***Module 1:***

**Introduction:**

What is an Algorithm? (T2:1.1), Algorithm Specification (T2:1.2), Analysis Framework (T1:2.1), Performance Analysis: Space complexity, Time complexity (T2:1.3).

**Asymptotic Notations:**

Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ), and Little-oh notation (o), Mathematical analysis of Non-Recursive and recursive Algorithms with Examples (T1:2.2, 2.3, 2.4). Important Problem Types: Sorting, Searching, String processing, Graph Problems, Combinatorial Problems.

**Fundamental Data Structures:**

Stacks, Queues, Graphs, Trees, Sets and Dictionaries. (T1:1.3,1.4).

Programming Examples:

1. <https://github.com/kmgowda/ds-programs-python/blob/master/src/add.py>
2. <https://github.com/kmgowda/ds-programs-python/blob/master/src/subtract.py>
3. <https://github.com/kmgowda/ds-programs-python/blob/master/src/bit-next.py>
4. <https://github.com/kmgowda/ds-programs-python/blob/master/src/bitflips.py>
5. <https://github.com/kmgowda/ds-programs-python/blob/master/src/multiply-rec.py>
6. <https://github.com/kmgowda/ds-programs-python/blob/master/src/kth-multiple.py>
7. <https://github.com/kmgowda/ds-programs-python/blob/master/src/flipbitwin.py>
8. <https://github.com/kmgowda/ds-programs-python/blob/master/src/min-difference.py>
9. <https://github.com/kmgowda/ds-programs-python/blob/master/src/missing-number.py>
10. <https://github.com/kmgowda/ds-programs-python/blob/master/src/missing-two-numbers.py>
11. <https://github.com/kmgowda/ds-programs-python/blob/master/src/rand7.py>
12. <https://github.com/kmgowda/ds-programs-python/blob/master/src/directed-graph.py>

***Module 2:***

**Divide and Conquer:**

General method, Binary search, Recurrence equation for divide and conquer, Finding the maximum and minimum (T2:3.1, 3.3, 3.4), Merge sort, Quick sort (T1:4.1, 4.2), Strassen’s matrix multiplication (T2:3.8), Advantages and Disadvantages of divide and conquer.

**Decrease and Conquer Approach:**

Topological Sort. (T1:5.3).

Programming Examples:

1. <https://github.com/kmgowda/ds-programs-python/blob/master/src/max-number.py>
2. <https://github.com/kmgowda/ds-programs-python/blob/master/src/max-square.py>
3. <https://github.com/kmgowda/ds-programs-python/blob/master/src/max-matrix.py>
4. <https://github.com/kmgowda/ds-programs-python/blob/master/src/quick-sort.py>
5. <https://github.com/kmgowda/ds-programs-python/blob/master/src/merge-sort.py>
6. <https://github.com/kmgowda/ds-programs-python/blob/master/src/smallest-k.py>

***Module 3:***

**Greedy Method:**

General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines (T2:4.1, 4.3, 4.5).

**Minimum cost spanning trees:**

Prim’s Algorithm, Kruskal’s Algorithm (T1:9.1, 9.2).

**Single source shortest paths:**

Dijkstra's Algorithm (T1:9.3).

**Optimal Tree problem:**

Huffman Trees and Codes (T1:9.4).

**Transform and Conquer Approach:**

Heaps and Heap Sort (T1:6.4).

Programming Examples:

1. <https://github.com/kmgowda/ds-programs-python/blob/master/src/heap-sort-1.py>
2. <https://github.com/kmgowda/ds-programs-python/blob/master/src/heap-sort-2.py>
3. <https://github.com/kmgowda/ds-programs-python/blob/master/src/heap-sort.py>
4. <https://github.com/kmgowda/ds-programs-python/blob/master/src/dijkstras-algorithm.py>
5. <https://github.com/kmgowda/ds-programs-python/blob/master/src/kruskal-algorithm.py>
6. <https://github.com/kmgowda/ds-programs-python/blob/master/src/prims-algorithm.py>
7. <https://github.com/kmgowda/ds-programs-python/blob/master/src/all-paths-graph.py>
8. <https://github.com/kmgowda/ds-programs-python/blob/master/src/bfs-dfs-graph.py>
9. <https://github.com/kmgowda/ds-programs-python/blob/master/src/weighted-directed-graph.py>
10. <https://github.com/kmgowda/ds-programs-python/blob/master/src/shortest-path-unweighted-graph.py>
11. <https://github.com/kmgowda/ds-programs-python/blob/master/src/work-order-graph.py>
12. <https://github.com/kmgowda/ds-programs-python/blob/master/src/living-people.py>

***Module 4:***

**Dynamic Programming:**

General method with Examples, Multistage Graphs (T2:5.1, 5.2).

