**Data Structure and Algorithm**

1. What is a data structure? What are some common data structures?

A data structure is an organized way of collecting data so that it becomes easy to access the required data in a short period of time.

Some common Data Structures are:

Stack, Queue, Linked List, Tree, Graph etc.

1. What is an algorithm? What are some common algorithm design techniques?

An algorithm is a step to solve a particular problem.

Some common algorithm are:

Binary Search Algorithm , Quick sort algorithm, Merge Sort Algorithm etc

1. What is the time complexity of an algorithm? How is it calculated?

Time complexity of an algorithm refers to how long it will take to execute an algorithm based on size.

It is calculated by time complexity and space complexity.

1. What is the space complexity of an algorithm? How is it calculated?

This evaluates the memory storage requirements of an algorithm. It tells us how much memory the algorithm consumes as a function of the input size.

1. What is the difference between an array and a linked list? When would you use one over the other?

**Data Storage**

Array - Stored contiguous memory allocation

Linked list - Not stored in contiguous memory allocation

**Flexibility**

Array - Fixed size

Linked list - Add or remove elements, not fixed size

Use Cases:

Arrays: When you need fast access by index and know the maximum size in advance.

Linked Lists: When you require dynamic size, frequent insertions/deletions, or don’t know the exact size beforehand.

1. -What is a stack? How is it implemented? What are some common use cases?

A stack is a data structure that follows the last in first out method. For instance a deck of cards, stack of plates etc.

Common use cases are Call logs, text editors etc.

1. What is a queue? How is it implemented? What are some common use cases?

A Queue is a data structure that follows the first in first out method. E.g;

Ticket line, traffic jam etc

1. What is a binary tree? How is it implemented? What are some common traversal algorithms?

A binary tree is a hierarchical data structure where each node has at most two children: a left child and a right child.

A binary tree can be implemented using a node-based approach.

Some common traversals are Common Traversal Algorithms, Inorder Traversal, Postorder Traversal etc.

1. What is a hash table? How is it implemented? What are some common use cases?

A hash table allows you to store data in a key-value format with direct access to its items in constant time.

Common Use Cases for Hash Tables: Caching, Database indexing, Symbol tables

1. What is a graph? How is it implemented? What are some common traversal algorithms?

A graph is a non-linear data structure composed of vertices (also known as nodes) and edges.

There are two techniques to make a graph: Adjacency Matrix, Adjacency list

1. What is dynamic programming? What are some common problems that can be solved using dynamic programming?

Dynamic programming is an optimization method used to solve recursive algorithms, particularly for computing or mathematical problems. It breaks down complex problems into simpler sub-problems and stores their results to avoid redundant computations.

Here are some common problems that can be effectively solved using dynamic programming: Knapsack Problem, Longest Common Subsequence, Shortest Common Supersequence etc

1. What is recursion? How does it work? What are some common problems that can be solved using recursion?

Recursion refers to the process in which a function calls itself directly or indirectly.

Recursive functions break down a problem into smaller subproblems.

Each recursive call works on a simpler version of the original problem.

Common Problems Solved Using Recursion:

Towers of Hanoi, Tree Traversals, Fibonacci Sequence etc