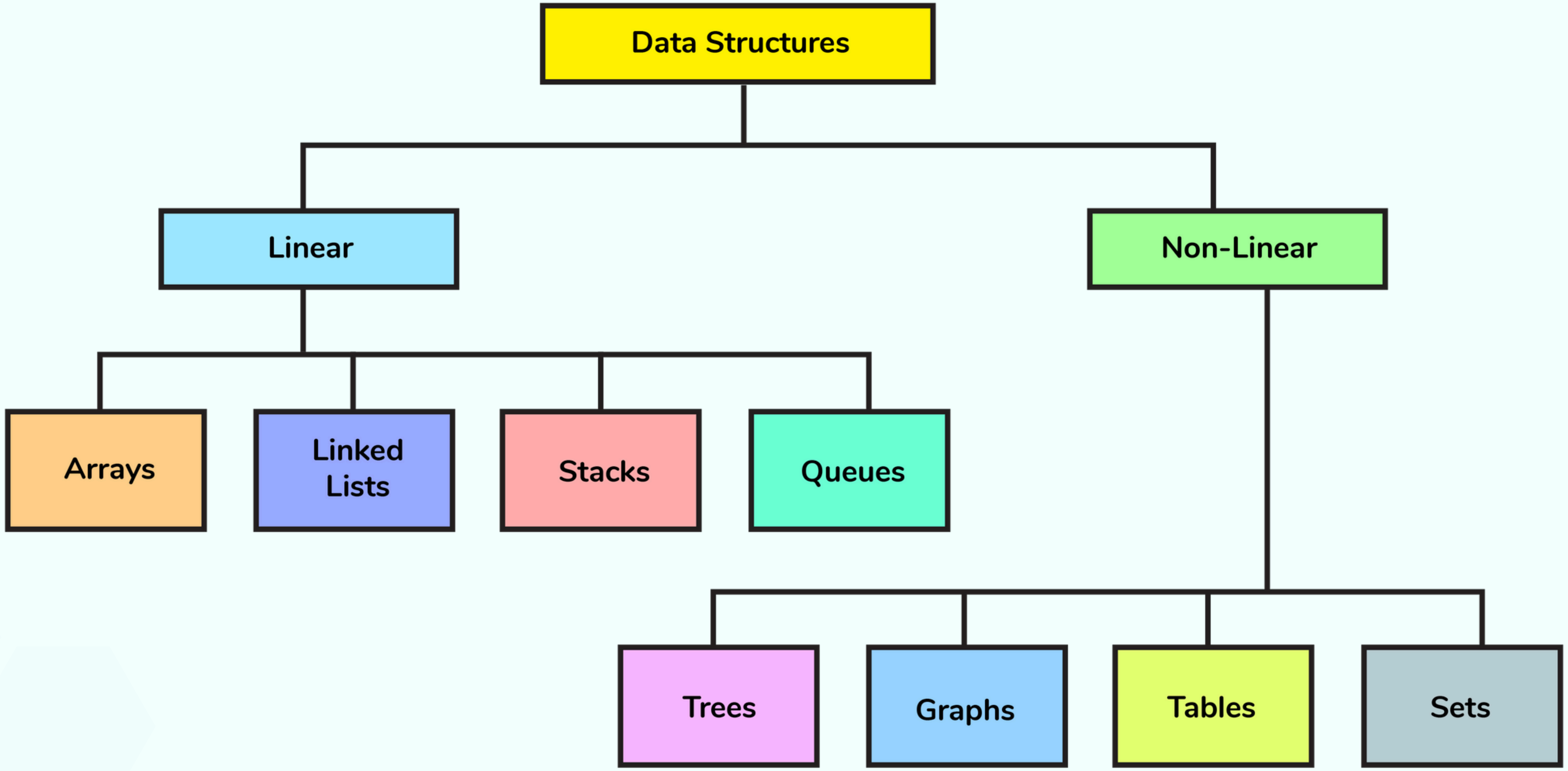
Intro to data structure & algorithms

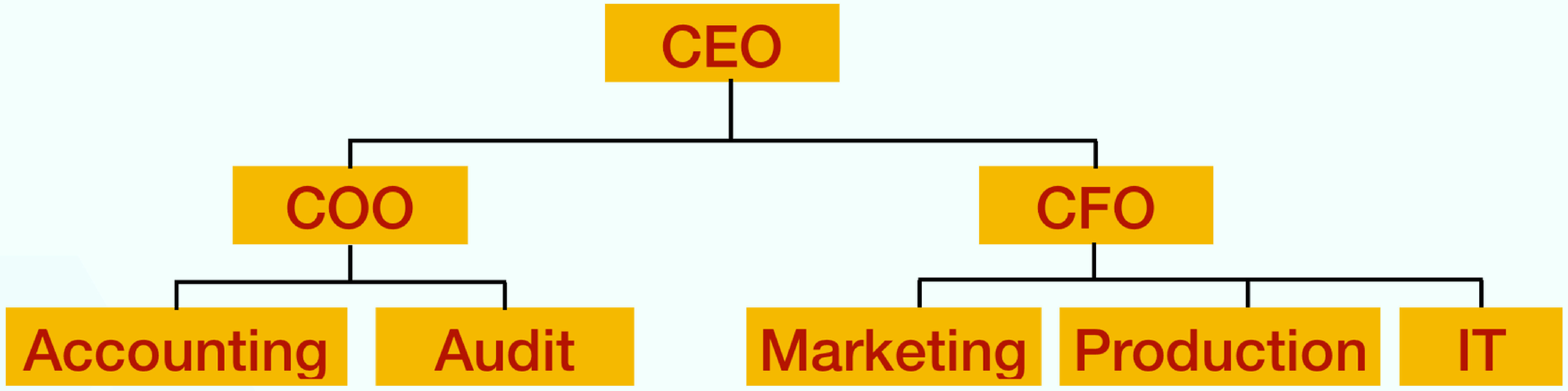
by Mariana Makram / ITI-sohag

Classification of Data Structures



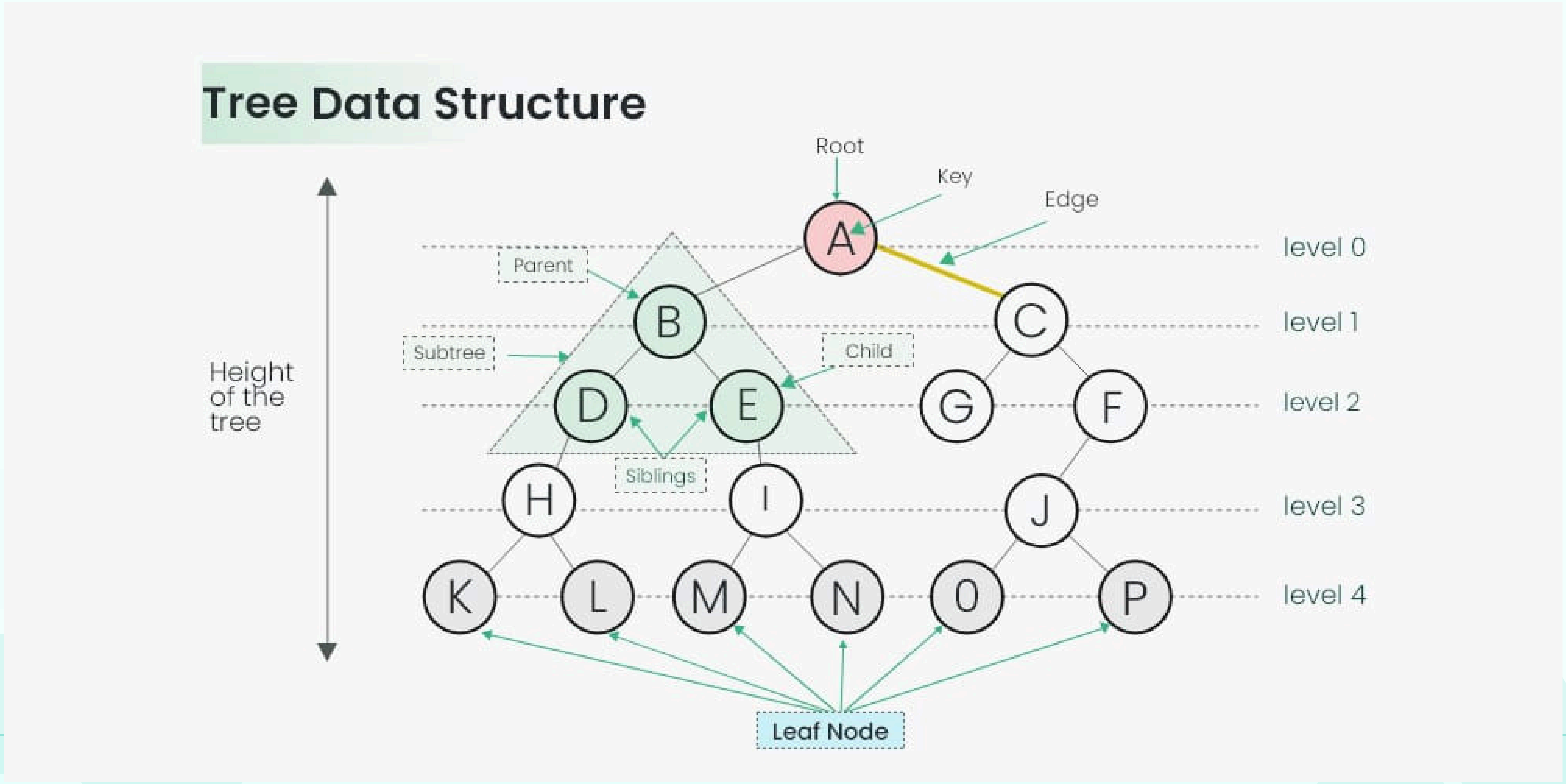
Tree Data Structures

A tree is a hierarchical data structure in which the elements (known as nodes) are linked together via edges such that there is only one path between any two node of the tree.



