Trees

Definition:

A tree is defined as a **collection of nodes** connected by edges, such that there is exactly one path between any two nodes in the tree.

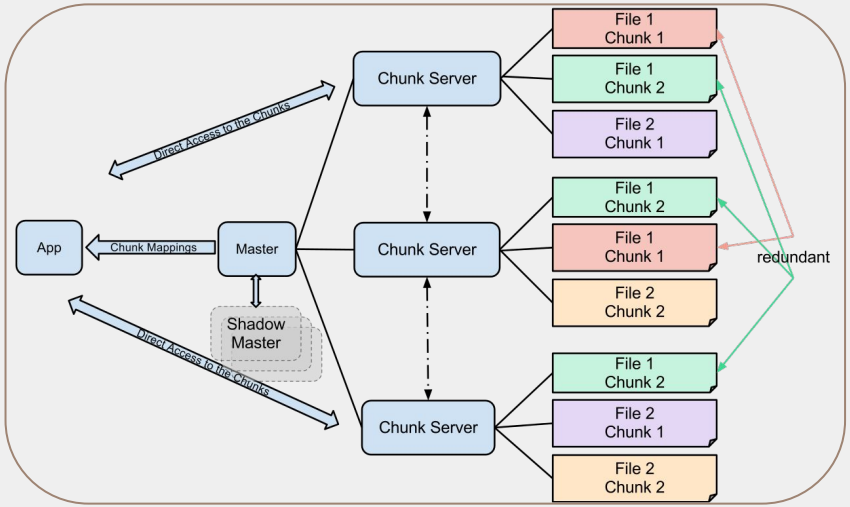
Easy Language: A tree is a **hierarchical data structure** that consists of **nodes** connected by **edges**. Each node in a tree can have zero or more children, but must have exactly one parent, except for the root node, which has no parent.

Properties of trees

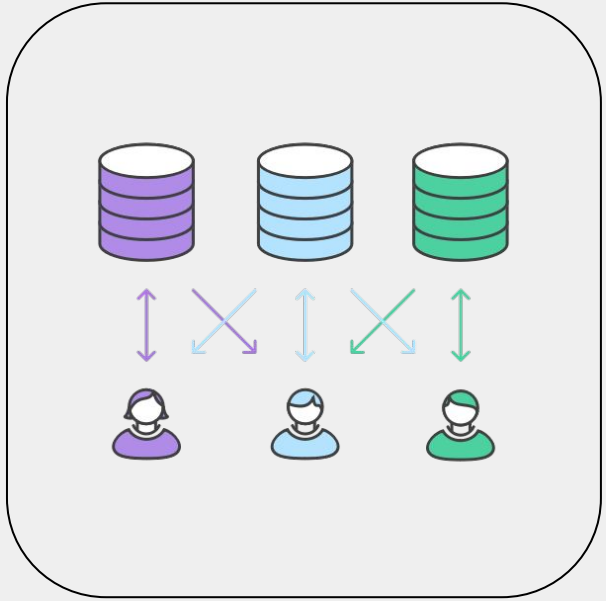
- A tree has at least one node, the root node.
- Each node in a tree has zero or more children.
- Each node in a tree has exactly one parent, except for the root node.
- There is exactly one path between any two nodes in a tree.

