

UNIVERSITY OF PETROLEUM & ENERGY STUDIES**College of Engineering Studies****Dehradun****COURSE PLAN**

Programme : B. Tech – CSE (All IBM Branches)

Course : Design and Analysis of Algorithms Lab

Course Code : CSEG 2103

No. of credits : 1

Semester : III

Session : 2020(July-Dec)

Batch : 2019-23

Prepared by : Mr. Vidyanand mishra

Email : Vidyanand.mishra@ddn.upes.ac.in

Co-Faculties :

Approved By

Faculty

UPES Campus

“Energy Acres”

P.O. Bidholi, , Dehradun

HOD/ Prog. Head

Tel : +91-135-2770137

Fax : +91 135- 27760904



COURSE PLAN

A. PREREQUISITE:

- a. Basic Knowledge of Programming and Data Structures.
- b. Basic Knowledge of Advanced Data Structure.

B. PROGRAM OUTCOMES (POs) and PROGRAM SPECIFIC OUTCOMES (PSOs) for DESIGN AND ANALYSIS OF ALGORITHMS:

B1. PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

- PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- PO2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

B2. Program Specific Outcomes (PSOs)

Computer Science Engineering with specialization in CSF, Engineering Graduates will be able to:

PSO 1. Perform system and application programming using computer system concepts, concepts of Data Structures, algorithm development, problem solving and optimizing techniques,

PSO 2. Apply software development and project management methodologies using concepts of front-end and back-end development and emerging technologies and platforms.

PSO 3. Apply computing knowledge to assess, design and propose cyber security solutions and perform forensic procedures on digital systems and cyber world using tools and technologies in the area of cyber security and cyber forensics.

C. COURSE OUTCOMES FOR DESIGN AND ANALYSIS OF ALGORITHMS

At the end of this course student should be able to

CO1. Apply mathematical techniques to find the Complexity of an algorithm design

CO2. Understand and analyze algorithms and estimate their worst and average case behavior.

CO3. Study good principles of algorithm designs

CO4. Design an appropriate data structure to reduce the complexity of an algorithm.

CO5. Implement algorithms in a programming language.

CO6. