DAR ES SALAAM INSTITUTE OF TECHNOLOGY

GROUP ASSIGMENT 1

GROUP NAME:DATA SCIENTISTS

PROGRAMME: ORDINARY DIPLOMA IN COMPUTER ENGINEERING

MODULE: FUNDAMENTAL OF DATA STRUCTURE AND ALOGARITHMS FOR TECHNICHIANS

CLASS: OD22COE

NO.	GROUP MEMBERS	REGISTRATION NUMBER
1	DEOGRATIUS FESTO AKARO	220222481463
2	EMMANUEL HURUMA	200220228460
3	DANSTAN FELIX MWINUKA	220222448637

A tree is a hierarchical data structure that consists of nodes, with a single node called the root, from which all other nodes branch out. Each node in a tree can have zero or more child nodes. Key characteristics of trees include:

1. Root Node: The topmost node in a tree.

2. Parent Node: A node that has one or more child nodes.

3. Child Node: A node that has a parent node above it.

4. Leaf Node: A node that does not have any child nodes.

5. Subtree A tree formed by a node and its descendants.

6. Depth: The length of the path from the root to a node.

7. Height: The length of the path from a node to the deepest leaf.

Types of Trees

1. Binary Tree: Each node has at most two children, referred to as the left child and the right child.

2. Binary Search Tree (BST): A binary tree where the left child contains only nodes with values less than the parent node, and the right child only nodes with values greater than the parent node.

3. AVL Tree: A self-balancing binary search tree where the difference between heights of left and right subtrees cannot be more than one for all nodes.

4. Red-Black Tree: A self-balancing binary search tree where each node contains an extra bit for denoting the color of the node, either red or black, to ensure the tree remains balanced during insertions and deletions.

5. B-tree: A self-balancing search tree in which nodes can have more than two children, designed to work well on systems that read and write large blocks of data.

Common Operations

- Insertion: Adding a node to the tree.

- Deletion Removing a node from the tree.

- Traversal: Visiting all the nodes in some order.

  - In-order Traversal: Left, Root, Right

  - Pre-order Traversal: Root, Left, Right

  - Post-order Traversal: Left, Right, Root

  - Level-order Traversal: Visiting nodes level by level.

Graphs

A graph s a collection of nodes (also called vertices) and edges (also called arcs) that connect pairs of nodes. Graphs are used to represent networks, such as social networks, computer networks, and transport networks.

Types of Graphs

1. Undirected Graph: An edge between two nodes does not have a direction.

2. Directed Graph (Digraph): An edge between two nodes has a direction, from one node to another.

3. Weighted Graph: Edges have weights, which could represent distances, costs, or any quantitative measurement.

4.Unweighted Graph: Edges do not have weights.

5. Cyclic Graph: Contains at least one cycle (a path that starts and ends at the same node).

6. cyclic Graph: Does not contain any cycles.

7. Connected Graph: There is a path between every pair of nodes.

8. Disconnected Graph: There is at least one pair of nodes without a path between them.

9. Tree: A special case of a connected, acyclic undirected graph.

Common Representations

1. Adjacency Matrix: A 2D array where a cell at row I and column j indicates the presence (and possibly the weight) of an edge between nodes I and j.

2.Adjacency List: An array of lists. The array index represents a node, and the list at that index contains all adjacent nodes.

3. Edge List: A list of all edges in the graph, where each edge is represented as a pair (or tuple) of nodes (and possibly the weight).

Common Algorithms

1. Depth-First Search (DFS): Explores as far down a branch as possible before backtracking.

2. Breadth-First Search (BFS): Explores all neighbors of a node before moving to the next level.

3. Dijkstra's Algorithm: Finds the shortest path from a source node to all other nodes in a weighted graph.

4. Bellman-Ford Algorithm: Computes shortest paths from a single source node to all other nodes in a graph that may have negative weight edges.

5. Floyd-Warshall Algorithm: Finds shortest paths between all pairs of nodes.

6.Kruskal's Algorithm: Finds the minimum spanning tree for a graph by adding edges in increasing weight order.

7. Prim's Algorithm: Finds the minimum spanning tree for a graph by building it out from a single starting node.