Applications of trees in computer science

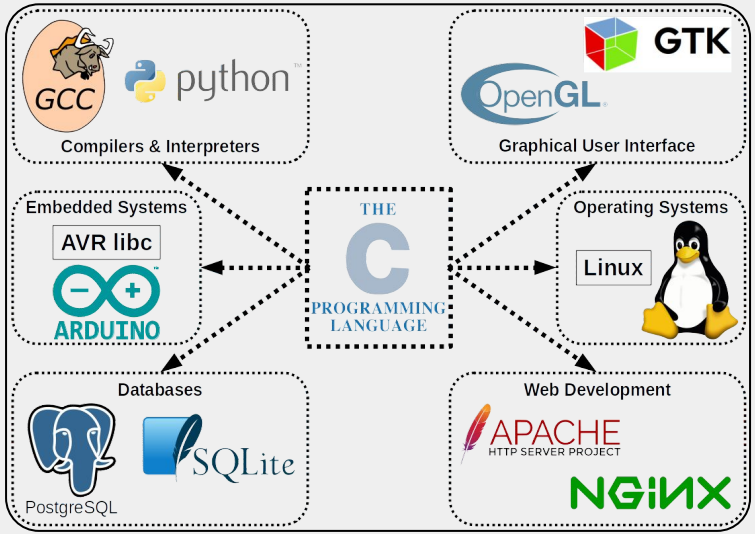
File Systems



Databases



Compilers


GCC
python
Compilers & Interpreters
GTK
OpenGL
Graphical User Interface
Embedded Systems
AVR libc
ARDUINO
THE
C
PROGRAMMING
LANGUAGE
Operating Systems
Linux
Databases
PostgreSQL
SQLite
Web Development
APACHE
HTTP SERVER PROJECT
NGINX

Search Engines

1

Types of Trees

Trees

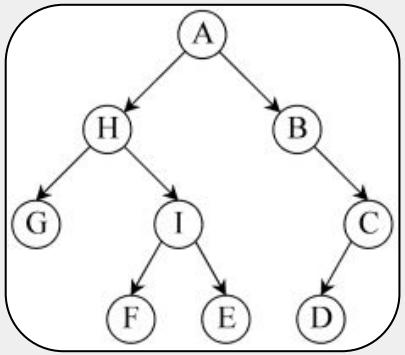
Types of Trees:

1. Binary Trees
2. AVL Trees
3. Red - Black Trees
4. B - Tree
5. Splay Tree
6. Trie

And Many More ...

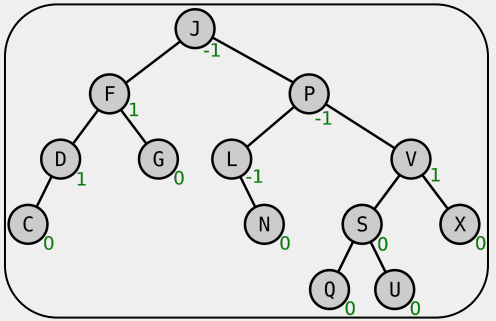
Binary Trees:

A binary tree is a tree in which each node can have at most two children. Binary trees are the most common type of tree used in data structures.



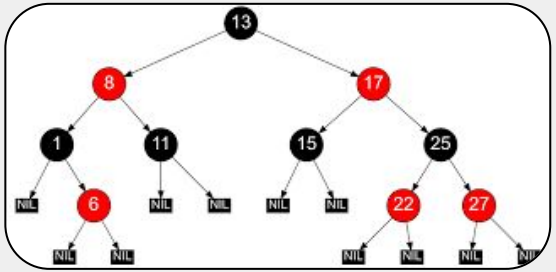
AVL Trees:

An AVL tree is a self-balancing binary search tree. This means that the tree automatically maintains its balance, even after insertion and deletion operations. AVL trees are often used in applications where fast search and insertion times are required.



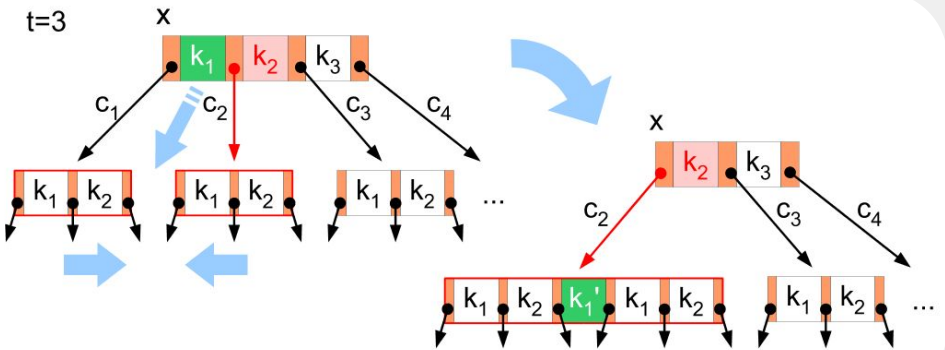
Red – Black Trees:

A red-black tree is another type of self-balancing binary search tree. Red-black trees are similar to AVL trees, but they have slightly different rules for maintaining balance. Red-black trees are often used in database systems and other applications where efficient search and insertion are required.



