



Introduction to Data Structure

Binary Tree



Overview

A tree is a frequently-used data structure to simulate a hierarchical tree structure. Each node of the tree will have a root value and a list of references to other nodes which are called child nodes. From graph view, a tree can also be defined as a directed acyclic graph which has N nodes and $N-1$ edges. A



Traverse A Tree

In the introduction, we have gone through the concept of a tree and a binary tree. In this chapter, we will focus on the traversal methods used in a binary tree. Understanding these traversal methods will definitely help you have a better understanding of the tree structure and have a solid foundation for the



Solve Problems Recursively

In previous sections, we have introduced how to solve tree traversal problem. And we tried to solve preorder, inorder and postorder traversal recursively. Actually, recursion is one of the most powerful and frequently-used methods for solving tree related problems. We are going to introduce two typical



Conclusion

After finishing the previous chapters, you should be familiar with binary trees and be able to solve basic problems related to them. In this chapter, we are going to provide you with more exercises to help you feel more confident with this topic.

Introduction



A `tree` is a frequently-used data structure to simulate a hierarchical tree structure.

Each node of the tree will have a root value and a list of references to other nodes which are called child nodes. From graph view, a tree can also be defined as a directed acyclic graph which has N nodes and $N-1$ edges.

A `Binary Tree` is one of the most typical tree structure. As the name suggests, a binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.

By completing this card, you will be able to:


1. Understand the concept of a `tree` and a `binary tree`;
2. Be familiar with different `traversal` methods;
3. Use `recursion` to solve binary-tree-related problems;





Traverse A Tree



<input checked="" type="checkbox"/>  A Traverse a Tree - Introduction	
<input type="checkbox"/>  Binary Tree Preorder Traversal	
<input type="checkbox"/>  Binary Tree Inorder Traversal	
<input type="checkbox"/>  Binary Tree Postorder Traversal	
<input type="checkbox"/>  A Binary Tree Traversal - Solution	
<input checked="" type="checkbox"/>  A Level-order Traversal - Introduct...	
<input type="checkbox"/>  Binary Tree Level Order Traversal	
<input checked="" type="checkbox"/>  A Binary Tree Level Traversal - Solu...	



Solve Problems Recursively	
<input checked="" type="checkbox"/>  A Solve Tree Problems Recursively	
<input type="checkbox"/>  Maximum Depth of Binary Tree	
<input type="checkbox"/>  Symmetric Tree	
<input type="checkbox"/>  Path Sum	
<input type="checkbox"/>  Count Univalued Subtrees	

Conclusion	
<input type="checkbox"/>  Construct Binary Tree from Inord...	
<input type="checkbox"/>  Construct Binary Tree from Preor...	
<input type="checkbox"/>  Populating Next Right Pointers i...	
<input type="checkbox"/>  Populating Next Right Pointers i...	
<input type="checkbox"/>  Lowest Common Ancestor of a B...	
<input type="checkbox"/>  Serialize and Deserialize Binary T...	