**✅ Operating System Topic-Wise List**

**1️⃣ Basics of Operating Systems**

* What is an operating system?
* Functions of OS
* Types of operating systems (batch, time-sharing, distributed, real-time, embedded)
* System calls
* Kernel vs. user mode
* Architecture of OS (monolithic, microkernel, layered)

**2️⃣ Process Management**

* Process vs. program
* Process states & PCB (Process Control Block)
* Context switching
* Process scheduling (long-term, short-term, medium-term)
* Scheduling algorithms (FCFS, SJF, Round Robin, Priority, Multilevel queue)
* Threads (user-level vs. kernel-level)
* Multithreading models

**3️⃣ Inter-Process Communication (IPC)**

* Shared memory
* Message passing
* Pipes, FIFOs
* Semaphores
* Monitors
* Signals

**4️⃣ Synchronization**

* Critical section problem
* Race conditions
* Classical synchronization problems (Dining Philosophers, Producer–Consumer, Readers–Writers)
* Locks, Mutexes
* Peterson’s algorithm
* Monitors and semaphores

**5️⃣ Deadlocks**

* Deadlock conditions
* Resource Allocation Graph
* Deadlock prevention
* Deadlock avoidance (Banker’s algorithm)
* Deadlock detection and recovery

**6️⃣ Memory Management**

* Logical vs. physical address
* Paging
* Segmentation
* Fragmentation (internal, external)
* Virtual memory
* Page replacement algorithms (LRU, FIFO, Optimal, Clock)
* Thrashing

**7️⃣ File Systems**

* File concepts
* File attributes
* Directory structures (single-level, two-level, tree, DAG)
* File allocation methods (contiguous, linked, indexed)
* Free space management
* Journaling

**8️⃣ I/O Management**

* I/O devices
* Device controllers
* Interrupt handling
* DMA (Direct Memory Access)
* Disk scheduling algorithms (FCFS, SSTF, SCAN, LOOK, C-SCAN)
* Buffering & caching

**9️⃣ Security & Protection**

* Goals of security (confidentiality, integrity, availability)
* User authentication
* Access control (ACL, capability lists)
* Security models (Bell-LaPadula, Biba)
* Malware types
* Firewalls

**🔟 Advanced Topics**

* Distributed systems (concepts of distributed OS)
* Multiprocessor and multicore systems
* Real-time operating systems
* Cloud OS concepts
* Virtualization (hypervisors)
* Containers (Docker, LXC)
* Android/Linux kernel architecture

DSA TOPICS

**1️⃣ Basics of Programming**

* Arrays
* Strings
* Math & Number Theory (prime, GCD, etc.)
* Bit Manipulation
* Time and Space Complexity (Big O)

**2️⃣ Data Structures**

✅ **Linear Data Structures**

* Arrays
* Linked Lists (singly, doubly, circular)
* Stacks
* Queues (circular queue, priority queue, deque)

✅ **Non-linear Data Structures**

* Trees
  + Binary Trees
  + Binary Search Trees
  + AVL Trees
  + Segment Trees
  + Trie
* Heaps (min-heap, max-heap)
* Hashing (Hash Table, Hash Map)
* Graphs (Adjacency List, Adjacency Matrix)

**3️⃣ Algorithms**

✅ **Searching**

* Linear Search
* Binary Search
* Ternary Search

✅ **Sorting**

* Bubble Sort
* Insertion Sort
* Selection Sort
* Merge Sort
* Quick Sort
* Heap Sort
* Counting/Radix/Bucket Sort

✅ **Recursion and Backtracking**

* Recursive functions
* Backtracking problems (like N-Queens, Sudoku Solver)

✅ **Divide and Conquer**

* Merge Sort
* Quick Sort
* Closest Pair of Points

✅ **Dynamic Programming**

* 0/1 Knapsack
* Longest Increasing Subsequence
* Longest Common Subsequence
* Coin Change
* Matrix Chain Multiplication

✅ **Greedy Algorithms**

* Activity Selection
* Fractional Knapsack
* Huffman Coding
* Minimum Spanning Tree (Kruskal, Prim)

✅ **Graph Algorithms**

* BFS (Breadth First Search)
* DFS (Depth First Search)
* Dijkstra’s Algorithm
* Bellman-Ford
* Floyd Warshall
* Topological Sort
* Union Find / Disjoint Set
* Bridges, Articulation points

✅ **String Algorithms**

* Pattern Matching (KMP, Rabin Karp)
* Z-Algorithm
* Trie-based problems
* Suffix Array / Suffix Tree

✅ **Miscellaneous**

* Sliding Window
* Two Pointers
* Fast Exponentiation
* Modular Arithmetic

**4️⃣ Advanced Topics (optional, if you are preparing for advanced roles)**

* Segment Trees / Fenwick Trees
* Heavy-Light Decomposition
* Mo’s Algorithm
* Centroid Decomposition
* Suffix Automata
* Network Flow Algorithms (Ford-Fulkerson, Edmonds-Karp)