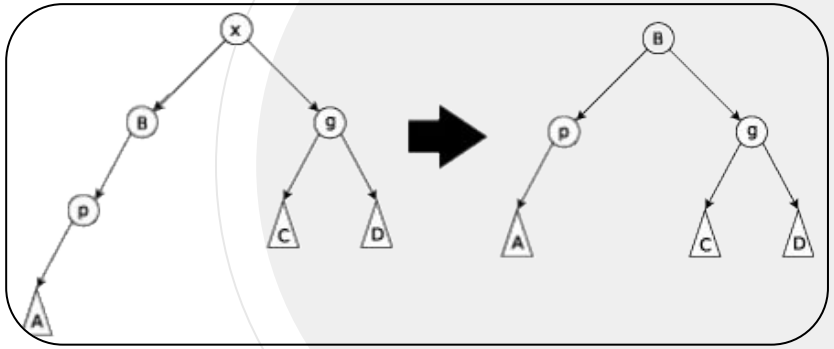
B - Trees:

A B-tree is a tree data structure that is designed for storing and retrieving large amounts of data efficiently. B-trees can have more than two children per node, and they can also store multiple values per node. B-trees are often used in database systems and file systems.



Splay Trees:

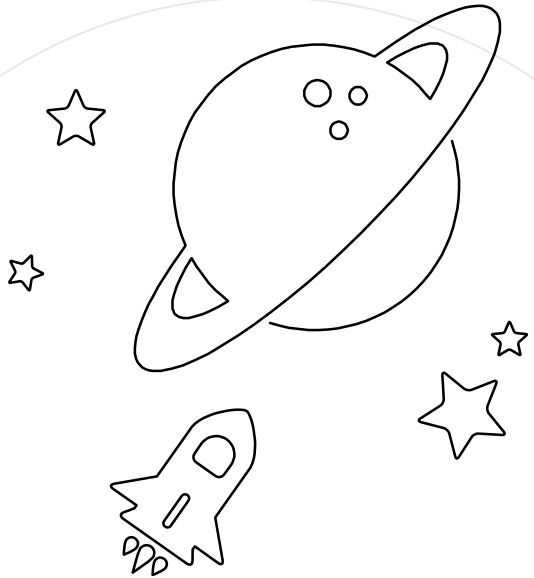
A splay tree is a self-balancing binary search tree that uses a technique called splaying to improve the performance of search operations. Splaying trees are often used in applications where frequently accessed data needs to be retrieved quickly.



2

Traversals

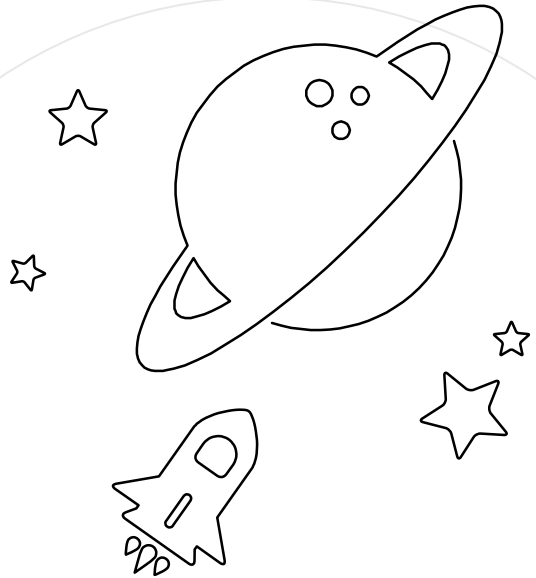
Trees



PreOrder Traversal

Preorder Traversal

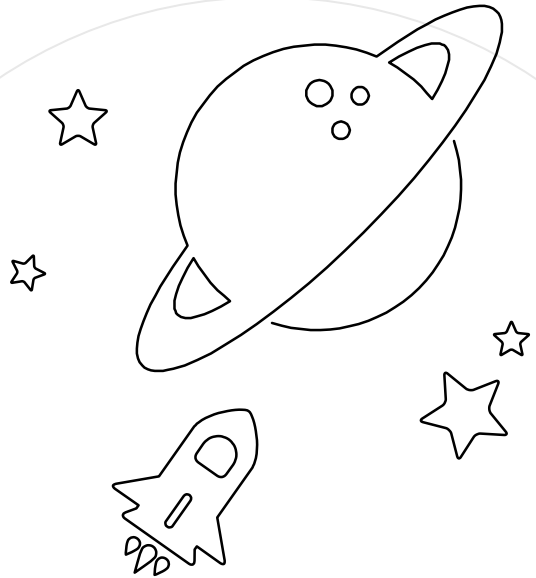
In a preorder traversal, the root node is visited first, followed by its left subtree, and then its right subtree. This process is repeated recursively for each subtree.



