Theory: Tree

13 minutes 9 / 9 problems solved

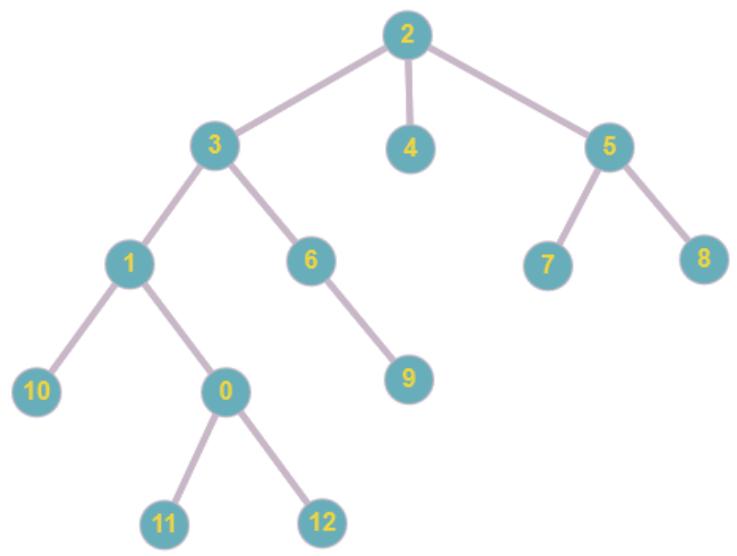
Start practicing

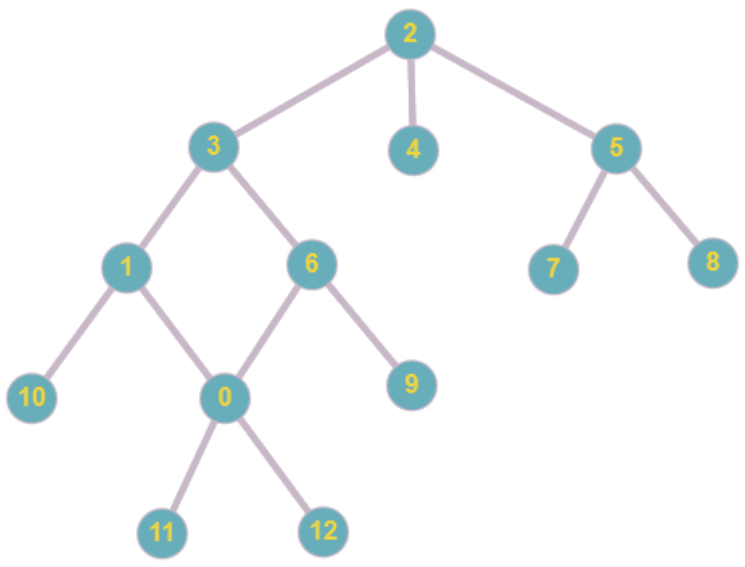
§1. Tree

The first thing that comes to mind is probably a branchy oak in a forest. You aren't too far off: that very image gave name to a specific type of data structure that we are going to look at. In a way, it resembles real trees (both have roots and leaves and a similar structure) and family trees with parents, children, and siblings. Keeping that analogy in mind, let's look closely at what trees are in computer science.

Any **tree** is a graph, but one with certain conditions: it is a connected graph without cycles. The main property of a tree is that there is only one path between any two nodes (also called vertices) of the graph. If there is at least one cycle in the graph, it means that there are pairs of nodes with more than one path connecting them: in other words, it isn't a tree.

Let's visualize that: the following figures show two graphs. The first is a tree and the second one isn't.





The second graph is not a tree because it contains a $1 - 3 - 6 - 0$ cycle, so there are two paths from node 2 to node 0: $2 - 3 - 1 - 0$ and $2 - 3 - 6 - 0$.

§2. Important terms and definitions

A tree starts with its topmost node which is called the **root node** or the root of a tree. For each tree node all nodes directly below it (connected by an edge) are called **child nodes** or node's children. In the example above (first figure) node 2 is the root of the tree, while nodes 3, 4, and 5 are root's children. So far quite simple, very much like a family tree.

