**Suggested Topics for Presentation or Written Exposition**

Analyze these in detail, presenting or writing them up so that others can really understand in depth. Go beyond what is provided in the text.

3.1 Selection Sort and Bubble Sort: Consider when one would want to use these

3.3 Closest Pair and Convex Hull Problems by brute force

3.4 (Exhaustive Search)

3.5 Depth-First Search and Breadth-First Search

4.1 (Insertion Sort)

4.2 Topological Sorting: Demonstrate the algorithms on more extensive examples

4.3 Johnson-Trotter Algorithm for generating permutations: Explain and demonstrate with a bigger example why the algorithm works

4.4 (Russian Peasant Multiplication)

4.4 Josephus Problem: Thoroughly explain how the 1-bit cyclic shift solves the problem

4.5 Lomuto Partitioning with Quickselect: Explain how it works and demonstrate Quickselect

4.5 Game of Nim: Discuss solutions to variants of Nim, Explain convincingly how the binary sum technique works

5.1 (Mergesort)

5.2 Quicksort: Compare to Quicksort in detail – when would each be used?

5.4 Multiplication of large numbers: Thoroughly explain the time complexity

5.4 Strassen's Matrix Multiplication: Thoroughly explain the time complexity

5.5 Closest-Pair Problem: Thoroughly present the algorithm and explain the time complexity in detail

5.5 Quickhull for Convex-Hull Problem: Demonstrate the algorithm on a complete example and show precisely how the points can be determined to be on the right or left of the line segment

6.2 (Gaussian Elimination)

6.2 LU Decomposition: Explain in detail how LU Decomposition works

6.3 AVL Trees: Discuss how it works and demonstrate the algorithm on more extensive examples

6.3 2-3 Trees: Discuss how it works and demonstrate the algorithm on more extensive examples

6.4 Heaps with Heapsort: Present in detail the time complexity analysis of the main methods of working with heaps

6.5 Horner's Rule to evaluate polynomials: Present Synthetic Division in detail with examples

6.5 (Binary exponentiation Left-Right and Right-Left)

7.1 (Comparison Counting Sort)

7.1 (Distribution Counting Sort)

7.2 (Horspool's Algorithm for string searching)

7.2 Boyer-Moore Algorithm for string searching: Thoroughly compare/contrast and demonstrate Horspool’s and Boyer-Moore

7.3 (Hashing closed and open)

7.4 B-Trees: Demonstrate the algorithm with a more complete example

8.1 (Basic Dynamic Programming examples)

8.2 (Knapsack Problem with Memory Functions (memoization))

8.3 Optimal Binary Search Tree construction: Explain in detail and a good example how this works

8.4 Warshal's Algorithm for transistive closure of directed graph: Demonstrate in detail how this works

8.4 Floyd's Algorithm for All-Pairs Shortest-Paths Problem: Demonstrate in detail how this works

9.1 Prim's Algorithm for minimum spanning tree: Demonstrate in detail how this works

9.2 Kruskal's Algorithm for minimum spanning tree: Demonstrate in detail how this works

9.2 Disjoint Subsets and Union-Find Algorithms: Demonstrate in detail how this works

9.3 Dijkstra's Algorithm for single-source shortest-paths problem: Demonstrate in detail how this works

9.4 Huffman's Algorithm for a Huffman Code: Demonstrate in detail how this works

10.1 Simplex Method for solving systems of linear inequalities: Demonstrate in detail how this works

10.2 Ford-Fulkerson method for maximum-flow problem: Demonstrate in detail how this works

10.3 Maximum matching in bipartite graphs: Demonstrate in detail how this works

10.4 Stable Marriage Problem: Demonstrate in detail how this works

11.1 Lower Bounds analysis

11.2 Decision Trees: Such as for the egg dropping problem

11.3 P, NP, and NP Completeness

11.4 Numerical Problems: Applications in Numerical Analysis

12.1 n-Queens Problem with Backtracking

12.1 Hamiltonian Circuit and Subset-Sum Problems with Backtracking

12.2 Assignment Problem with Branch-and-Bound

12.2 Knapsack Problem with Branch-and-Bound

12.2 Traveling Salesman Problem with Branch-and-Bound

12.3 Traveling Salesman Problem - Approximation Algorithms

12.3 Knapsack Problem - Approximation Algorithms