Design and Analysis of Algorithms (BHCS08)

Unit	Topic	Reference	Total Hours
1	Algorithm Design Techniques: Iterative technique: Applications to Sorting and Searching (review), their correctness and analysis. Divide and Conquer: Application to Sorting and Searching (review of binary search), merge sort, quick sort, their correctness and analysis. Dynamic Programming: Application to various problems, their correctness and analysis. Greedy Algorithms: Application to various problems, their correctness and analysis.	Ch 2 [1] Ch 7 [1] 6.1 , 6.2, 6.4 [2] 4.1, 4.2, 4.4, 4.5(excluding reverse delete algorithm), 4.6 [2]	12 8 12
2	More on Sorting and Searching: Heapsort, Lower Bounds using decision trees, sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Medians & Order Statistics, complexity analysis and their correctness.	Ch 6 [1] Ch 8 [1] 9.1, 9.2, 9.3 [1]	12
3	Advanced Analysis Technique: Amortized analysis	17.1, 17.2, 17.3 [1]	4
4	Graphs: Graph Algorithms - Breadth First Search, Depth First Search and its Applications.	Ch 3 [2]	12

References

1. Cormen, T.H., Leiserson, C.E. Rivest, R.L., & Stein, C.(2015). Introduction to Algorithms. 3rd edition. PHI.

2. Kleinberg, J., & Tardos, E. (2013). Algorithm Design. 1st edition. Pearson Education India. Additional Resources

Practical List

1. i. Implement Insertion Sort (The program should report the number of comparisons)
   ii. Implement Merge Sort (The program should report the number of comparisons)
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. Implement Bucket Sort
6. Implement Randomized Select
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 using both Prims and Kruskals algorithm
10. Write a program to solve the weighted interval scheduling problem
11. Write a program to solve the 0-1 knapsack problem

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 nlogn.