**Data Structures**

**Basic Data Structures**

1. Arrays
   * Static and Dynamic Arrays
   * Multidimensional Arrays
2. **🗆** Linked Lists
   * **🗆** Singly Linked List
   * **🗆** Doubly Linked List
   * **🗆** Circular Linked List
3. **🗆** Stacks
   * **🗆** Implementation using Arrays and Linked Lists
   * **🗆** Applications of Stacks (e.g., expression evaluation, backtracking)
4. **🗆**  Queues
   * **🗆** Simple Queue
   * **🗆** Circular Queue
   * **🗆** Priority Queue
   * **🗆** Deque (Double-ended Queue)
5. **🗆** Trees
   * **🗆** Binary Trees
     + **🗆** Properties and Types (Complete, Full, Perfect)
     + **🗆** Traversal Methods (Preorder, Inorder, Postorder, Level Order)
   * **🗆** Binary Search Trees (BST)
     + **🗆** Insertion, Deletion, Search operations
     + **🗆** Self-Balancing Trees (e.g., AVL Tree, Red-Black Tree)
   * **🗆** Heaps
     + **🗆** Min-Heap, Max-Heap
     + **🗆** Heap Operations and Applications (e.g., Heap Sort, Priority Queue)

**Intermediate Data Structures**

1. **🗆** Trie (Prefix Tree)
   * **🗆** Construction, Search, Deletion
   * **🗆** Applications (e.g., Autocomplete, Spell Checker)
2. **🗆** Segment Tree
   * **🗆** Range Query, Point Update
   * **🗆** Lazy Propagation
3. **🗆** Fenwick Tree (Binary Indexed Tree)
   * **🗆** Point Update, Range Query
   * **🗆** Applications
4. **🗆** Graphs
   * **🗆** Representation
     + **🗆** Adjacency Matrix
     + **🗆** Adjacency List
     + **🗆** Edge List
   * **🗆** Graph Traversal
     + **🗆** Depth-First Search (DFS)
     + **🗆** Breadth-First Search (BFS)
   * **🗆** Shortest Path Algorithms
     + **🗆** Dijkstra's Algorithm
     + **🗆** Bellman-Ford Algorithm
     + **🗆** Floyd-Warshall Algorithm
     + **🗆** A\* Search Algorithm
   * **🗆** Minimum Spanning Tree (MST)
     + **🗆** Kruskal's Algorithm
     + **🗆** Prim's Algorithm
   * **🗆** Advanced Graph Algorithms
     + **🗆** Topological Sorting
     + **🗆** Strongly Connected Components (Tarjan's Algorithm)
     + **🗆** Network Flow (Ford-Fulkerson Algorithm, Edmonds-Karp Algorithm)
     + **🗆** Bipartite Graph Checking
5. **🗆** Hashing
   * **🗆** Hash Tables
     + **🗆** Open Addressing (Linear Probing, Quadratic Probing)
     + **🗆** Separate Chaining
   * **🗆** Hash Functions
     + **🗆** Properties and Design
     + **🗆** Collision Resolution Techniques

**Advanced Data Structures**

1. **🗆** Skip List
   * **🗆** Insertion, Deletion, Search
   * **🗆** Applications
2. **🗆** B-Trees and B+ Trees
   * **🗆** Properties and Operations
   * **🗆** Usage in Databases
3. **🗆** Splay Trees
   * **🗆** Splaying, Insertion, Deletion
4. **🗆** Treap
   * **🗆** Priority and BST Property
   * **🗆** Operations and Applications
5. **🗆** Suffix Tree and Suffix Array
   * **🗆** Construction
   * **🗆** Applications in String Matching
6. **🗆** Disjoint Set (Union-Find)
   * **🗆** Union by Rank, Path Compression
   * **🗆** Applications in Kruskal’s Algorithm
7. **🗆** Specialized Data Structures
   * **🗆** Bloom Filter
     + **🗆** Probabilistic Data Structure
     + **🗆** False Positive Rate
   * **🗆** Quad Trees and Octrees
     + **🗆** Spatial Partitioning
     + **🗆** Applications in Graphics and Game Development
   * **🗆** K-D Trees
     + **🗆** Multidimensional Search
     + **🗆** Nearest Neighbor Search
   * **🗆** R-trees
     + **🗆** Spatial Indexing
     + **🗆** Applications in Geographic Information Systems (GIS)
8. **🗆** String Data Structures
   * **🗆** Rabin-Karp Algorithm
     + **🗆** Rolling Hash Technique
   * **🗆** Knuth-Morris-Pratt (KMP) Algorithm
     + **🗆** Prefix Function
   * **🗆** Boyer-Moore Algorithm
     + **🗆** Good Suffix, Bad Character Heuristics
9. **🗆** Concurrency Data Structures
   * ⬜ Concurrent Linked Lists
     + ⬜ Lock-based and Lock-free Implementations
   * **🗆** Concurrent Queues
     + **🗆** Blocking and Non-blocking Queues
   * **🗆** Skip List
     + **🗆** Concurrent Variants

**Sorting Algorithms**

**Comparison-Based Sorting**

1. **🗆** Simple Sorts
   * **🗆** Bubble Sort
   * **🗆** Selection Sort
   * **🗆** Insertion Sort
2. **🗆** Efficient Sorts
   * **🗆** Merge Sort
   * **🗆** Quick Sort
   * **🗆** Heap Sort
3. **🗆** Hybrid Sorts
   * **🗆** Timsort
   * **🗆** IntroSort

**Non-Comparison-Based Sorting**

1. **🗆** Counting Sort
2. **🗆** Radix Sort
3. **🗆** Bucket Sort

**Specialized Sorts**

1. **🗆** Shell Sort
2. **🗆** Pigeonhole Sort

**Searching Algorithms**

**Linear Search**

1. **🗆** Simple Linear Search

**Binary Search**

1. **🗆** Classic Binary Search
2. **🗆** Binary Search Variants

**Hash-Based Search**

1. **🗆** Direct Addressing
2. **🗆** Hash Tables

**Tree-Based Search**

1. **🗆** Binary Search Tree (BST)
2. **🗆** AVL Tree
3. ⬜ Red-Black Tree
4. **🗆** B-Trees and B+ Trees

**Graph-Based Search**

1. **🗆** Depth-First Search (DFS)
2. **🗆** Breadth-First Search (BFS)

**String Search Algorithms**

1. **🗆** Naive String Search
2. **🗆** Knuth-Morris-Pratt (KMP) Algorithm
3. **🗆** Rabin-Karp Algorithm
4. **🗆** Boyer-Moore Algorithm

**Advanced Search Techniques**

1. **🗆** Exponential Search
2. **🗆** Interpolation Search
3. **🗆** Fibonacci Search
4. **🗆** Jump Search

**Certainly! Here’s a concise list of essential data structure topics:**

1. **Array**
2. **String**
3. **LinkedList**
4. **Stack and Queue**
5. **Tree**
6. **Binary Search Tree (BST)**
7. **Graph**
8. **Trie**
9. **Heap**
10. **Hashing**
11. **Recursion, Backtracking, and Dynamic Programming**
12. **Greedy Algorithms**
13. **Sorting and Searching**
14. **Pattern Searching**
15. **Divide and Conquer Algorithms**
16. **Number Theory**
17. **Bit Manipulation**