

118. Pascal's Triangle

Solved

Easy Topics Companies

Given an integer `numRows`, return the first numRows of **Pascal's triangle**.

In **Pascal's triangle**, each number is the sum of the two numbers directly above it as shown:



Example 1:

Input: `numRows = 5`
Output: `[[1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1]]`

Example 2:

13.7K 198 134 Online

Code

C++ Auto

```
1 class Solution {
2 public:
3     vector<vector<int>> generate(int numRows) {
4         vector<vector<int>> ans;
5
6         for (int i = 0; i < numRows; ++i)
7             ans.push_back(vector<int>(i + 1, 1));
8
9         for (int i = 2; i < numRows; ++i)
10            for (int j = 1; j < ans[i].size() - 1; ++j)
11                ans[i][j] = ans[i - 1][j - 1] + ans[i - 1][j];
12
13        return ans;
14    }
15 };
```

Saved

In 1, Col 1

Continue to work on your code from 12/24/2024, 11:56:46 AM [Restore](#)

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

191. Number of 1 Bits

Solved

Easy Topics Companies

Given a positive integer `n`, write a function that returns the number of **set bits** in its binary representation (also known as the **Hamming weight**).

Example 1:

Input: `n = 11`
Output: 3
Explanation:
The input binary string **1011** has a total of three set bits.

Example 2:

Input: `n = 128`
Output: 1
Explanation:
The input binary string **10000000** has a total of one set bit.

Code

C++ Auto

```
1 class Solution {
2 public:
3     int hammingWeight(uint32_t n) {
4         int ans = 0;
5
6         for (int i = 0; i < 32; ++i)
7             if ((n >> i) & 1)
8                 ++ans;
9
10        return ans;
11    }
12};
```

Saved Ln 12, Col 3

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input
n =

Description Accepted Editorial Solutions Submissions

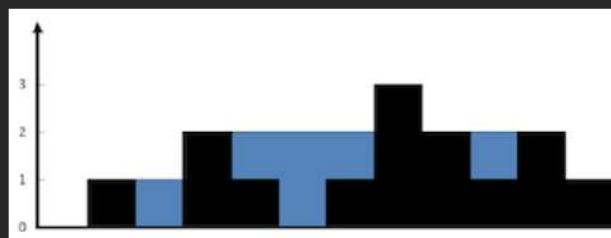
42. Trapping Rain Water

Solved

Hard Topics Companies

Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining.

Example 1:



Input: height = [0,1,0,2,1,0,1,3,2,1,2,1]

Output: 6

Explanation: The above elevation map (black section) is represented by array [0,1,0,2,1,0,1,3,2,1,2,1]. In this case, 6 units of rain water (blue section) are being trapped.

Example 2:

33.8K 377 ☆

422 Online

Code

C++ Auto

```
1 class Solution {
2 public:
3     int trap(vector<int>& height) {
4         const int n = height.size();
5         int ans = 0;
6         vector<int> l(n); // l[i] := max(height[0..i])
7         vector<int> r(n); // r[i] := max(height[i..n))
8
9         for (int i = 0; i < n; ++i)
10             l[i] = i == 0 ? height[i] : max(height[i], l[i - 1]);
11
12         for (int i = n - 1; i >= 0; --i)
13             r[i] = i == n - 1 ? height[i] : max(height[i], r[i + 1]);
14
15         for (int i = 0; i < n; ++i)
16             ans += min(l[i], r[i]) - height[i];
17
18         return ans;
19     }
```

Saved

Ln 20, Col 3

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

height =

297. Serialize and Deserialize Binary Tree

Solved

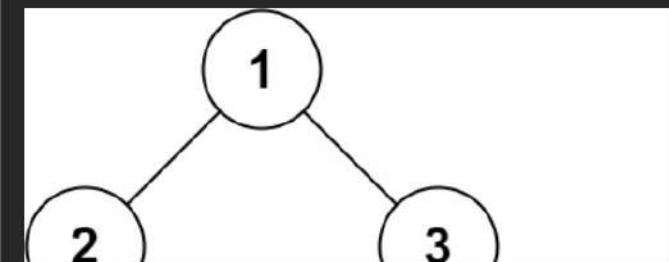
Hard Topics Companies

Serialization is the process of converting a data structure or object into a sequence of bits so that it can be stored in a file or memory buffer, or transmitted across a network connection link to be reconstructed later in the same or another computer environment.

Design an algorithm to serialize and deserialize a binary tree. There is no restriction on how your serialization/deserialization algorithm should work. You just need to ensure that a binary tree can be serialized to a string and this string can be deserialized to the original tree structure.

Clarification: The input/output format is the same as [how LeetCode serializes a binary tree](#). You do not necessarily need to follow this format, so please be creative and come up with different approaches yourself.

Example 1:



10.6K 89

95 Online

Code

C++ Auto

```
1 class Codec {
2 public:
3     // Encodes a tree to a single string.
4     string serialize(TreeNode* root) {
5         if (root == nullptr)
6             return "";
7
8         string s;
9         queue<TreeNode*> q{{root}};
10
11         while (!q.empty()) {
12             TreeNode* node = q.front();
13             q.pop();
14             if (node != nullptr) {
15                 s += to_string(node->val) + " ";
16                 q.push(node->left);
17                 q.push(node->right);
18             } else {
```

Saved

Ln 53, Col 3

Testcase Test Result

Accepted Runtime: 3 ms

Case 1 Case 2

Input

root =