

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

A data structure is a way of organizing and storing data in a computer so that it can be accessed and manipulated efficiently.

Some common data structures include:

- Arrays
- Linked Lists
- Stacks
- Queues
- Trees
- Hash Tables
- Graphs

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

An algorithm is a step-by-step procedure or set of rules for solving a computational problem.

Common algorithm design techniques include:

- Divide and Conquer
- Greedy Algorithms
- Dynamic Programming
- Backtracking

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

Time complexity of an algorithm represents the amount of time taken by an algorithm to run as a

function of the length of the input.

It's typically calculated using Big O notation, which describes the upper bound of the time taken.

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

Space complexity of an algorithm represents the amount of memory space required by an algorithm to run as a function of the length of the input.

It's also typically calculated using Big O notation.

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

An array stores elements of the same data type sequentially in memory, allowing constant-time

access to elements by index. A linked list consists of nodes where each node contains a data

field and a reference to the next node in the sequence. Arrays offer faster access to elements by

index, while linked lists offer faster insertion and deletion at arbitrary positions.

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

A stack is a linear data structure that follows the Last-In, First-Out (LIFO) principle. It is typically

implemented using arrays or linked lists.

Common use cases include expression evaluation, function call management, and

Backtracking.

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

A queue is a linear data structure that follows the First-In, First-Out (FIFO) principle. It is typically implemented using arrays or linked lists.

Common use cases include job scheduling, breadth-first search, and implementing caches.

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, referred to as the left child and the right child. It is implemented using nodes with references to

their children.

Common traversal algorithms include in-order, pre-order, and post-order traversal.

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

A hash table is a data structure that stores key-value pairs and provides efficient insertion, deletion, and lookup operations. It is implemented using an array combined with a hash function. Common use cases include implementing associative arrays, caching, and symbol Tables.

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

A graph is a collection of nodes (vertices) and edges that connect pairs of nodes. It can be implemented using adjacency matrices or adjacency lists.

Common traversal algorithms include depth-first search (DFS) and breadth-first search (BFS).

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

Dynamic programming is a technique used to solve problems by breaking them down into simpler subproblems and solving each subproblem only once. It's often used for optimization problems and problems with overlapping subproblems.

Common problems include the knapsack problem, longest common subsequence, and Fibonacci sequence generation.

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

Recursion is a programming technique where a function calls itself in order to solve smaller instances of the same problem. Recursion involves two parts: base case(s) and recursive case(s).

Common problems that can be solved using recursion include factorial calculation, fibonacci sequence generation, and tree traversal.