# DSA Trees

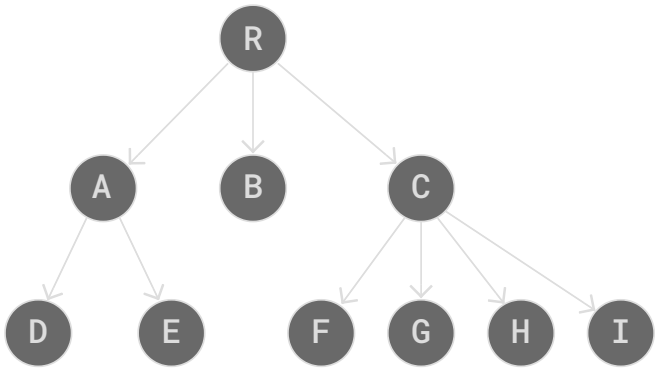
❮ Previous Next ❯

## Trees

The Tree data structure is similar to Linked Lists in that each node contains data and can be linked to other nodes.

We have previously covered data structures like Arrays, Linked Lists, Stacks, and Queues. These are all linear structures, which means that each element follows directly after another in a sequence. Trees however, are different. In a Tree, a single element can have multiple 'next' elements, allowing the data structure to branch out in various directions.

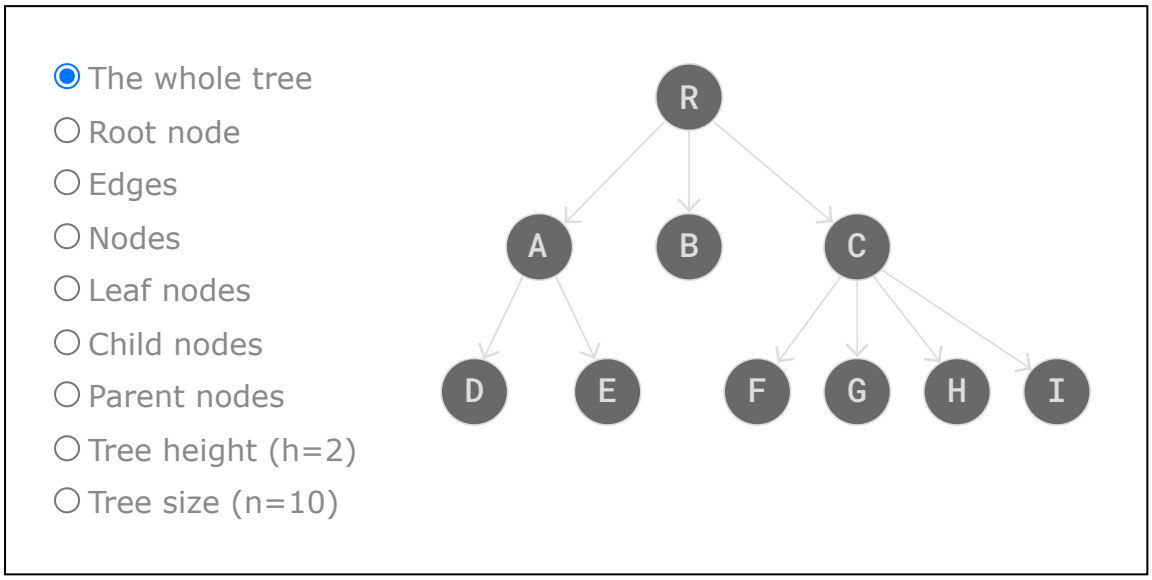
The data structure is called a "tree" because it looks like a tree, only upside down, just like in the image below.




The Tree data structure can be useful in many cases:

- Hierarchical Data: File systems, organizational models, etc.
- Databases: Used for quick data retrieval.
- Routing Tables: Used for routing data in network algorithms.
- Sorting/Searching: Used for sorting data and searching for data.
- Priority Queues: Priority queue data structures are commonly implemented using trees, such as binary heaps.

visualization below.




The first node in a tree is called the **root** node.

A link connecting one node to another is called an **edge**.

A **parent** node has links to its **child** nodes. Another word for a parent node is **internal** node.

A node can have zero, one, or many child nodes.

A node can only have one parent node.

Nodes without links to other child nodes are called **leaves**, or **leaf nodes**.

The **tree height** is the maximum number of edges from the root node to a leaf node. The height of the tree above is 2.

The **height of a node** is the maximum number of edges between the node and a leaf node.

The **tree size** is the number of nodes in the tree.

---

# Types of Trees

Trees are a fundamental data structure in computer science, used to represent hierarchical relationships. This tutorial covers several key types of trees.

**Binary Search Trees (BSTs):** A type of Binary Tree where for each node, the left child node has a lower value, and the right child node has a higher value.

**AVL Trees:** A type of Binary Search Tree that self-balances so that for every node, the difference in height between the left and right subtrees is at most one. This balance is maintained through rotations when nodes are inserted or deleted.

Each of these data structures are described in detail on the next pages, including animations and how to implement them.

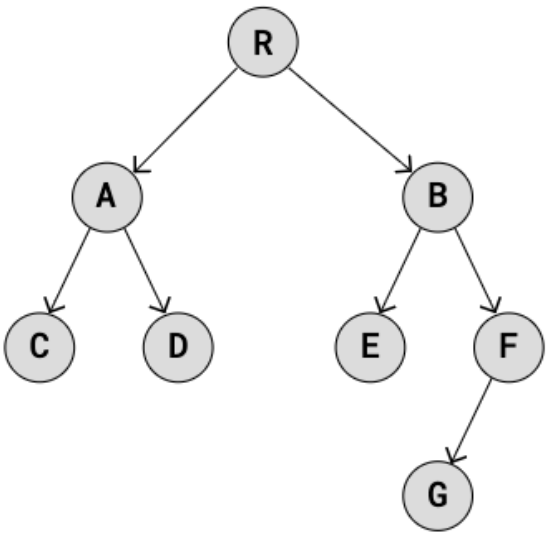
---

# DSA Exercises

## Test Yourself With Exercises

### Exercise:

In a Tree data structure, like the one below:




What are nodes C, D, E, and G called?