A Corporate Tree

Tree data structure

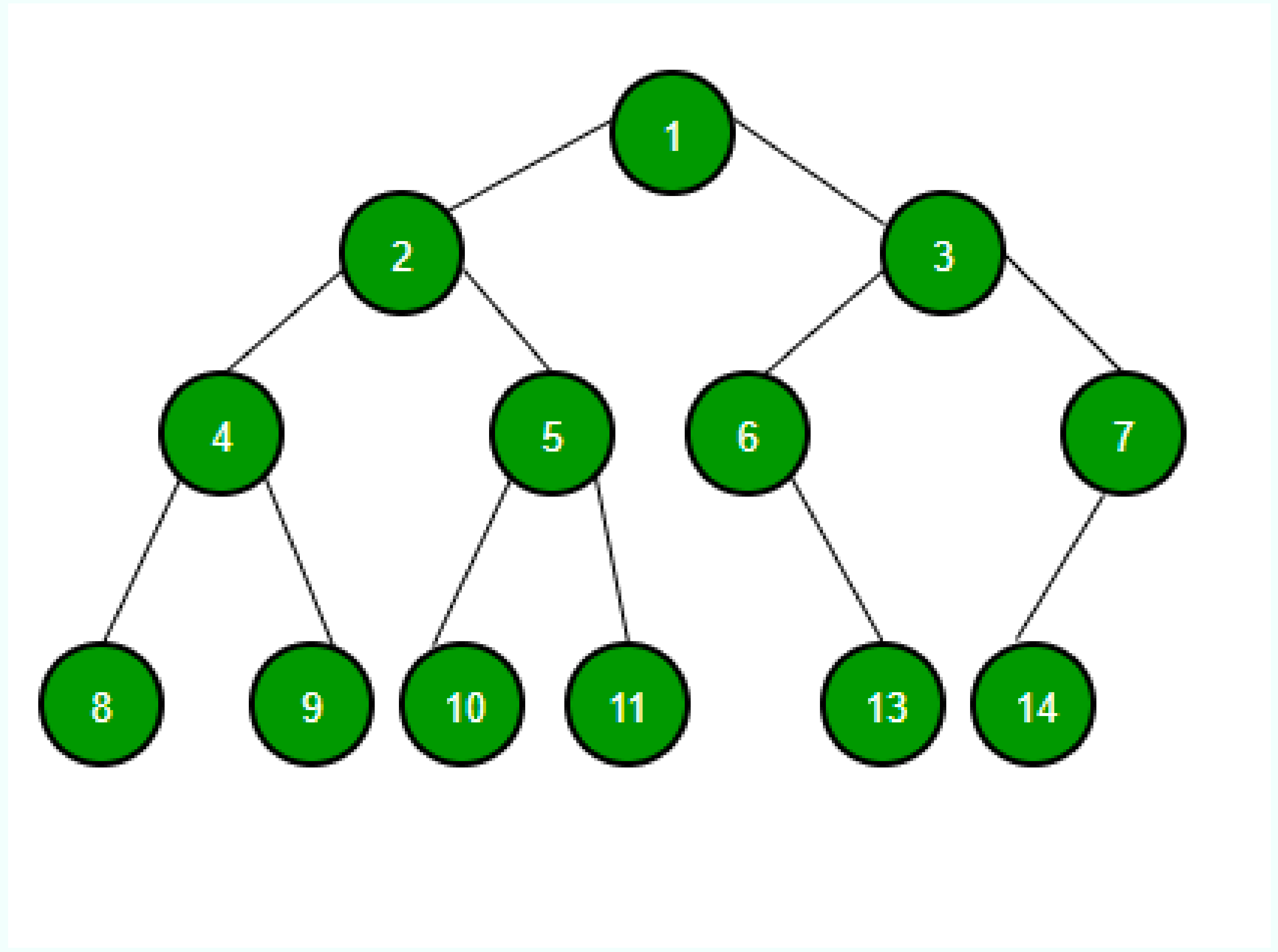


Types of Tree Data Structure

- **Binary tree**
  - Each node can have a maximum of two children linked to it.
- **Ternary Tree**
  - A Ternary Tree is a tree data structure in which each node has at most three child nodes, usually distinguished as “left”, “mid” and “right”.
- **N-ary Tree or Generic Tree**
  - Many children at every node.
  - The number of nodes for each node is not known in

