16 Algorithmic Tricks

to solve a

Coding Question

in an

Interview

DesignGurus.org

1. **Sliding Window**

Use sliding window technique for problems that involve handling the input data in a specific window size.

2. HashTable

Use HashTable to improve the time complexity of lookups and other operations.

3. Two Pointers

Use the two pointers technique for problems that involve pointers traversing an array. The pointers can move in opposite or the same directions and sometime they move with different speed like the case with Fast and Slow pointer approach.

4. Backtracking

Use backtracking or brute force to generate all possible solutions and then filter them based on constraints.

5. Greedy Algorithms

Use greedy algorithms for problems with an optimal substructure property.

6. Divide and Conquer

Use divide and conquer for problems that can be broken down into subproblems.

7. Dynamic Programming

Use dynamic programming or memoization to optimize recursive solutions.

8. Recursion

Use recursion to simplify the problem and make it more elegant.

9. Binary Search

Use binary search for problems that involve searching for a specific value in a sorted array or list.

10. BFS and DFS

Use breadth-first search (BFS) or depth-first search (DFS) for problems that involve traversing a tree or a graph.

11. Meet-in-the-Middle

Use the Meet-in-the-middle technique to solve problems that can be broken down into two subproblems that can be solved independently.

12. Recursion Tree

Use the recursion tree method to visualize and analyze the time complexity of recursive solutions.

13. Union-Find

Use the Union-Find algorithm for solving problems that involve disjoint sets, such as detecting cycle in a graph, Kruskal's minimum spanning tree algorithm.

14. Trie

Use the Trie data structure for problems that involve searching for strings, such as auto-complete, spell-checking, IP routing.

15. Binary Search Tree

Use the Binary search tree for storing data, where you can perform operation like search, insert, delete, in O(log n) time.

16. Segment Tree

Use the Segment Tree/Fenwick Tree for Range Queries and Updates in an array/list.

➡ Follow these techniques to distinguish yourself from others!

➡ All these patterns are discussed in "Grokking the Coding Interview" and "Grokking Dynamic Programming" from **DesignGurus.org**

➡ New year sale: **20% off** on all courses.