

In **Facebook / Meta Hacker Cup** (and similar contests like Codeforces, Google Kick Start, Code Jam, ICPC), the problems are algorithmic and math-heavy.

Here are the **most common and important topics** used repeatedly in these contests 📖

□ 1. Data Structures

These are the foundation for most problems:

- Arrays, Stacks, Queues, Linked Lists
 - **Trees** (Binary Trees, Binary Search Trees, Segment Trees, Fenwick Trees)
 - **Graphs** (Adjacency list/matrix, BFS, DFS)
 - **Priority Queues / Heaps**
 - **Union-Find (Disjoint Set Union - DSU)**
 - Hash Maps / Sets
 - Tries (prefix trees)
-

⚙️ 2. Algorithms

Contest problems usually require clever combinations of algorithms:

- **Sorting** (Merge Sort, Quick Sort, Counting Sort)
 - **Searching** (Binary Search, Ternary Search)
 - **Divide and Conquer**
 - **Greedy Algorithms**
 - **Dynamic Programming (DP)**
 - Knapsack, LIS, LCS, DP on Trees, DP with Bitmask
 - **Graph Algorithms**
 - Dijkstra, Bellman-Ford, Floyd-Warshall
 - MST (Kruskal, Prim)
 - Topological Sort
 - Strongly Connected Components (Tarjan / Kosaraju)
 - **Shortest Path Variants** (multi-source, multi-edge weight)
 - **Flow Algorithms** (Edmonds–Karp, Dinic)
 - **Binary Lifting / LCA**
-

🔢 3. Number Theory

Very common in competitive contests:

- Modular Arithmetic
- GCD, LCM, Extended Euclid
- Modular Inverse, Fermat's Little Theorem

- Sieve of Eratosthenes
 - Euler's Totient Function
 - Combinatorics (nCr , factorial mod)
 - Modular Exponentiation
 - Chinese Remainder Theorem (CRT)
-

□ 4. Mathematics & Geometry

- Coordinate Geometry
 - Convex Hull (Graham Scan / Jarvis March)
 - Line Intersection
 - Area / Distance / Angle calculations
 - Vector cross product
 - Sweep Line algorithms
-

□ 5. Strings

- Pattern Matching (KMP, Z, Rabin-Karp)
 - Prefix Function
 - Suffix Array / Suffix Automaton
 - Rolling Hash
 - Palindrome algorithms (Manacher's)
 - Trie / Aho-Corasick automaton
-

🔍 6. Advanced Topics (for harder rounds)

- DP Optimization (Convex Hull Trick, Monotonic Queue, Divide and Conquer DP)
 - Matrix Exponentiation
 - Bitmask DP
 - Meet-in-the-middle
 - Binary Indexed Trees (Fenwick Tree)
 - Segment Trees with Lazy Propagation
 - Persistent Data Structures
 - Heavy-Light Decomposition
 - Centroid Decomposition
 - Probability / Expected Value problems
-

□ 7. Problem Patterns

You'll often see:

- Simulation problems (implement logic carefully)
 - Constructive problems (build required output step-by-step)
 - Optimization problems (min/max cost, time, or distance)
 - Counting problems (combinatorial or DP-based)
 - Path-finding (in grids or graphs)
 - String manipulation (substring, prefix/suffix, palindromes)
-

🏆 Tip for You (as a Java & C++ coder)

Since you're comfortable with both:

- Use **C++ STL** for speed (maps, sets, priority_queue, vector)
- Use **Java** for cleaner OOP-style problems (DP, recursion-heavy)
- Practice on **past Meta Hacker Cup** and **Codeforces Div2 A–D** problems.