**Tentative Guidelines for B.Sc. (Hons.) Computer Science V Sem**

**Algorithms and Advanced Data Structures**

| Unit Name | Syllabus | Guidelines | Suggested No. of Lectures |
|---|---|---|---|
| **Unit 1:** List and Iterator ADTs | Vectors, Lists, Sequences | 6.1, 6.2, 6.3 [1] | 4 |
| **Unit 2:** Hash Tables, Dictionaries | Hash Functions, Collision resolution schemes | 10.1 – 10.3 [4] | 6 |
| **Unit 3:** Strings | String Matching: Brute Force, KMP algorithm;<br>Tries: Standard Tries, Compressed Tries, Suffix Tries, Search Engines | 12.3.1, 12.3.3, 12.5 [1] | 8 |
| **Unit 4:** More on Trees | 2-4 Trees, B Trees | 7.1.8 [4]<br>7.1.1 [4] | 8 |
| **Unit 5:** More on Graphs | Bellman Ford Algorithm, Union Find Data Structures - application Kruskal's algorithm | 6.8 [3]<br>4.6 [3] | 8 |
| **Unit 6:** Randomization | Randomized Quicksort, Randomized Select, Skip lists | 7.3, 9.2 [2]<br>9.4, 9.4.1 [1] | 6 |
| **Unit 7:** Network Flows | Ford Fulkerson algorithm for the max flow problem | 7.1 [3] | 5 |

**Essential/Recommended readings**


1. Goodrich, M.T, Tamassia, R., & Mount, D. Data Structures and Algorithms Analysis in C++, 2nd edition, Wiley, 2011 (Note: An e-copy of this book can be procured from the publisher for the college library).
2. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, 4th edition, Prentice Hall of India, 2022.
3. Kleinberg, J., Tardos, E. Algorithm Design, 1st edition, Pearson, 2013.
4. Drozdek, A. Data Structures and Algorithms in C++, 4th edition, Cengage Learning. 2012.

## Practical List (30 Hours)

1. Write a program to sort the elements of an array using Randomized Quick Sort (the program should report the number of comparisons).
2. Write a program to find the $i^{th}$ smallest element of an array using Randomized Select.
3. Write a program to determine the minimum spanning tree of a graph using Kruskal's algorithm.
4. Write a program to implement the Bellman-Ford algorithm to find the shortest paths from a given source node to all other nodes in a graph.
5. Write a program to implement a B-Tree.
6. Write a program to implement the Tree Data structure, which supports the following operations:
    a. Insert
    b. Search
7. Write a program to search a pattern in a given text using the KMP algorithm.
8. Write a program to implement a Suffix tree.