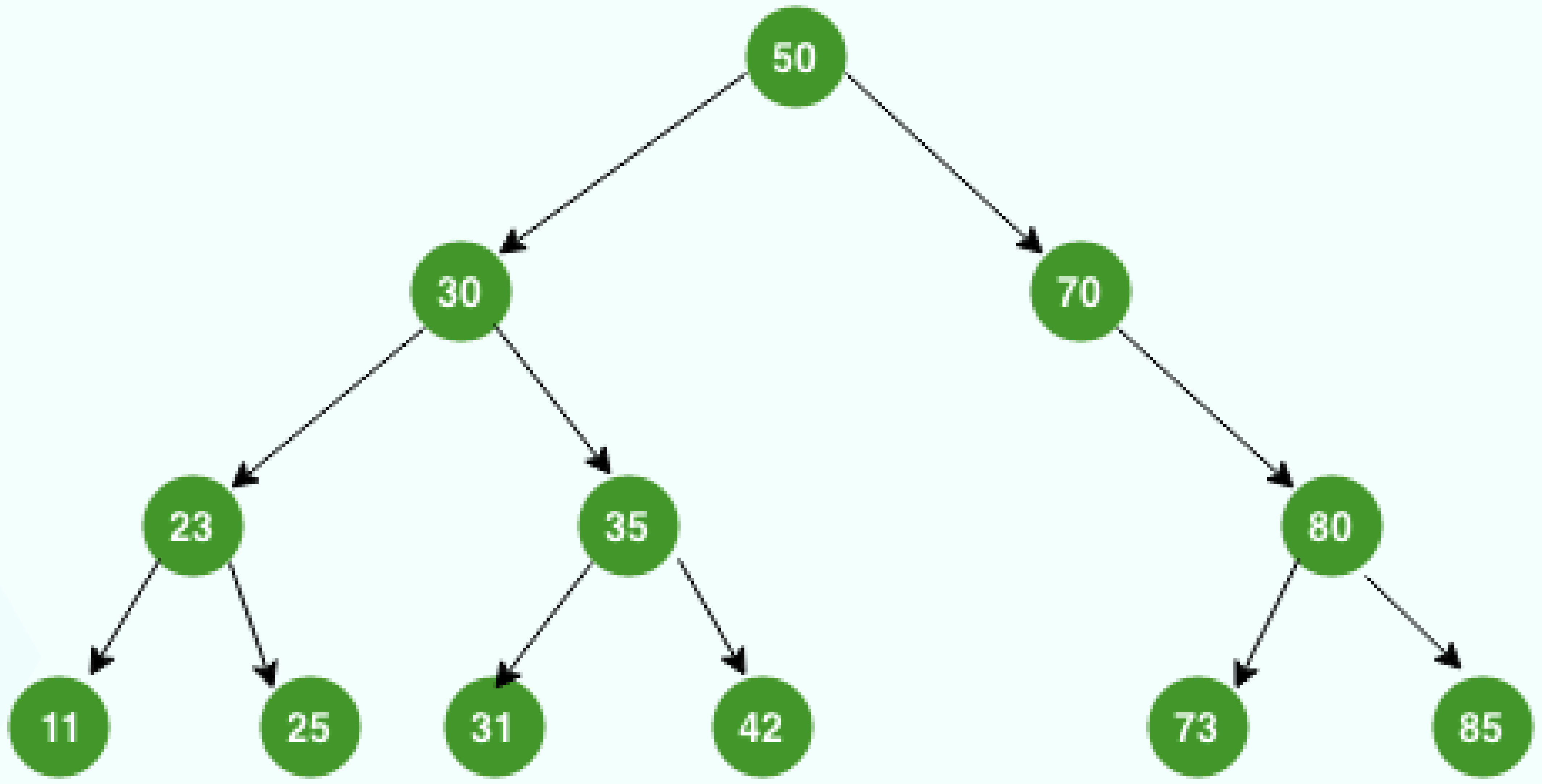
Binary tree

A Binary Tree Data Structure is a hierarchical data structure in which each node has at most two children, referred to as the left child and the right child.



Binary search tree

Is a Binary Tree with the left child containing values less than the parent node and the right child containing values greater than the parent node.



Binary Search Tree (BST)

Binary Search Tree is a node-based binary tree data structure which has the following properties:

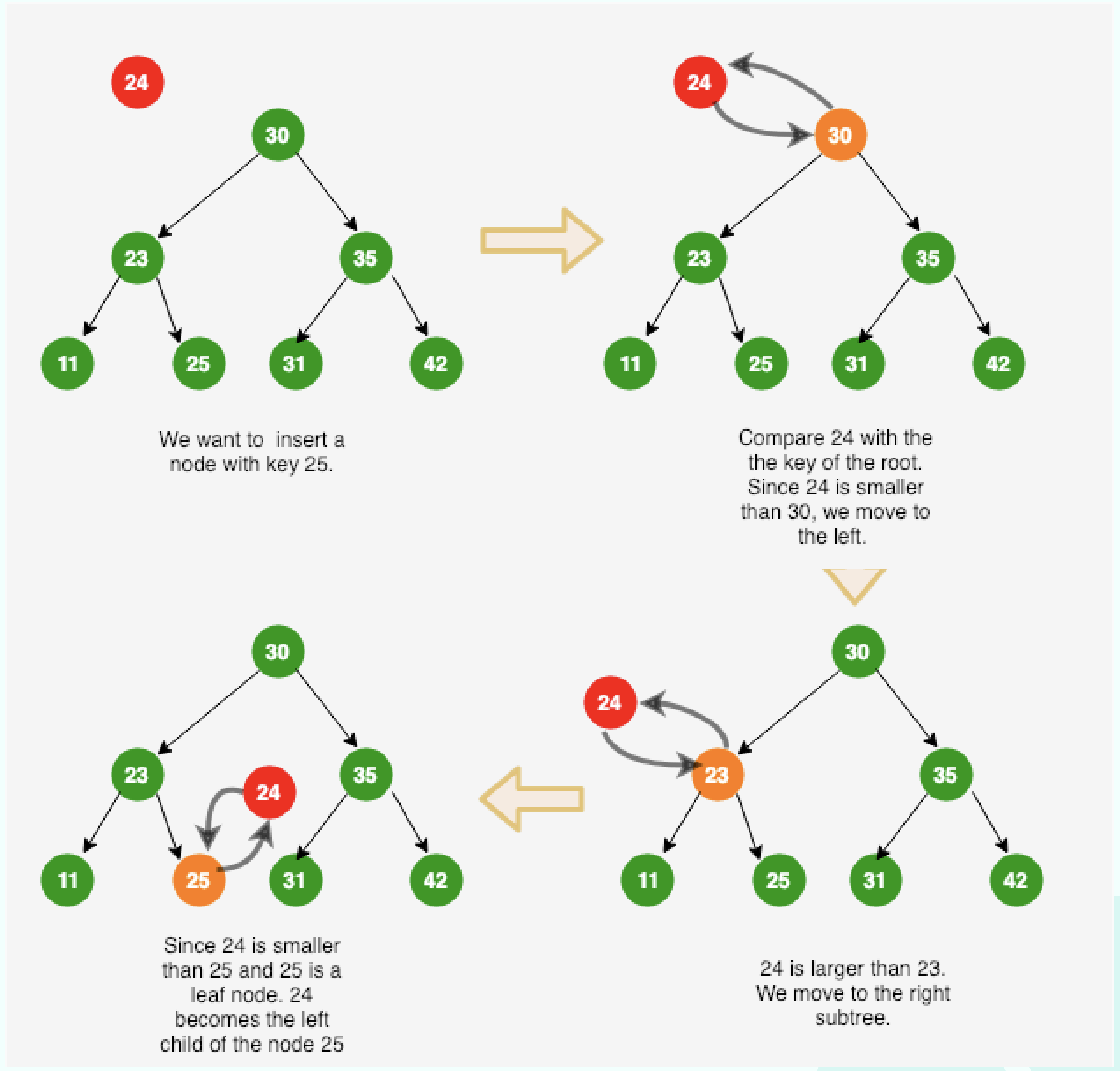
- The left subtree of a node contains only nodes with keys less than the node's key.
- The right subtree of a node contains only nodes with keys greater than the node's key.
- The left and right subtree each must also be a binary search tree.

