# **Interview Topic -**

#### 1. DSA-

- A. Array
  - 1) Math
  - 2) sorting (bubble,insertion, Selection, Merge, Quick, Heap)
  - 3) searching
- B. String
  - 1) frequency\_array[26]
  - 1) pattern matching->KMP/robin karp
- C. Sliding window / two pointer /prefix sum/ robin\_karp(rolling hash)
- D. Link List
- E. Stack /monotonic stack
- F. Queue
- G. Binary Search
- H. Bit Manipulation
- I. Recursion /Backtracking
- J. Greedy Algorithms
- K. Dynamic Programming
  - 1) Knapsack (0/1, bounded, unbounded)
  - 2) 1D DP( fibonacci ,min step , Kadane algo(largest sum contiguous subarray))
  - 3) LCS (longest common subsequence)
  - 4) LIS (longest increasing subsequence)
  - 5) MCM (matrix chain multiplication)
  - 6) 2D DP(grid dp)
  - 7) Advance DP(dp+binary search+tree+geometry+3D+bitmasking+graph+etc)
- L. Tree
  - 1) traversal(BFS/DFS)
  - 2)Binary Tree
  - 3)Binary\_Search\_Tree
  - 4)AVL Tree
  - 5) Red-Black Tree
  - 6) B-Tree
- M. Hashing /Heap (Priority\_Queue)

- N. Graph (bfs,dfs,toposort, shortest\_path/cost)
- O. Minimum\_Spanning\_Tree (MST)
- P. Disjoint Set /Union (DSU)
- Q. Trie Data Structure (TDS)

### R. etc/cp

- 1) Number Theory /Modular\_Arithmetics
- 2) Square root decomposition
- 3) Segment Tree
- 4) Fenwick Tree
- 5) Policy based Data Structure (PBDS)

### 2. Theory Subjects

- A. Operating System (Linux/ Windows)
- B. DBMS (SQL/NoSQL command)
- C. Computer Network

## 3. System Design

- A. Low Level Design (Design Patterns/ OOPs Concepts /Solid Principles /UML)
- B. High Level Design

(Scalability/ Performance/ Latency and Throughput /Consistency / Availability /Partition Tolerance /CAP Theorem /Domain Name System /Content Delivery Network /Load Balancers and Reverse Proxy /Microservices /Databases /Caching /Message Queues

## 4. Work Experience /Technology Domain

# For Vikas-> Full Stack Developer (MERN Stack)

- A. Project Introduction
- B. Modules description /Functionality
- C. Tools /Technology /Platform

Refer to - interviewBit