codestudio POWERED BY CODING NINJAS

Practice ▾ Guided Paths Interview Prep ▾ Challenges ▾ New Knowledge Centre ▾ New Community ▾ New

Problem Submissions

C++ (g++ 5.4) ▾

Correct Answer

Report

Submitted on May 13, 2022, 11:14:53 PM

Penalty	Score	Test Cases
0%	80	12/12

Runtime	Language
198ms	C++ (g++ 5.4)

Previous Submissions

Submitted On	Status	Score	Penalty
28 mins ago	Correct Answer 12/12 Test cases Passed	80 / 80	0%
28 mins ago	Correct Answer 12/12 Test cases Passed	80 / 80	0%
30 mins ago	Correct Answer 12/12 Test cases Passed	80 / 80	0%
30 mins ago	Correct Answer 12/12 Test cases Passed	80 / 80	0%
30 mins ago	Correct Answer 12/12 Test cases Passed	80 / 80	0%
48 mins ago	Time Limit Exceeded	60 / 80	0%
14 mins ago	Correct Answer 12/12 Test cases Passed	80 / 80	0%

```cpp
#include <bits/stdc++.h>
int solve(int rows, int columns, vector<vector<int>>& dp){
    if(rows == 1 or columns == 1)
        return dp[rows][columns] = 1;

    if(dp[rows][columns] == -1)
        dp[rows][columns] = solve(rows - 1, columns, dp) + solve(rows, columns - 1, dp);

    return dp[rows][columns];
}
int uniquePaths(int m, int n) {
    // Write your code here.
    vector<int> dp(n, 0); //for storing previous row in "dp" matrix i.e dp[row - 1][0 to m]
    //int ans = solve(m, n, dp);
    //return dp[m][n];

    //Tabulation DP


    for(int column = 0; column < n; column++)
        dp[column] = 1;

    for(int row = 1; row < m; row++){
        int prev = 1;//for each new line Initially prev = 1;
        for(int column = 1; column < n; column++){
            dp[column] = dp[column] + prev;//we have dp[row][column] storedin "prev" variable
            prev = dp[column];
        }

    }
    return dp[n - 1];












}
```

Console ^

Previous Next

Show Hint ✓ Last saved on 11:20:30 PM

▶ Run Code Submit