Basic Operations Binary Search Tree (BST)

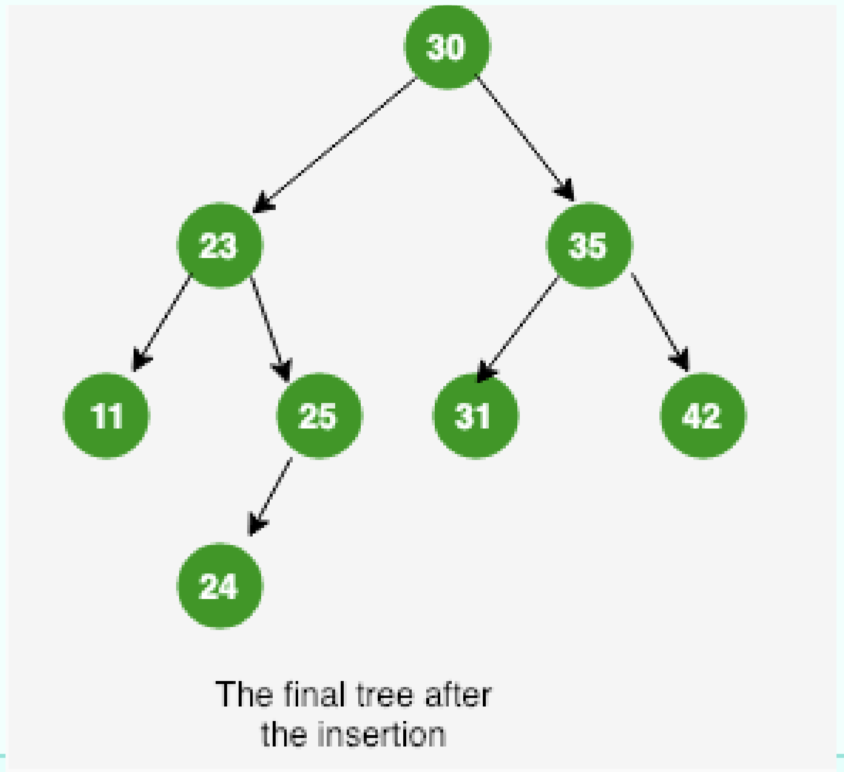
- Insertion in Binary Search Tree
- Searching in Binary Search Tree
- Deletion in Binary Search Tree
- Binary Search Tree (BST) Traversals

