

UNIVERSITY OF DELHI

CNC-II/093/1/Misc./2025/16

Dated: 21.10.2025

NOTIFICATION

Sub: Amendment to Ordinance V

Following addition be made to Appendix-II-A to the Ordinance V (2-A) of the Ordinances of the University;

Add the following:

The following revision/ modifications in DSC-10: Design and Analysis of Algorithm and DSC 13: Algorithms and Advanced Data Structures of BSc (H) Computer Science are notified herewith for the information of all concerned as under:

Modifications in DSC-10: Design and Analysis of Algorithm

Addition	Deletion
1. Unit 6: Hash Functions, Collision resolution schemes (8 hours) 2. Unit 5: Dynamic Programming (hours changed from 6 to 7, resulting in addition of 1 hour) Total : 9 hours added	1. Unit 6: Intractability (6 Hours) 2. Unit 7: Advanced Analysis of Algorithms: Amortized Analysis (3 Hours) Total : 9 Hours removed.

Modifications in DSC-13: Algorithms and Advanced Data Structures

Addition	Deletion
1. Unit 1: Advanced Analysis of Algorithms: Amortized Analysis (3 Hours) 2. Unit 7: Intractability (7 Hours) Total : 10 Hours added	1. Unit 1: List and Iterator ADTs: Vectors, Lists, Sequences (4 Hours) 2. Unit 2: Hash Tables, Dictionaries: Hash Functions, Collision resolution schemes (6 Hours) Total : 10 Hours removed

The revised syllabi are enclosed at Annexure-1.

h.w. 21/10/25

REGISTRAR

Annexure-1

Course Code/ Course Title	List of changes incorporated
DSC 10: Design and Analysis of Algorithms	Merged: Unit 3 has been merged with Unit 1, and the combined unit will be completed in 16 hours. Unit 4 has been merged with Unit 2, with the addition of two new topics: Shortest Path Problem and Interval Scheduling. The combined unit will be completed in 16 hours. Renamed: Unit 5 has been renamed as Unit 3, with one new topic added: Weighted Interval Scheduling. The unit will be completed in 9 hours. Unit 6 has been renamed as Unit 4 and will be completed in 4 hours.
DSC 13: Algorithms and Advanced Data Structures	Added: 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. Merged: Unit 7 is merged with Unit 3, and the combined unit (Unit 2) will be completed in 13 hours. Unit 5 is merged with Unit 4, and the combined unit (Unit 3) will be completed in 16 hours. Removed(10 hours) Unit 1: List and Iterator ADTs: Vectors, Lists, Sequences Unit 2: Hash Tables, Hash Functions and Collision Resolution Schemes has now been removed and moved to DSC 10. Renamed: Unit 6 has been renamed Unit 4 and will be completed in 6 hours.

DSC10/DSC07/GE7a: DESIGN AND ANALYSIS OF ALGORITHMS

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Design and Analysis of Algorithms	4	3	0	1	Pass in Class XII	Data Structures

Course Objectives

The course is designed to develop an understanding of various algorithm design techniques and apply them to problem-solving. The course shall also enable the students to verify the correctness of algorithms and analyze their time complexity.

Learning Outcomes

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

- Analyze and compare the asymptotic time and space complexity of algorithms to assess their efficiency.
- Design and implement algorithms using fundamental techniques such as divide and conquer, greedy, and dynamic programming.
- Apply and evaluate standard algorithms for searching, sorting, graph processing, and optimization problems.
- Demonstrate understanding of algorithm correctness and justify design choices through theoretical analysis.
- Construct and analyze hashing-based data structures using appropriate hash functions and collision resolution schemes.

Syllabus

Unit 1

(16 hours)

Linear Search, Binary Search, Insertion Sort, Selection Sort, Bubble Sort, Heapsort, Linear Time Sorting, running time analysis and correctness, Introduction to divide and conquer technique, Merge Sort, Quick Sort, Randomized quicksort, Maximum-subarray problem, Strassen's algorithm for matrix multiplication.

Unit 2

(16 hours)

Review of graph traversals, graph connectivity, testing bipartiteness, Directed Acyclic Graphs and Topological Ordering, Introduction to the Greedy algorithm design approach, Minimum Spanning Tree, Shortest Path Problem, Interval Scheduling, fractional knapsack problem, and analysis of time

complexity.

Unit 3

(9 hours)

Introduction to the Dynamic Programming approach, Weighted Interval Scheduling, Integer Knapsack problem, application to the subset sum problem and analysis of time complexity.

Unit 4

(4 hours)

Hash Tables, Hash Functions, Collision Resolution Schemes.

Essential/recommended readings

1. Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. Introduction to Algorithms, 4th edition, Prentice Hall of India, 2022.
2. Kleinberg, J., Tardos, E. Algorithm Design, 1st edition, Pearson, 2013.

Additional references

1. Basse, S., Gelder, A. V., Computer Algorithms: Introduction to Design and Analysis, 3rd edition, Pearson, 1999.

Practical List (If any): (30 Hours)

1. Write a program to sort the elements of an array using Insertion Sort (The program should report the number of comparisons).
2. Write a program to sort the elements of an array using Merge Sort (The program should report the number of comparisons).
3. Write a program to sort the elements of an array using Heap Sort (The program should report the number of comparisons).
4. Write a program to multiply two matrices using Strassen's algorithm for matrix multiplication
5. Write a program to sort the elements of an array using Radix Sort.
6. Write a program to sort the elements of an array using Bucket Sort.
7. Display the data stored in a given graph using the Breadth-First Search algorithm.
8. Display the data stored in a given graph using the Depth-First Search algorithm.
9. Write a program to determine a minimum spanning tree of a graph using Prim's algorithm.
10. Write a program to implement Dijkstra's algorithm to find the shortest paths from a given source node to all other nodes in a graph.
11. Write a program to solve the weighted interval scheduling problem.
12. Write a program to solve the 0-1 knapsack problem.

For the algorithms at S.No. 1, 2, and 3, test the algorithm on 100 different input sizes, varying from 30 to 1000. For each size, find the number of comparisons averaged on 10 different input instances; plot a graph for the average number of comparisons against each input size. Compare it with a graph of $n \log n$.

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.