

## CS-9: Design and Analysis of Algorithms

**Iterative Techniques:** 8L  
Sequential search, insertion sort, bubble sort, selection sort, merging two sorted arrays, correctness and analysis.

[1]: [chap 3: 3.5 up to 3.5.3]:

[2]: [chap 2: Sec. 2.1 up to p20, p24-p28, p31-p33]:

**Divide and Conquer:** 8L  
Binary search, Merge sort, quick sort, Heap sort, correctness and analysis.

[1]: [chap 3: 3.5.7]:

[2]: [chap 2 :p34-p37, chap 6: 6.1-6.4, chap 7: 7.1-7.4]:

**Dynamic Programming and Greedy algorithms:** 8L  
Fractional knapsack problem, 0-1 knapsack problem, activity selection problem, rod cutting, longest common subsequence.

[2]: [chap 15: 15.1, 15.4, ch16: 16.1, 16.2]:

**Linear time Sorting and Searching techniques:** 6L  
Count sort, radix sort, bucket sort, finding minimum and maximum, order statistic in expected linear time, their analysis.

[2]: [chap 8: 8.2-8.4, ch9: 9.1,9.2]:

**String Matching:** 4L  
Brute force method, KMP Algorithm: computing fail links, analysis.

[1]: [chap 11: 11.1-11.3]:

**Height Balanced Trees:** 4L  
Red Black Trees: rotation, insertion, deletion, analysis of operations

[2]: [chap 13: 13.1-13.4]:

## UNDERGRADUATE PROGRAMME IN COMPUTER SCIENCE

### **Graph Traversal Algorithms:** 6L

Representation of graphs, breadth first search, depth first search, analysis of traversal methods, applications.

[2]: [chap 22: 22.1-22.5]:

### **Minimum Spanning Tree and shortest path:** 4L

Prim's algorithms, Kruskal's algorithm, Dijkstra's algorithm, analysis of these algorithms.

[2]: [chap 23: 23.1,23.2, ch24: 24.3]:

## **Recommended Reading Material**

### **Text Books**

1. Sara Baase, *Computer Algorithms: Introduction to design and analysis*, Third Edition, Pearson, 2002.
2. T.H.Cormen, C.E. Leiserson, R.L.Rivest, C.Stein, *Introduction to Algorithms*, Third Edition, Prentice Hall of India, 2010.

### **Reference Books**

3. J.Kleinberg and E.Tardos, *Algorithm Design*, First Edition, Pearson, 2012.
4. S.Dasgupta, C.Papadimitriou, U.Vazirani, *Algorithms*, First Edition, Tata McGraw Hill, 2013.

### **Online Reading/Supporting Material**

5. <http://nptel.iitm.ac.in/video.php?subjectId=106102064>

## **LIST OF PRACTICALS BASED ON CS -9: DESIGN AND ANALYSIS OF ALGORITHMS**

<b>S. No.</b>	<b>Practical Title</b>
1.	<ol style="list-style-type: none"><li>i. Implement Insertion Sort (The program should report the number of comparisons)</li><li>ii. Implement Merge Sort(The program should report the number of comparisons)</li></ol>

## UNDERGRADUATE PROGRAMME IN COMPUTER SCIENCE

2.	Implement Heap Sort(The program should report the number of comparisons)
3.	Implement Randomized Quick sort (The program should report the number of comparisons)
4.	Implement Radix Sort
5.	Create a Red-Black Tree and perform following operations on it: i. Insert a node ii. Delete a node iii. Search for a number & also report the color of the node containing this number.
6.	Write a program to determine the LCS of two given sequences
7.	Implement Breadth-First Search in a graph
8.	Implement Depth-First Search in a graph
9.	Write a program to determine the minimum spanning tree of a graph

For the algorithms at S.No 1 to 3 test run the algorithm on 100 different inputs of sizes varying from 30 to 1000. Count the number of comparisons and draw the graph. Compare it with a graph of  $n \log n$ .

- For the algorithms at S.No 4 test run the algorithm on 100 different inputs of sizes varying from 30 to 1000. Count the number of operations and draw the graph.
- For plotting the above mentioned graphs, on x axis plot different values of  $n$ (input size) and on y axis plot corresponding number of comparisons.