**Linear vs nonlinear**

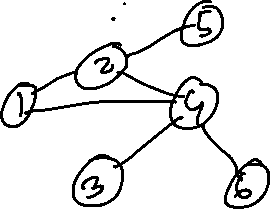
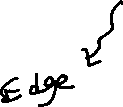
A linear data structure stores its elements in a linear or sequential manner, such as an array or linked list, stack, queue, dequeue where each element has a unique predecessor and successor. A non-linear data structure stores its elements in a non-sequential manner, such as a tree or graph, where elements may have multiple predecessors and/or successors. Linear structures are simpler to implement and access, while non-linear structures allow for more complex relationships between elements.

\*Most popular binary tree where each node has at most two children. 80% of tree uses binary tree like: Binary search, heap, AVL tree, red black tree, treat, spray tree, segment tree.



**Graph:**

1. Node/vertex-> 1,2,3,4,5,6
2. Edge: connect two nodes. 1-2,2-5 etc

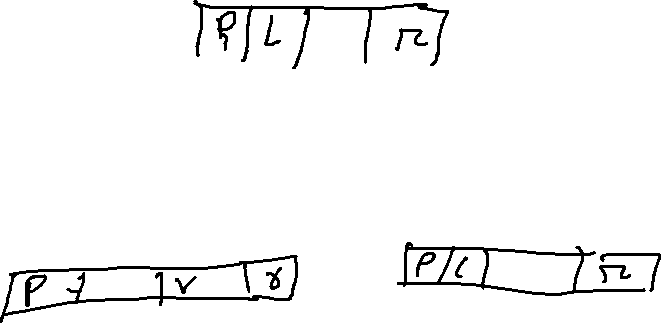
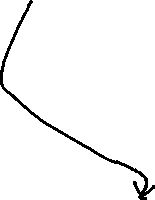
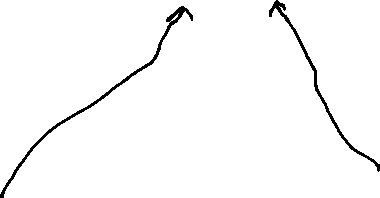
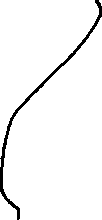
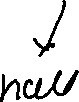
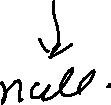
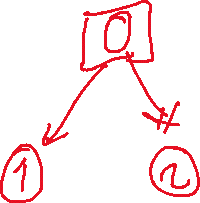
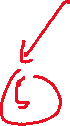
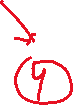
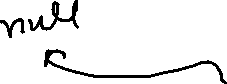
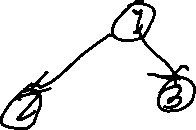
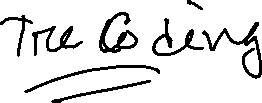
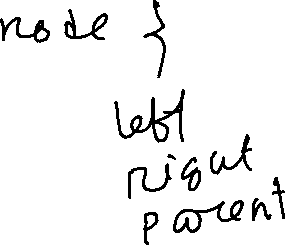


Application:

1. Shortest path

Cycle: 1->2->4->1

1. Tree: Acyclic Graph means does not contain any cycle. Every node has a parent, but root does not parent. Tree is one kind of graph.
2. Binary Tree: Has at most two child. Left child and right child.
3. Full binary tree: Every node has 0 or 2 child
4. Complete Binary tree: all level filled up without last level
5. Perfect Binary tree: All levels are filled.



Binary Tree:

Traversal of Binary Tree:

1. BFS: Bread First search. Level wise visit. It creates a queue
2. DFS: Depth first search. Such as: in order, pre order, post order