Insertion in Binary Search Tree

- Case 1: BST is empty
- Case 2: BST is not empty



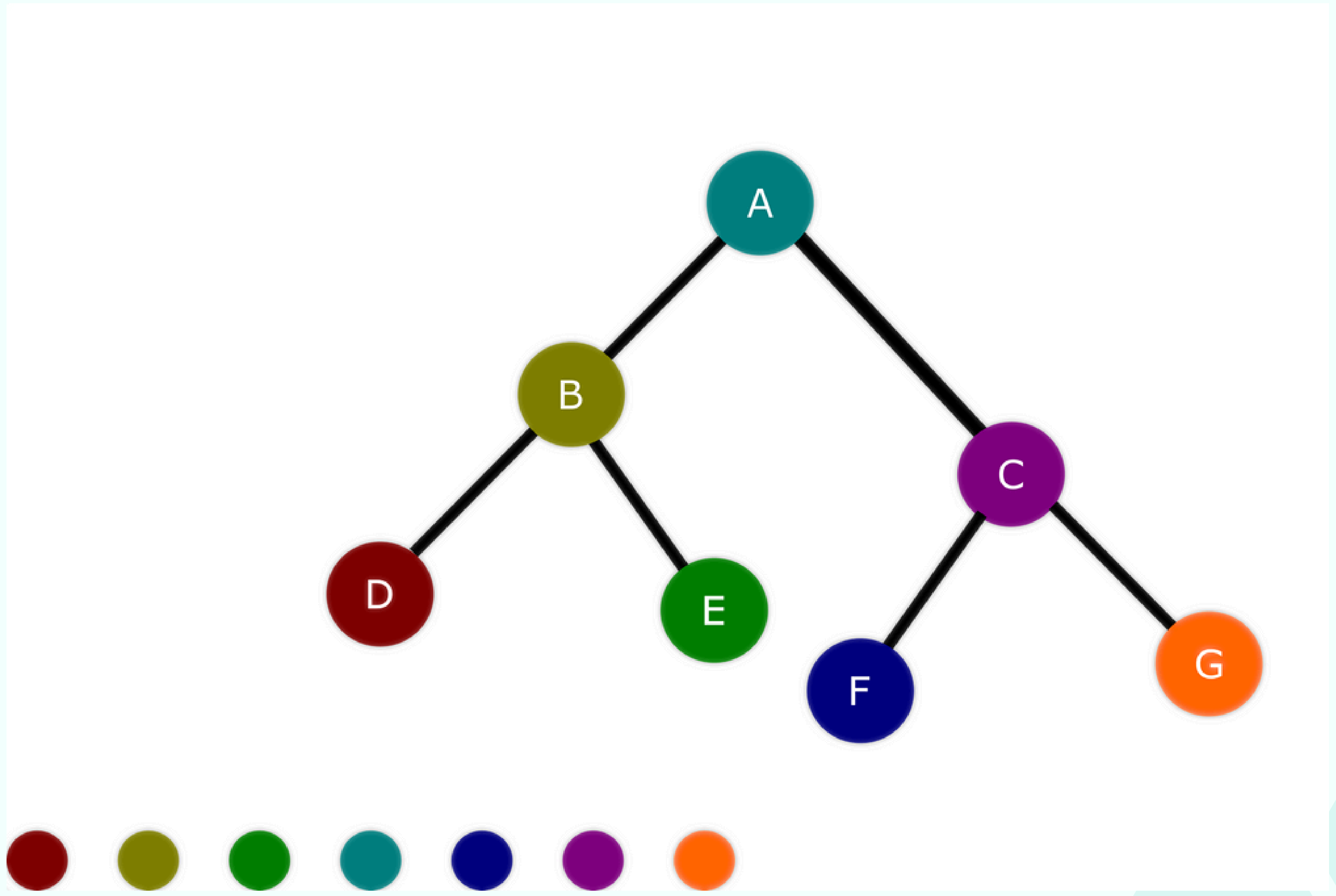
24
30
23
35
11
25
31
42
We want to insert a node with key 25.
Compare 24 with the the key of the root. Since 24 is smaller than 30, we move to the left.
24 is larger than 23. We move to the right subtree.
Since 24 is smaller than 25 and 25 is a leaf node. 24 becomes the left child of the node 25



Binary Search Tree (BST) Traversals

Traversing a tree means visiting and outputting the value of each node in a particular order.

- Inorder => Left, Node, Right.
- Preorder => Node, Left, Right.
- Post order => Left, Right, Node.

