Perform the following activities, record and report in standard form.

(NOTE: Exercise extra caution while performing these exercises and codes)

1. Perform various Virtual Machine based exercises on https://vulnhub.com/
2. Perform Capture the Flag (CTF) exercises from https://www.hacker101.com/
3. Follow the lessons and activities from https://www.hackingloops.com/ethical-hacking/
4. Google site for hacking https://google-gruyere.appspot.com/
5. OWASP WebGoat https://github.com/WebGoat/WebGoat

GENERIC ELECTIVES (GE-7f): 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
		Lecture	Tutorial	Practical/ Practice		
GE 7f: Design and Analysis of Algorithms	**4**	**3**	**0**	**1**	Pass in Class XII	Data Structures

Course Objectives

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

Learning Outcomes

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

- Compute and compare the asymptotic time complexity of algorithms.
- Use appropriate algorithm design technique(s) for solving a given problem.

Syllabus

Unit 1 **(8 hours)**

Searching, Sorting, Selection: Linear Search, Binary Search, Insertion Sort, Selection Sort, Bubble Sort, Heapsort, Linear Time Sorting, running time analysis and correctness.

Unit 2 **(5 hours)**

Graphs: Review of graph traversals, graph connectivity, testing bipartiteness, Directed Acyclic Graphs and Topological Ordering, Minimum Spanning Trees.

Unit 3 **(8 hours)**

Divide and Conquer: Introduction to divide and conquer technique, Merge Sort, Quick Sort, Randomised quicksort, Maximum-subarray problem, Strassen's algorithm for matrix multiplication.

Unit 4 **(5 hours)**

Greedy algorithms: Introduction to the Greedy algorithm design approach, application to minimum spanning trees, fractional knapsack problem, and analysis of time complexity.

Unit 5 **(5 hours)**

Dynamic Programming: Introduction to the Dynamic Programming approach, application to subset sum, integer knapsack problems, and analysis of time complexity.

Unit 6 **(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 the 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 the 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.

(Computer Science Courses for Undergraduate Programme of study with Computer Science discipline as one of the three Core Disciplines)

(For e.g. courses for B.Sc. Programme with Computer Science as discipline)

DISCIPLINE SPECIFIC CORE COURSE (DSC07): 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		
DSC07: Design and Analysis of Algorithms	**4**	**3**	**0**	**1**	Pass in Class XII	Data Structures

Course Objective

The course is designed to develop an understanding of different algorithm design techniques and use them for 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:

- Compute and compare the asymptotic time complexity of algorithms.
- Use appropriate algorithm design technique(s) for solving a given problem.

Syllabus

Unit 1 **(8 hours)**

Searching, Sorting, Selection: Linear Search, Binary Search, Insertion Sort, Selection Sort, Bubble Sort, Heapsort, Linear Time Sorting, running time analysis and correctness.

Unit 2 **(5 hours)**

Graphs: Review of graph traversals, graph connectivity, testing bipartiteness, Directed Acyclic Graphs and Topological Ordering, Minimum Spanning Trees.