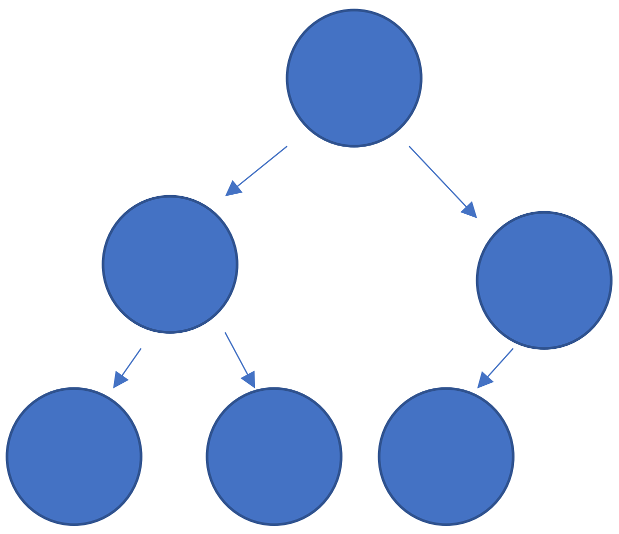
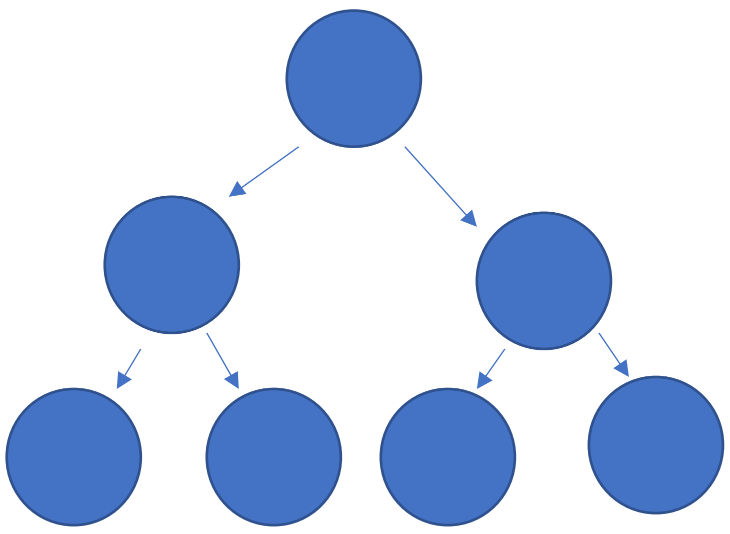
## Binary Trees

- Every node has 0, 1 or 2 children – that's why it's called binary tree
- Children are referred to as left child and right child
- In practice, we use binary search trees
- Binary tree is complete if every level except the last level, has two children
- Full binary tree is complete and every node other than the leaves has to have 2 children
- Binary tree can be incomplete

**Complete**



**Full / Complete**



## Binary Search Tree (BST)

- Can perform insertions, deletions, and retrievals in O(logn) time
- The left child always has a smaller value than its parent
- The right child always has a larger value than its parent
- This means everything to the left of the root is less than the value of the root, and everything to the right of the root is greater the value of the root
- Because of that, we can do a binary search
- Binary Tree usually don't support to have duplicate values. If we want to store duplicate values we'll have to choose to store these values always to the left or the right or have a counter in each node
- If we insert a sorted data in a binary tree we will end up with a linked list which is not a good situation. If we search this tree we'll not going to get O(logn), but linear time O(n)

There are self-balancing trees like **AVL tree** or **red-black tree**