InOrder Traversal

Inorder Traversal

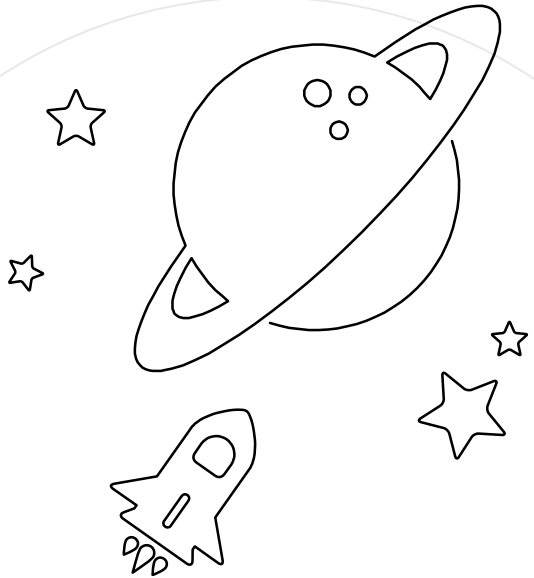
In an inorder traversal, the left subtree of a node is visited first, followed by the node itself, and then its right subtree. This process is repeated recursively for each subtree.



PostOrder Traversal

Postorder Traversal

In a postorder traversal, the left subtree of a node is visited first, followed by its right subtree, and then the node itself. This process is repeated recursively for each subtree.



Level -Order Traversal

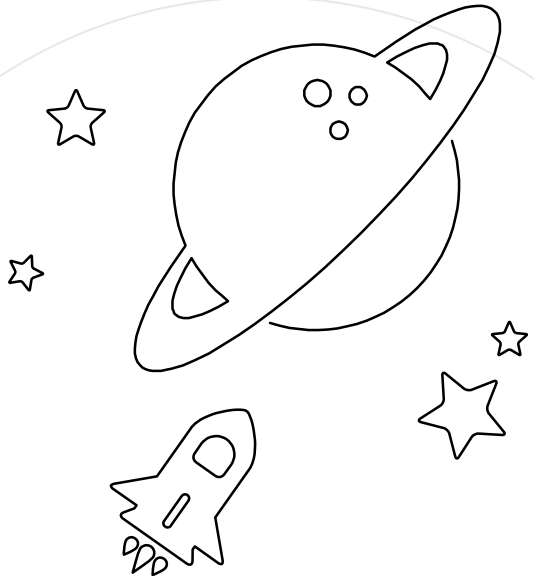
Level Order Traversal

In a level-order traversal, all nodes at the same level of the tree are visited before moving on to the next level. The traversal starts at the root node and proceeds level by level, from top to bottom.

