* **Arrays**
  + **Introduction**
  + **Memory management**
  + **Input and Output**
  + **std::vector**
  + **Searching**
    - **Linear Search**
    - **Binary Search**
    - **Modified Binary Search**
    - **Binary Search on 2D Arrays**
  + **Sorting**
    - **Insertion Sort**
    - **Selection Sort**
    - **Bubble Sort**
    - **Count Sort**
    - **Radix Sort**
    - **Cyclic Sort**
  + **Sliding window**
  + **Two Pointer**
  + **Subarray Questions**
* **Pattern questions**
* **Maths for DSA**
  + **Introduction**
  + **Complete Bitwise Operators**
  + **Prime numbers**
  + **HCF / LCM**
  + **Sieve of Eratosthenes**
  + **Newton's Square Root Method**
  + **Number Theory**
  + **Euclidean algorithm**
  + **Advanced Concepts for CP (later in the course)**
    - **Bitwise + DP**
    - **Extended Euclidean algorithm**
    - **Modulo Properties**
    - **Modulo Multiplicative Inverse**
    - **Linear Diophantine Equations**
    - **Fermat’s Theorem**
    - **Wilson's Theorem**
    - **Lucas Theorem**
    - **Chinese Remainder Theorem**
* **Recursion**
  + **Introduction**
  + **Why recursion?**
  + **Flow of recursive programs - stacks**
  + **Convert recursion to iteration**
  + **Tree building of function calls**
  + **Tail recursion**
  + **Sorting:**
    - **Merge Sort**
    - **Quick Sort**
  + **Backtracking**
    - **Sudoku Solver**
    - **N-Queens**
    - **N-Knights**
    - **Maze problems**
  + **Recursion String Problems**
  + **Recursion Array Problems**
  + **Recursion Pattern Problems**
  + **Subset Questions**
* **Space and Time Complexity Analysis**
  + **Introduction**
  + **Comparisons of various cases**
  + **Solving Linear Recurrence Relations**
  + **Solving Divide and Conquer Recurrence Relations**
  + **Big-O, Big-Omega, Big-Theta Notations**
  + **Get equation of any relation easily - best and easiest approach**
  + **Complexity discussion of all the problems we do**
  + **Space Complexity**
  + **Memory Allocation of various languages**
  + **NP-Completeness and Hardness**
* **Greedy Algorithms**
* **Stacks & Queues**
  + **Introduction**
  + **Interview problems**
  + **Push efficient**
  + **Pop efficient**
  + **Queue using Stack and Vice versa**
  + **Circular Queue**
* **Linked List**
  + **Introduction**
  + **Fast and slow pointer**
  + **Cycle Detection**
  + **Single and Doubly LinkedList**
  + **Reversal of LinkedList**
* **Dynamic Programming**
  + **Introduction**
  + **Recursion + Recursion DP + Iteration + Iteration Space Optimized**
  + **Complexity Analysis**
  + **0/1 Knapsack**
  + **Subset Questions**
  + **Unbounded Knapsack**
  + **Subsequence questions**
  + **String DP**
* **Trees**
  + **Introduction**
  + **Binary Trees**
  + **Binary Search Trees**
  + **DFS**
  + **BFS**
  + **AVL Trees**
  + **Segment Tree**
  + **Fenwick Tree / Binary Indexed Tree**
* **Square Root Decomposition**
* **Heaps**
  + **Introduction**
  + **Theory**
  + **Priority Queue**
  + **Heapsort**
  + **Two Heaps Method**
  + **k-way merge**
  + **Top k elements**
  + **Interval problems**
* **HashMap**
  + **Introduction**
  + **Theory - how it works**
  + **Comparisons of various forms**
  + **Limitations and how to solve**
  + **Map using LinkedList**
  + **Map using Hash**
  + **Chaining**
  + **Probing**
  + **Huffman-Encoder**
* **Tries**
* **Graphs**
  + **Introduction**
  + **BFS**
  + **DFS**
  + **Working with graph components**
  + **Minimum Spanning Trees**
  + **Kruskal Algorithm**
  + **Prims Algorithm**
  + **Dijkstra’s shortest path algorithm**
  + **Topological Sort**
  + **Bellman ford**
  + **A\* pathfinding Algorithm**