



DSA Trees

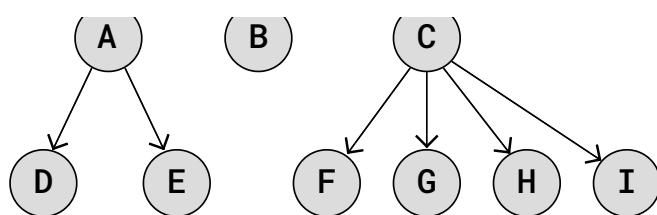
 [Previous](#)[Next](#)

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/Search: Used for sorting data and searching for data.
- Priority Queues: Priority queue data structures are commonly implemented using trees, such as binary heaps.

Tree Terminology and Rules

Learn words used to describe the tree data structure by using the interactive tree visualization below.

The whole tree

Root node

Edges

Nodes

Leaf nodes

Child nodes

Parent nodes

Tree height ($h=2$)

Tree size ($n=10$)

```

graph TD
    R((R)) --> A((A))
    R --> B((B))
    R --> C((C))
    A --> D((D))
    A --> E((E))
    C --> F((F))
    C --> G((G))
    C --> H((H))
    C --> I((I))
  
```

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.



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 Trees: Each node has up to two children, the left child node and the right child node. This structure is the foundation for more complex tree types like Binary Search Trees and AVL 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.

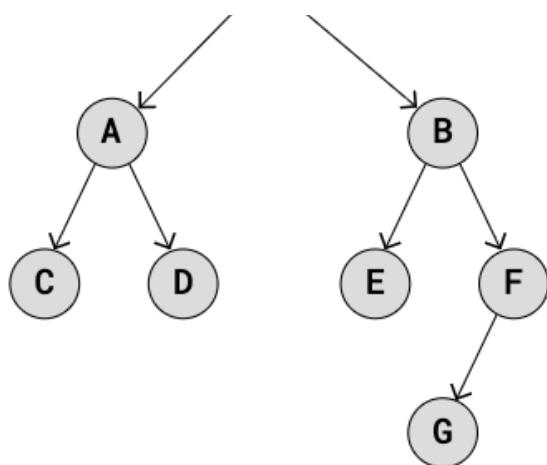
AVL Trees: A type of Binary Search Tree that self-balances so that for every node, the difference in height between the left and right subtrees is at most one. This balance is maintained through rotations when nodes are inserted or deleted.

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:



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

Nodes C, D, E, and G
are called nodes.

[Submit Answer »](#)

[Start the Exercise](#)

[Previous](#)

[Sign in to track progress](#)

[Next >](#)



Tutorials ▾

References ▾

Exercises ▾



Sign In

☰ SS

JAVASCRIPT

SQL

PYTHON

JAVA

PHP

HOW TO

W3.CSS

C

C



COLOR PICKER



PLUS

SPACES

GET CERTIFIED

FOR TEACHERS

FOR BUSINESS

CONTACT US

Top Tutorials

[HTML Tutorial](#)
[CSS Tutorial](#)



Tutorials ▾

References ▾

Exercises ▾



Sign In

☰ SS

JAVASCRIPT

SQL

PYTHON

JAVA

PHP

HOW TO

W3.CSS

C

C

PHP Tutorial
Java Tutorial
C++ Tutorial
jQuery Tutorial

Top References

HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
Java Reference
AngularJS Reference
jQuery Reference

Top Examples

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

Get Certified

HTML Certificate
CSS Certificate
JavaScript Certificate
Front End Certificate
SQL Certificate
Python Certificate
PHP Certificate
jQuery Certificate
Java Certificate
C++ Certificate
C# Certificate
XML Certificate



FORUM ABOUT ACADEMY

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning.
Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our [terms of use](#), [cookies](#) and [privacy policy](#).

Copyright 1999-2026 by Refsnes Data. All Rights Reserved. W3Schools is Powered by W3.CSS.