3

Operations

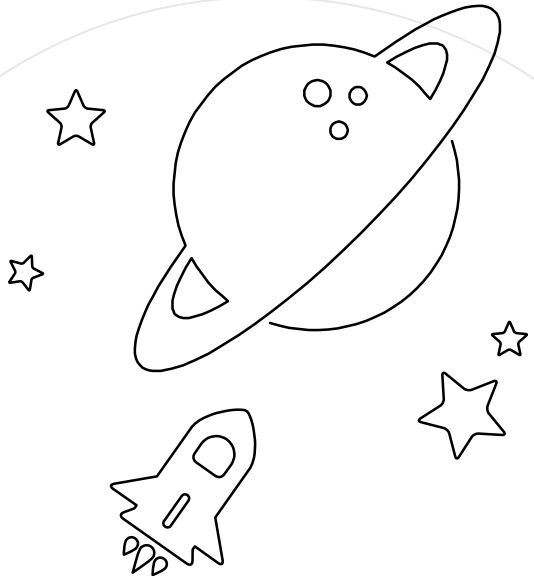
Trees



Insertion

Insertion

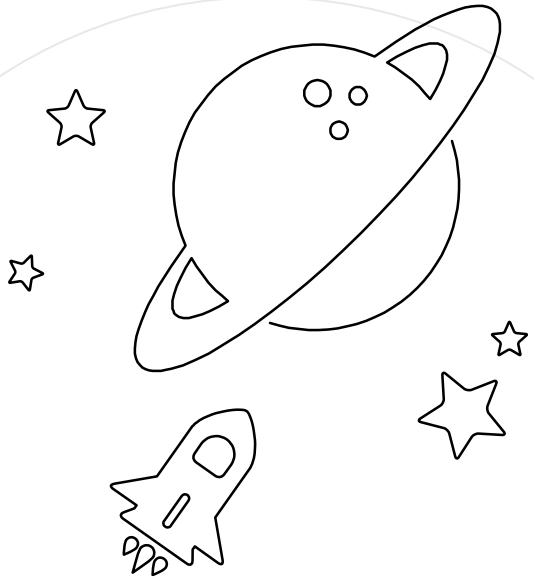
Inserting a node into a tree involves finding the appropriate location for the node and then adding it to the tree. The location of the node is determined by the type of tree and the values of the nodes that are already in the tree.



Deletion

Deletion

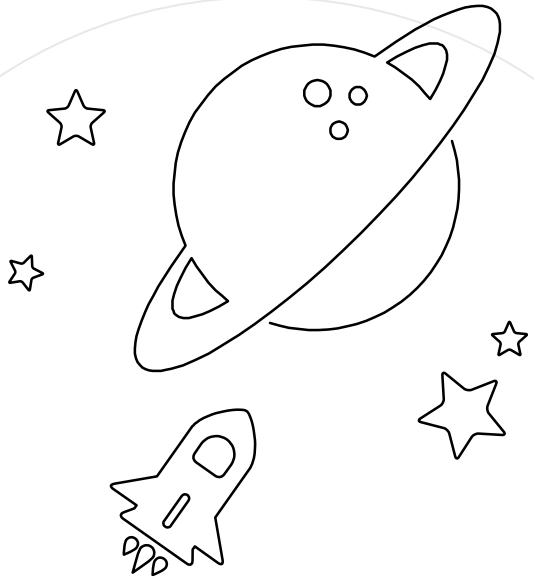
Deleting a node from a tree involves finding the node and then removing it from the tree. The process of deleting a node can be complex, as it is necessary to maintain the balance of the tree.



Min And Max Values

Finding the Min and Max values

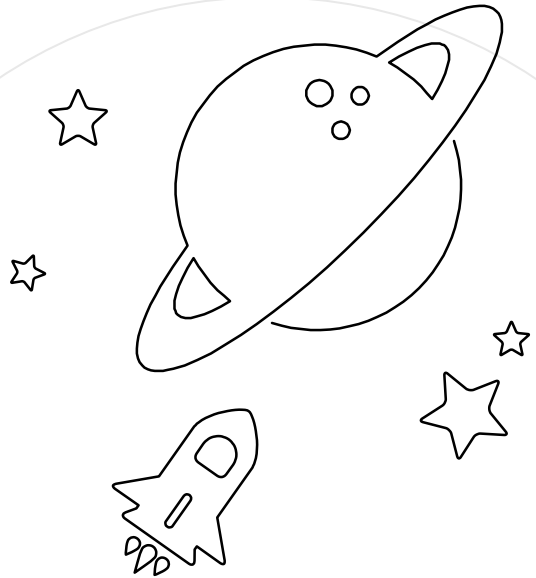
Finding the minimum and maximum values in a tree can be done by recursively traversing the tree and keeping track of the smallest and largest values that have been visited.



Height and Depth

Finding the height and depth of node

The height of a node in a tree is the number of edges on the longest path from the node to the root node. The depth of a node is the number of edges on the path from the node to the root node.



Balanced

Check if Tree is Balanced

The height of a node in a tree is the number of edges on the longest path from the node to the root node. The depth of a node is the number of edges on the path from the node to the root node.

Thanks!

Any questions?