Let's go over the basic definitions related to trees:

- the **root node** is the topmost node of a tree;
- the **depth** of a node is the number of edges from the root to this node;
- a **child** is a node directly below a given node, connected by an edge;

Current topic:

- ✓ Tree

Topic depends on:

- ✓ Recursion basics
- ✓ Graph

Topic is required for:

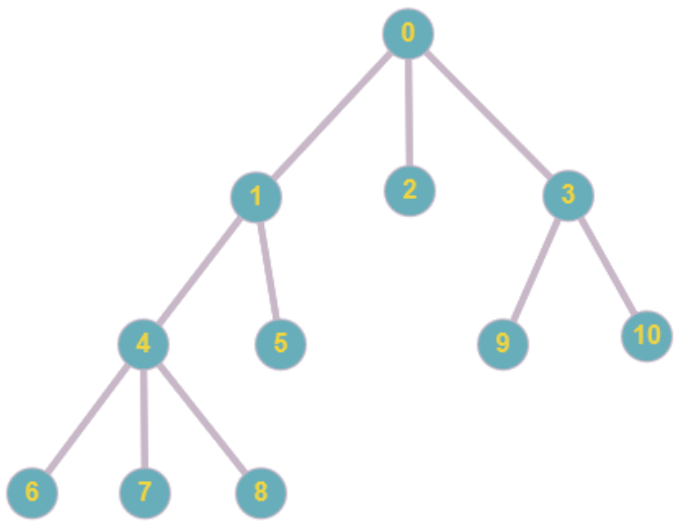
- ✓ Binary heap
- ✓ Binary search tree
- ✓ Tree traversals
- ✓ Spanning trees
- Trees in Java
- ✓ Suffix tree

Table of contents:

- a **parent** is a node directly above a given node, connected by an edge;
- a **leaf** is a node without children;
- the **height of a node** is the number of edges on the longest path from the node to a leaf;
- the **height of a tree** is the height of its root node.

In this and the following graph-related topics, we will use both Node and Vertex as names for points in the graph: treat them as synonyms. In the literature, they are also used interchangeably.

Nodes or edges in the tree may contain information: for example, the number of the vertex or the cost of moving along the edge (the sum of edges from the root to this vertex). The image below shows an example of a tree with numbered nodes:

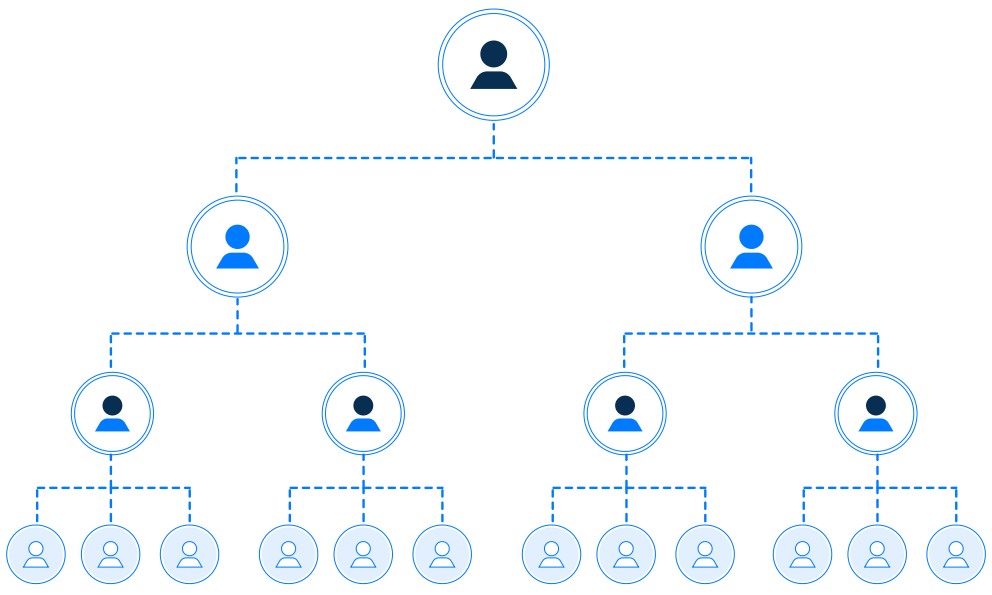


In this example, node 0 is the root of the tree, while nodes 6, 7, 8, 5, 2, 9, and 10 are leaves.

Note that if you add an extra edge to the tree, a cycle will be formed, and if you remove any of the edges, the graph will become disconnected.

A tree in which every node has no more than two children is called a **binary tree**. If the number of nodes in the tree is n, then the depth of the tree is at least $log_2(n)$. From that point on, we will discuss specifically binary trees, unless stated otherwise.

§3. Trees in practice



The tree data structure is very common in practice: for example, we have already mentioned family trees. Taxonomy in biology makes use of the same hierarchical structure; if you want to bring it closer to home, you can think about the "boss-subordinate" structure in a company, where children stand for subordinates, and the parent is the boss.

Trees are widely used in IT because they allow us to process data considerably faster. Virtually all parsers of different grammars use trees to a varying degree. Also, in many DBMS's, indexes are based on trees for faster data processing.

Report a typo

86 users liked this theory. **1** didn't like it. **What about you?**

Start practicing

Comments (2) Hints (0) Useful links (3) Show discussion