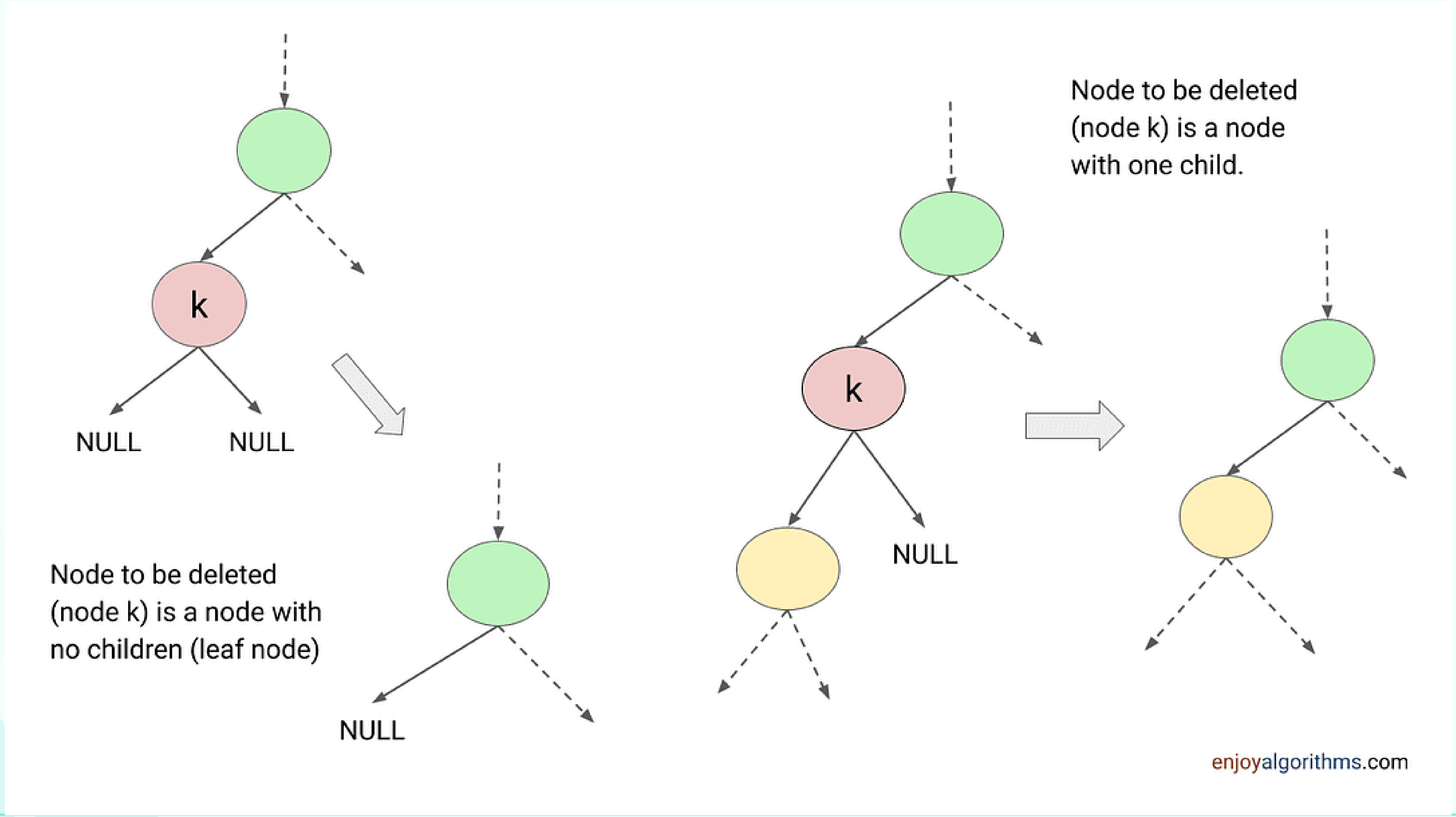


Deletion in Binary Search Tree

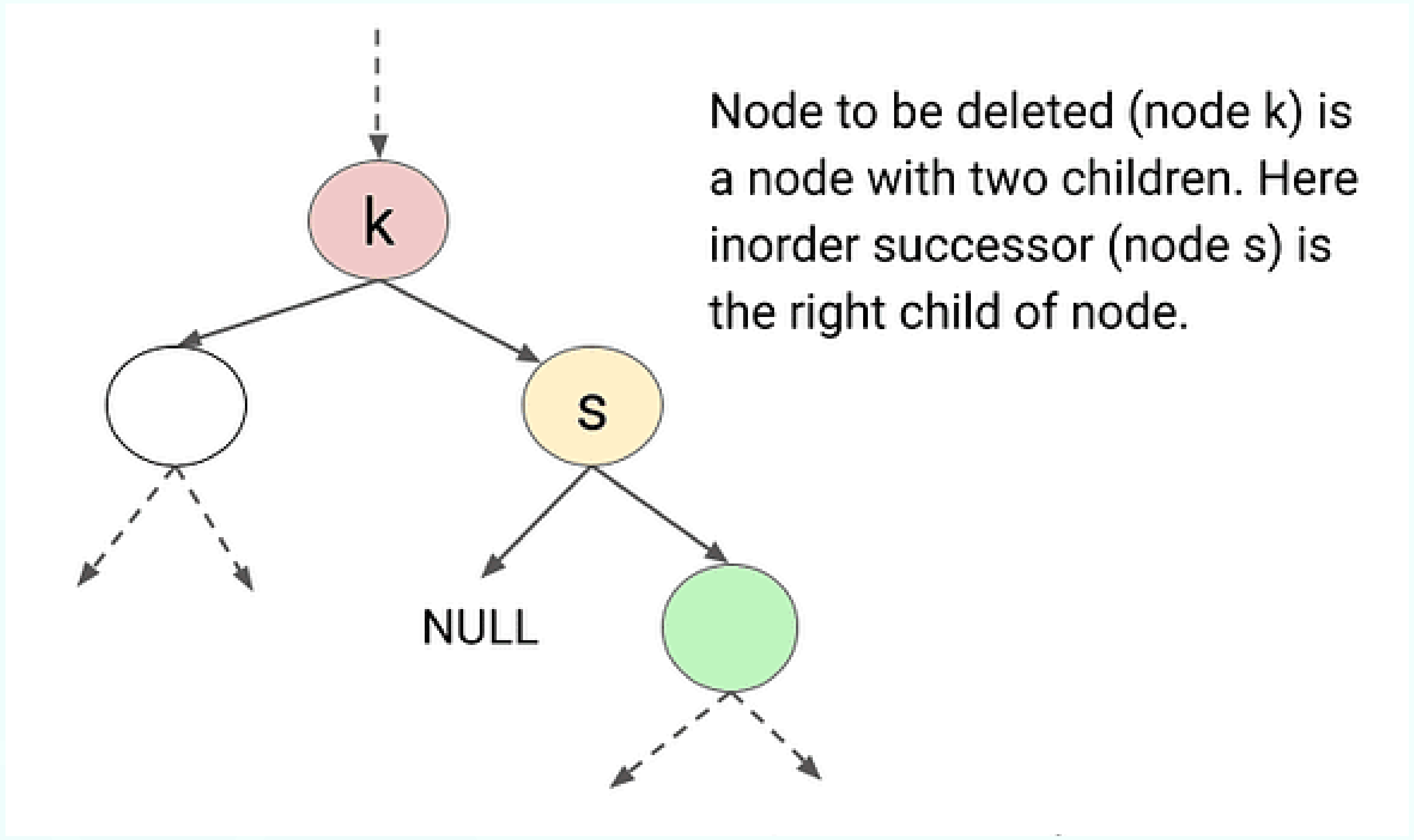
- To delete a node in a BST, we need to:
  - first search for that node.
  - check if there are any nodes present in the left and right subtree of that node.
  - If yes, then we need to appropriately link its subtrees back into the tree somewhere else.

