Lecture 1:

1. Insertion Sort Algorithm (code, example, dry run)
2. Insertion Sort time complexity analysis
3. Best case, worst case analysis

Lecture 2:

1. Asymptotic Notations
2. Big-Oh, Big-Omega, Big-theta

Lecture 3:

1. Divide and Conquer
2. Merge Sort Algorithm (example, code, dry run to understand recursive calls)
3. Merge Sort time complexity analysis
4. Solving Recursions (Tree Method)

Lectures 4 &5:

1. Merge Sort space complexity analysis
2. Insertion Sort space complexity analysis (In-place algorithm)
3. Loop invariants (proof of correctness):
    a. Sum of array
    b. Insertion Sort
    c. Merge Sort
4. Solving Recursions:
    a. Tree method
    b. Back Substitution/Iterative method

Lecture 6:

1. More Recursions Practice:
    a. Tree method
    b. Back Substitution/Iterative method
2. Maximum subarray sum problem:
    a. Brute force algo 1 — O (n^3)
    b. Brute force algo 2 — O (n^2)
    c. Divide and conquer approach (crossing sub-array case)

Lecture 7:

1. Maximum subarray sum problem:
    a. Complete Divide and Conquer solution
    b. Complexity analysis
2. Quick Sort
    a. Algorithm (using Divide and conquer)
    b. Space Complexity Analysis (in-place algorithm)
    c. Time Complexity Analysis

d. Best Case
e. Worst Case
f. Average case
g. Tight bounds of average case of Quick Sort
h. Considering first element as pivot
i. Loop Invariant

Lecture 8:

1. Quiz 1
2. Solving Recursions:
    a. Difference between Substitution and Back Substitution (Iterative) Method
    b. Substitution Method (with Mathematical Induction)
    c. Master Theorem
    (LEARN TO SOLVE EQUATIONS!!!)

Lecture 9:

1. Extended Master Theorem
2. Counting Inversions:
    a. Brute Force Solution
    b. Divide and Conquer Solution