**DSA Basics checklist**

***Structures***

1. Arrays
   * *Traversal*
   * *searching*
     1. *Linear search*
     2. *Binary search*
   * *Sorting*
     1. *Bubble sort*
     2. *Insertion sort*
     3. *Selection sort*
     4. *Merge sort*
     5. *Quick sort*
     6. *Count sort*
     7. *Radix sort*
     8. *Heap sort*
   * *Reversing*
   * *Kadane’s Algorithm*
2. Matrix
   * *Traversal*
     1. *Row-wise*
     2. *Column-wise*
     3. *Spiral traversal*
   * *Multiply 2 matrices*
   * *Find transpose*
   * *Binary search in sorted matrix*
3. Strings
   * *Traversal*
   * *Reversing*
   * *Is-palindrome*
   * *Is-anagram*
4. Linked List
   * *Creating*
   * *Traversal*
   * *Reversing*
   * *Finding mid (Slow-fast pointer)*
   * *Searching*
   * *Sorting (merge sort)*
   * *Insertion/Deletion of nodes*
   * *Circular & doubly LLs*
   * *Detect Cycle (Floyd’s Cycle Detection)*
   * *Find insertion of two LLs*
5. Stack
   * *Implementation*
     1. *With array*
     2. *With Linked list*
   * *Next greater element*
   * *Get bottom-most element*
   * *Next greater element*
   * *Valid parenthesis*
   * *Reverse stack*
6. Queue
   * *Implementation*
     1. *With array*
     2. *With Linked list*
   * *Circular Queue, Double ended queue*
   * *Priority queue*
   * *Implementing stack using queue & vice versa*
7. Map (or hashmaps)
   * *Implementation*
8. Binary tree
   * *Creating*
   * *Traversal*
     1. *Pre-order (DFS)*
     2. *Post-order (DFS)*
     3. *In-order (DFS)*
     4. *Level-order (BFS)*
   * *Searching*
   * *Find depth/height of tree*
   * *Count the nodes*
   * *Diameter of tree*
   * *Lowest common ancestor*
   * *Symmetric tree*
   * *Are-identical*
   * *Zig-zag traversal*
   * *Convert tree to linked-list (level-wise)*
9. Binary search tree
   * *Creating*
   * *Insert/delete nodes*
   * *Validate a BST*
   * *Convert sorted array into BST*
   * *Make balanced tree/AVL tree*
10. Heaps
    * *Max & Min heap*
    * *Insert/delete*
    * *Heapify*
11. Tries
    * *Implementation*
    * *Longest common prefix*
12. Graphs
    * *Creating*
    * *Traversal*
      1. *DFS*
      2. *BFS*
    * *Has-path*
    * *Searching*
    * *Detect Cycle*
    * *Topological sort*
      1. *Using DFS*
      2. *Kahn’s algorithm (BFS)*
    * *Bipartite graph*
    * *Number of connected components*
    * *Shortest path*
      1. *Djikstra’s algorithm*
      2. *Bellman-ford algorithm*
      3. *Floyd warshall algorithm*
    * *Minimum Spanning tree:*
      1. *Krushal’s algorithm*
      2. *Prim’s algorithm*
13. Segment trees
    * *Constructing*

***Algorithms/Patterns***

1. Two pointer
   * *Two sum (sorted array)*
   * *Checking for palindrome*
   * *Container with most water*
   * *Buy N sell stocks*
2. Binary Search
   * *Search in a rotated sorted array*
3. Bit Manipulation
   * *Check if a number is power of 2*
   * *Brian kernighan’s algorithm*
4. Backtracking/recursion
   * *Factorial & Fibonacci*
   * *N-Queens*
   * *Word search*
   * *Find permutations*
5. Divide and Conquer
   * *Merge & quick sort*
6. Greedy
   * *Activity selection Problem*
   * *Job scheduling*
   * *Merge intervals*
7. Dynamic programming
   * *Fibonacci*
   * *0-1 knapsack*
   * *Unbounded knapsack*
   * *Longest common subsequence*
   * *Kadane’s algorithm*
   * *Catalan number*
   * *DP on matrices*