**Transitive Closure:**

Warshall’s Algorithm,

**All Pairs Shortest Paths:**

Floyd's Algorithm, Optimal Binary Search Trees, Knapsack problem ((T1:8.2, 8.3, 8.4), Bellman-Ford Algorithm (T2:5.4), Travelling Sales Person problem (T2:5.9), Reliability design (T2:5.8).

Programming Examples:

1. <https://github.com/kmgowda/ds-programs-python/blob/master/src/dinominations.py>
2. <https://github.com/kmgowda/ds-programs-python/blob/master/src/count-steps.py>
3. <https://github.com/kmgowda/ds-programs-python/blob/master/src/cellphone-wordset.py>
4. <https://github.com/kmgowda/ds-programs-python/blob/master/src/count-2s.py>
5. <https://github.com/kmgowda/ds-programs-python/blob/master/src/bst-subarray.py>
6. <https://github.com/kmgowda/ds-programs-python/blob/master/src/short-sequence.py>

***Module 5:***

**Backtracking:**

General method (T2:7.1), N-Queens problem (T1:12.1), Sum of subsets problem (T1:12.1), Graph coloring (T2:7.4), Hamiltonian cycles (T2:7.5).

**Programme and Bound:**

Assignment Problem, Travelling Sales Person problem (T1:12.2),

**0/1 Knapsack problem (T2:8.2, T1:12.2):**

LC Programme and Bound solution (T2:8.2), FIFO Programme and Bound solution (T2:8.2).

**NP-Complete and NP-Hard problems:**

Basic concepts, non deterministic algorithms, P, NP, NP-Complete, and NP-Hard classes (T2:11.1). RBT: L1, L2, L3

Programming Examples:

1. <https://github.com/kmgowda/ds-programs-python/blob/master/src/n-queens-rec.py>
2. <https://github.com/kmgowda/ds-programs-python/blob/master/src/n-queens.py>
3. <https://github.com/kmgowda/ds-programs-python/blob/master/src/random_node.py>
4. <https://github.com/kmgowda/ds-programs-python/blob/master/src/paranthesis.py>
5. <https://github.com/kmgowda/ds-programs-python/blob/master/src/tree_path_sum.py>
6. <https://github.com/kmgowda/ds-programs-python/blob/master/src/sum-sequence.py>
7. <https://github.com/kmgowda/ds-programs-python/blob/master/src/2d-word-search.py>
8. <https://github.com/kmgowda/ds-programs-python/blob/master/src/cellphone-wordset.py>
9. <https://github.com/kmgowda/ds-programs-python/blob/master/src/masseuse.py>
10. <https://github.com/kmgowda/ds-programs-python/blob/master/src/robo-grid.py>
11. <https://github.com/kmgowda/ds-programs-python/blob/master/src/tree_path_sum.py>
12. <https://github.com/kmgowda/ds-programs-python/blob/master/src/pond-sizes.py>

**Textbooks:**

1. Introduction to the Design and Analysis of Algorithms, Anany Levitin:, 2rd Edition, 2009. Pearson.

2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press

**Reference Books:**

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI.

2. Design and Analysis of Algorithms , S. Sridhar, Oxford (Higher Education).

***Programming Examples Reference Book:***

1. ***Cracking The Coding interview by Gayle Lakman*** 
   1. [***https://github.com/chaudharyachint08/Self-Learning/blob/master/BOOKS/DSA%20%26%20Competitive%20Programming/Cracking-the-Coding-Interview-6th-Edition-189-Programming-Questions-and-Solutions.pdf***](https://github.com/chaudharyachint08/Self-Learning/blob/master/BOOKS/DSA%20%26%20Competitive%20Programming/Cracking-the-Coding-Interview-6th-Edition-189-Programming-Questions-and-Solutions.pdf)
   2. [***https://github.com/chaudharyachint08/Self-Learning/blob/master/BOOKS/DSA%20%26%20Competitive%20Programming/Cracking-the-Coding-Interview-6th-Edition-189-Programming-Questions-and-Solutions.pdf***](https://github.com/chaudharyachint08/Self-Learning/blob/master/BOOKS/DSA%20%26%20Competitive%20Programming/Cracking-the-Coding-Interview-6th-Edition-189-Programming-Questions-and-Solutions.pdf)