Mr cooper prep

🔥 Graphs

Common patterns:

- BFS / DFS traversal
- Shortest path (BFS, Dijkstra)
- Connected components
- Cycle detection

Practice Qs:

1️⃣ Number of Islands

2️⃣ Graph Valid Tree

3️⃣ Course Schedule (Cycle detection in Directed Graph)

4️⃣ Rotten Oranges (Shortest Path BFS)

5️⃣ Shortest Path in Binary Matrix (BFS)

🔥 **Key reminder**: Think of graphs as adjacency lists, practice with different representations!

🚀 Arrays

Patterns:

- Sliding window (fixed or variable size)
- Two pointers (start and end)
- Prefix sums
- HashMap for frequency

Practice Qs:

1️⃣ Two Sum

2️⃣ Longest Subarray with Sum K

3️⃣ Container With Most Water

4️⃣ Product of Array Except Self

5️⃣ Kadane’s Algorithm (Maximum Subarray)

📜 Strings

Patterns:

- Two pointers
- HashMap for counting chars
- Sliding window for substrings

Practice Qs:

1️⃣ Longest Substring Without Repeating Characters

2️⃣ Valid Anagram

3️⃣ Palindromic Substrings

4️⃣ Group Anagrams

5️⃣ Longest Palindromic Substring (Expand Around Center)

🧠 Dynamic Programming (DP)

Patterns:

- Memoization / Tabulation
- Subproblem recurrence
- State representation

Practice Qs:

1️⃣ Climbing Stairs (Fib pattern)

2️⃣ Longest Common Subsequence

3️⃣ Coin Change

4️⃣ House Robber

5️⃣ Palindromic Substrings (DP table)

🌳 Trees

Patterns:

- Recursive traversal (inorder, preorder, postorder)

- BFS / DFS
- Height / diameter / symmetry checks

Practice Qs:

1️⃣ Maximum Depth of Binary Tree

2️⃣ Symmetric Tree

3️⃣ Lowest Common Ancestor

4️⃣ Diameter of Binary Tree

5️⃣ Serialize and Deserialize Binary Tree

💡 1️⃣ DBMS Core Concepts (Must-Know for MCQs)

What is a DBMS?

- Software to store, manage, retrieve data.
- Examples: MySQL, Oracle, SQL Server, PostgreSQL.

3 Levels of DBMS Architecture:

1. **External level** – user views
2. **Conceptual level** – logical structure
3. **Internal level** – physical storage

🔗 Logical Independence – changes at the logical level don't affect external level.

🔗 Physical Independence – changes at storage level don't affect logical level.

Key Data Models:

- **Hierarchical:** parent-child, tree structure (like XML).
- **Network:** complex graph-like structure.
- **Relational:** tables with rows and columns (SQL).
- **Object-oriented:** data as objects.

Relational Model Basics:

- **Relation**: table

- **Tuple**: row
- **Attribute**: column
- **Degree**: number of attributes
- **Cardinality**: number of tuples

Keys:

- **Primary Key**: uniquely identifies a tuple.
- **Candidate Key**: minimal set of attributes to uniquely identify.
- **Super Key**: any set of attributes that uniquely identify.
- **Foreign Key**: links one table to another.

Integrity Constraints:

- **Entity Integrity**: primary key can’t be NULL.
- **Referential Integrity**: foreign key must match primary key of another table or be NULL.
- **Domain Constraints**: data type/format rules.

Normalization:

- Removes redundancy and ensures data integrity.

Form	Key Point
1NF	Atomic columns (no repeating groups)
2NF	No partial dependency (non-prime attributes depend on whole PK)
3NF	No transitive dependency (non-prime attributes don’t depend on other non-prime attributes)

Transactions & ACID Properties:

- **A**tomicity – all-or-nothing
- **C**onsistency – valid state
- **I**solation – concurrent transactions don’t interfere
- **D**urability – once committed, stays committed

Concurrency Control:

- **Lock-based protocols** – shared/exclusive locks
- **Timestamp ordering**
- **Deadlock handling** – wait-die, wound-wait

Indexing:

- **Primary Index** – built on primary key
- **Secondary Index** – built on non-primary key
- **Clustered Index** – data physically ordered
- **Non-clustered Index** – separate structure