Data Structure Workouts

1. Learn the concepts of Tree. Complete at least three sample workouts.
2. Learn the concepts of Binary Search Tree. Complete at least three sample workouts.
   Example:
   a. Create a Binary Search Tree with insertion, contains, delete, three traversals (postorder, preorder, in order).
   b. Find the closest value to a given number in a Tree.
   c. Validate whether a given tree is BST or not.
3. Learn the concepts of Heap. Complete at least three sample workouts.
   Example:
   a. Create a min heap & max heap with build, insert, remove.
4. Learn the concept of Heap sort. Complete at least three sample workouts
5. Learn the concepts of Trie. Complete at least 3 sample workouts.
6. Learn the concepts of Graph. Complete at least three sample workouts.
7. Learn the concepts of Graph traversals (BFS, DFS).
8. Do at least 3 problems each for every structure from any competitive coding websites
9. Learn about the applications of all structures you covered this week

Write a short description about this task

*A **tree** is a hierarchical data structure that consists of nodes connected by edges or branches. It resembles a tree in real life, where the nodes represent the various parts of a tree, while the edges represent the branches.*

In a tree, there is a single node called the root node that has no parent. The root node is the topmost node in the tree. The other nodes in the tree are called child nodes, and they are connected to their parent nodes by edges.

Each node in a tree can have zero or more child nodes, and each child node can have its child nodes. A node with no child nodes is called a leaf node. Nodes that are not leaf nodes are called internal nodes.

Link to the folder containing code and screenshot of the output

https://drive.google.com/drive/folders/1D0rjP5yfPo7RGc2lBqgFJbX_RVYnmxf-?usp=share_link

Write a short description about this task

*A **Binary Search Tree(BST)** follows some order to arrange the elements. In a Binary search tree, the value of the left node must be smaller than the parent node, and the value of the right node must be greater than the parent node. This rule is applied recursively to the left and right subtrees of the root.*

Link to the folder containing code and screenshot of the output

https://drive.google.com/drive/folders/191dRGKllqBE157LnpG2DPMaX781_Bgju?usp=share_link

Write a short description about this task

*A **Heap** is a special Tree-based data structure in which the tree is a complete binary tree. Generally, Heaps can be of two types:*

***Max-Heap**: In a Max-Heap the key present at the root node must be greatest among the keys present at all of it's children. The same property must be recursively true for all sub-trees in that Binary Tree.*

***Min-Heap**: In a Min-Heap the key present at the root node must be minimum among the keys present at all of it's children. The same property must be recursively true for all sub-trees in that Binary Tree.*

Link to the folder containing code and screenshot of the output

https://drive.google.com/drive/folders/1l9hScXw3h2UrfvMpK4hUzfA1vs3_SuKs?usp=share_link

Write a short description about this task

***Heap sort** is a comparison-based sorting algorithm that uses a binary heap data structure.*

The basic idea of heap sort is to turn the input array into a binary heap data structure, where the largest element is at the root. Then, the largest element is removed from the root and placed at the end of the array. This process is repeated for the remaining elements in the heap until all elements have been removed.

Link to the folder containing code and screenshot of the output

https://drive.google.com/drive/folders/1l9hScXw3h2UrfvMpK4hUzfA1vs3_SuKs?usp=share_link

Write a short description about this task

Trie *is a tree-based data structure that is used to store and retrieve data quickly. It is also known as a prefix tree or digital tree. The word "trie" comes from the word "retrieval," which is the primary function of this data structure.*

Trie data structure is used for fast retrieval of key-value pairs, where keys are usually strings. The tree structure of a trie is such that each node represents a single character of the key. The root node of the trie represents an empty string, and each child node represents a character in the key.

One of the significant advantages of using trie data structure is that it can perform prefix matching in linear time. That is, given a prefix, it can efficiently find all keys that start with that prefix.

Link to the folder containing code and screenshot of the output

https://drive.google.com/drive/folders/1Bqh_vrBg2Sda1tI6YkI0f4glyoIsIRMS?usp=share_link

---

Write a short description about this task

A ***Graph*** *is a non-linear data structure consisting of vertices and edges. The vertices are sometimes also referred to as nodes and the edges are lines or arcs that connect any two nodes in the graph.*

Two types of graph representations are adjacency matrix and adjacency list.

Two types of traversal used are Breadth First Search and Depth First Search.

Link to the folder containing code and screenshot of the output

https://drive.google.com/drive/folders/1B5Mx41GiZRxIYPDdSoOKvgQAIVyh1ppS?usp=share_link

---

Write a short description about this task

Graph can be traversed by using the algorithms BFS and DFS.

Graph traversal is a common operation in computer science where you visit every vertex (node) and edge (link) in a graph data structure. There are two popular methods for traversing a graph: breadth-first search (BFS) and depth-first search (DFS).

Breadth-First Search (BFS)

BFS is a graph traversal algorithm that starts at the root node and visits all the nodes at the current depth before moving on to the next level. It visits all the nodes of the same level before moving on to the next level. This algorithm uses a queue data structure to store the nodes that need to be visited.

Depth-First Search (DFS)

DFS is another popular graph traversal algorithm that starts at the root node and explores as far as possible along each branch before backtracking. This algorithm uses a stack data structure to store the nodes that need to be visited.

Link to the folder containing code and screenshot of the output

https://drive.google.com/drive/folders/1B5Mx41GiZRxIYPDdSoOKvgQAIVyh1ppS?usp=share_link

Write a short description about this task

1. *Trees are a fundamental data structure in computer science with a wide range of applications. Here are some common applications of trees:*

File Systems

Network Routing

Database Systems

2. *Some of the applications of binary search trees:*

Searching

Insertion and Deletion

Dictionary

Balancing

3. *Some applications of the heap data structure:*

Heap Sort

Priority Queue

Dijkstra's Algorithm

Huffman Coding

4. *One of the main applications of Trie are;*

Autocomplete functionality(suggestions)

Spell-checking and dictionary applications

5. Some common applications of graphs:

Social Networks

Transportation Networks

Internet Networks

Computer Networks

Financial Networks: