Problem Submissions Soluti

C++ (g++ 5.4)

Correct Answer

Report

Submitted on May 15, 2022, 2:46:49 PM

Penalty	Score	Test Cases
0%	80	11/11

Runtime	Language
386ms	C++ (g++ 5.4)

Previous Submissions

Submitted On	Status	Score	Penalty	Run
9 mins ago	Correct Answer 11/11 Test cases Passed	80 / 80	0%	312
4 mins ago	Correct Answer 11/11 Test cases Passed	80 / 80	0%	306
14 mins ago	Correct Answer 11/11 Test cases Passed	80 / 80	0%	328

```cpp
int solve(int r, int c, vector<vector<int>>& triangle, int n, vector<vector<int>>& dp){

    if(r == (n - 1))
        return dp[r][c] = triangle[r][c];

    if(dp[r][c] != -1)
        return dp[r][c];

    int d = solve(r + 1, c, triangle, n, dp) + triangle[r][c];
    int dw = solve(r + 1, c + 1, triangle, n, dp) + triangle[r][c];

    return dp[r][c] = min(d, dw);
}
int minimumPathSum(vector<vector<int>>& triangle, int n){
    // Write your code here.
    /*
        //This is Recursive and Memoization Approach

        //For Recursion:
            //  Time Complexity = O(2^n)
            //  Space Complexity = O(n) for stack space

        //For Memoization:
            //  Time Complexity = O(n * n)
            //  Space Complexity = O(n * n)

    vector<vector<int>> dp(n, vector<int>(n, -1));
    int ans = solve(0, 0, triangle, n, dp);
    return ans; //or return dp[n - 1][n - 1];
    */

    /*
        //This is Tabulation Approach
        //  Time Complexity = O(n * n)
        //  Space Complexity = O(n * n)

    vector<vector<int>> dp(n, vector<int>(n, -1));
    for(int i = 0; i < n; i++)
        dp[n - 1][i] = triangle[n - 1][i];

    for(int i = n - 2; i >= 0; i--){
        for(int j = 0; j <= i; j++){
            dp[i][j] = triangle[i][j] + min(dp[i + 1][j], dp[i + 1][j + 1]);
        }
    }

    return dp[0][0];
    */


        //This is space optimized solution
        //  Time Complexity = O(n * n)
        //  Space Complexity = O(n)
    vector<int> dp(n, -1);
    for(int i = 0; i < n; i++)
        dp[i] = triangle[n - 1][i];

    for(int i = n - 2; i >= 0; i--){
        for(int j = 0; j <= i; j++){
            int temp = triangle[i][j] + min(dp[j], dp[j + 1]);
            dp[j] = temp;
        }
    }
    return dp[0];
}
```

Console

Previous Next

Show Hint Last saved on 2:53:12 PM

Run Code Submit