problem List < > ⌂ Submit Premium

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 66 / 66 testcases passed

Siddhant0705 submitted at Feb 22, 2026 22:42

Runtime 0 ms | Beats 100.00% Memory 15.07 MB | Beats 99.86%

Analyze Complexity

Runtime distribution chart showing 75% of execution time in 1ms, 25% in 3ms, and small portions in 2ms, 4ms, and 5ms.

Code C++

```
</> Code C++ Auto
1 class Solution {
2 public:
3     int minPathSum(vector<vector<int>>& grid) {
4         int m = grid.size();
5         int n = grid[0].size();
6
7         for(int i = 1; i < m; i++) {
8             grid[i][0] += grid[i-1][0];
9
10            for(int j = 1; j < n; j++) {
11                grid[0][j] += grid[0][j-1];
12
13                for(int i = 1; i < m; i++) {
14                    for(int j = 1; j < n; j++) {
15                        grid[i][j] += min(grid[i-1][j], grid[i][j-1]);
16
17                    }
18                }
19
20            return grid[m-1][n-1];
21        }
22    };
}
```

Saved Ln 22, Col 3

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

```
1 class Solution {
2 public:
3     int minPathSum(vector<vector<int>> &grid) {
4
5         int m = grid.size();
6         int n = grid[0].size();
7
8         for(int i = 1; i < m; i++)
9             for(int j = 1; j < n; j++)
10                 grid[i][j] += min(grid[i-1][j], grid[i][j-1]);
11
12     return grid[m-1][n-1];
13 }
```

Problem List < > ✎

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 196 / 196 testcases passed

Siddhant0705 submitted at Feb 22, 2026 22:43

Runtime 0 ms | Beats 100.00% Memory 15.29 MB | Beats 34.38%

Analyze Complexity

Runtime chart: A bar chart showing runtime distribution. The y-axis ranges from 0% to 150%. The x-axis shows time intervals: 1ms, 2ms, 3ms, 4ms. A single large blue bar reaches nearly 100% at the 1ms mark.

Code C++ Auto

```
16 if(nums[left] <= nums[mid]) {  
17     if(nums[left] <= target && target < nums[mid]) {  
18         right = mid - 1;  
19     } else {  
20         left = mid + 1;  
21     }  
22 }  
23 else { // right half is sorted  
24     if(nums[mid] < target && target <= nums[right]) {  
25         left = mid + 1;  
26     } else {  
27         right = mid - 1;  
28     }  
29 }  
30 }  
31 }  
32 }  
33 }  
34 return -1;  
35 };  
36 }
```

Code | C++

```
1 class Solution {  
2 public:  
3     int search(vector<int>& nums, int target) {  
4         int left = 0;  
5         int right = nums.size() - 1;  
6         while(left <= right) {  
7             ...  
8         }  
9     }  
10 }
```

Saved Ln 36, Col 3

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3