

- Sorting Algorithms (Bubble sort, Insertion Sort, Counting Sort, Merge Sort, Radix Sort):
 - How they work
 - Demonstration for a given array of numbers
 - Best case & Worst case
 - Running time & Memory requirement
 - Comparison (with reasoning) which one is better
- Computational Complexity/Asymptotic Notation
 - Statement-by-Statement Analysis of Code Segment/Function
 - Finding the form of the function for the running-time complexity
 - Finding the most appropriate notation if the expression for the function is given
 - Best case/lower bound, Worst case/upper bound and Tight bound Analysis
- Recurrence Relations
 - Solving given recurrence relations using backward substitution, recursion tree or master theorem
 - Finding the recurrence relation from a given code segment and then solve
- Partitioning, Quickselect, Quicksort
 - Demonstration for a given array
 - How the recursive calls are made on a given array
 - Running-time
 - Best/worst case
 - Effect of partitioning on Quicksort

- Binary Tree, BST & 2-3 tree
 - Construction
 - Insert, delete, traversal, search operation (and their complexity)
 - How to perform those operations for given elements
 - For BST: Recursive and iterative version of search operation
 - Tree-Sort and its running-time complexity

** Sample Practice Problems: Relevant example and exercise problems from the book chapter referenced at the end of each lecture and relevant Assignment Problems