- Case 1: Node to be deleted is a node with no children (leaf node)
- Case 2: Node to be deleted is a node with one child
- Case 3: Node to be deleted is a node with two children

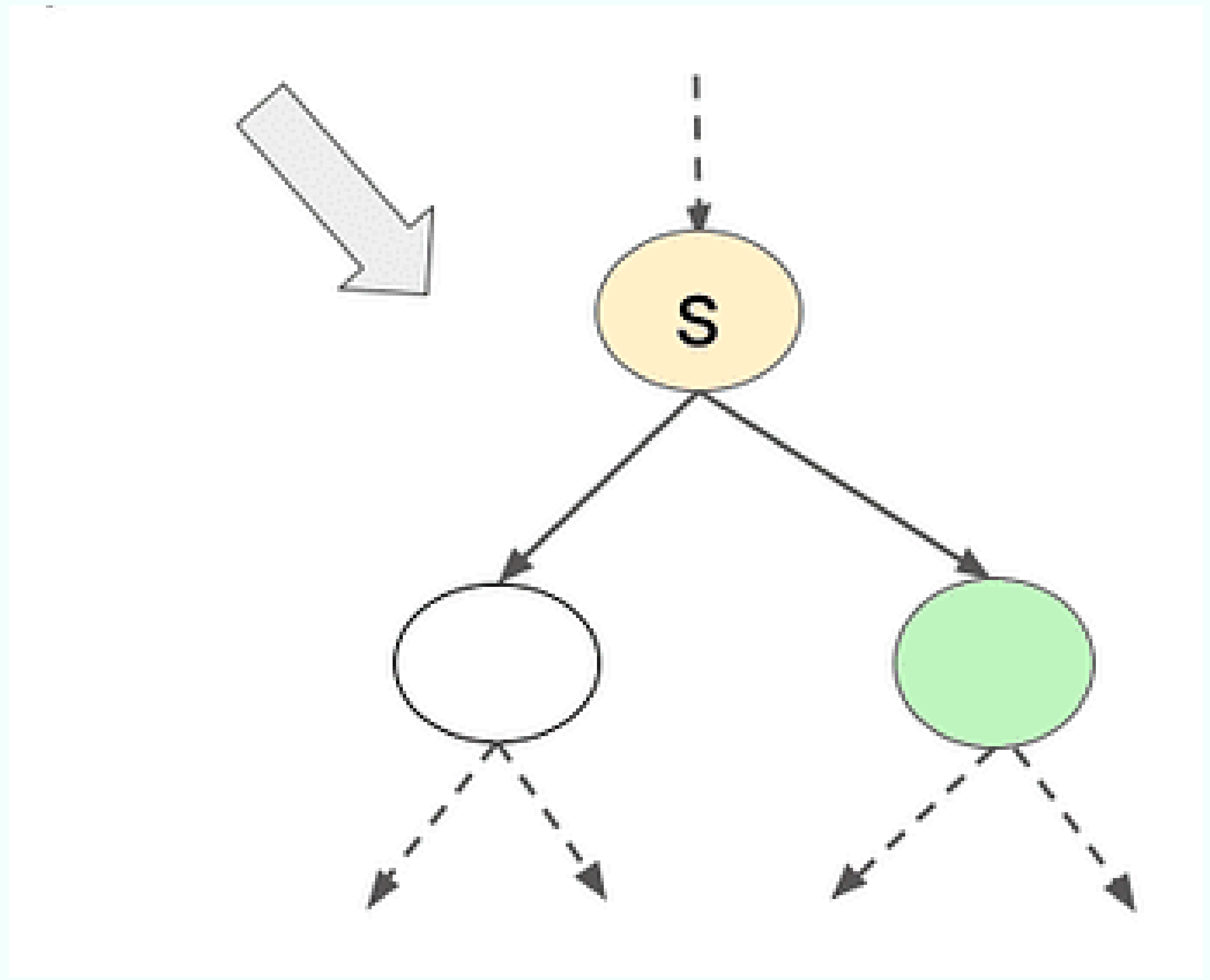
Deletion in Binary Search Tree


k
NULL
NULL
NULL
Node to be deleted (node k) is a node with no children (leaf node)
Node to be deleted (node k) is a node with one child.
k
NULL
enjoyalgorithms.com

Deletion in Binary Search Tree



Node to be deleted (node k) is a node with two children. Here inorder successor (node s) is the right child of node.



Question and Answer

Thank You