

DSC13: ALGORITHMS AND ADVANCED DATA STRUCTURES

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course
		Lecture	Tutorial	Practical/ Practice		
DSC 13 Algorithms and Advanced Data Structures	4	3	0	1	Pass in Class XII	DSC 07 Data Structures with C++, DSC 10 Design and Analysis of Algorithms

Learning Objectives

This course is designed to build upon the fundamentals of data structures and algorithm design and gain exposure to advanced data structures and algorithms for new problems.

Learning Outcomes

On successful completion of this course, the student will be able to:

1. Comprehend and use data structures and algorithms for string matching.
2. Comprehend and use disc-based data structures.
3. Implement and analyze advanced data structures and algorithms for graphs.
4. Appreciate the strength of randomization in data structures and algorithms.
5. Understand the principles of network flow problems.

Unit 1

(10 hours)

Advanced Analysis of Algorithms and Intractability: Amortized Analysis, Concept of polynomial time computation, polynomial time reductions, decision vs optimization problems, introduction to NP, NP-hard and NP-Complete classes.

Unit 2

(13 hours)

Algorithms on Strings and Flows: String Matching: KMP algorithm; Tries: Standard Tries, Compressed Tries, Suffix Tries, Search Engines, and Ford Fulkerson algorithm for max flow problem.

Unit 3

(16 hours)

More on Trees and Graphs: 2-4 Trees, B Trees, Bellman Ford Algorithm, Union Find Data

Structures - application Kruskal's algorithm

Unit 4

(6 hours)

Randomization: Randomized Quicksort, Randomized Select, Skip List

References

1. Goodrich, M.T, Tamassia, R., & Mount, D.,. Data Structures and Algorithms Analysis in C++, 2nd edition, Wiley, 2011.
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.

Suggested Practical List

1. Write a program to sort the elements of an array using Randomised 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 Randomised 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 Trie Data structure, which supports the following operations:
 - I. Insert
 - II. 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.