Here are some data structures and sorting algorithms that I'm familiar with:

Data Structures I'm Familiar With:

1. Arrays: Basic structure for storing elements sequentially.

2. Linked Lists: Lists where each element points to the next one.

3. Stacks: Last In, First Out (LIFO) data structure.

4. Queues: First In, First Out (FIFO) data structure.

5. Hash Tables: Key-value store with fast average time complexity for lookups, insertions, and deletions.

6. Binary Trees: Trees where each node has at most two children.

7. Binary Search Trees (BST): Binary trees with the property that for each node, values in its left subtree are less and values in its right subtree are greater.

8. Heaps: Tree-based data structure where the parent nodes are either greater than or equal to (max-heap) or less than or equal to (min-heap) the child nodes.

9. Graphs: Representations of networks consisting of vertices (nodes) and edges (connections).

Sorting Algorithms I'm Familiar With:

1. Bubble Sort: Simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order.

2. Insertion Sort: Builds the final sorted array one item at a time, comparing each new item with already sorted items and inserting it into the correct position.

3. Selection Sort: Divides the input list into two parts: the sublist of items already sorted and the sublist of items remaining to be sorted, and repeatedly selects the smallest element from the unsorted sublist and swaps it into place.

4. Merge Sort: Divide-and-conquer algorithm that divides the input array into two halves, sorts each half recursively and then merges the sorted halves.

5. Quick Sort: Divide-and-conquer algorithm that selects a 'pivot' element and partitions the array into two sub-arrays around the pivot, recursively sorting each sub-array.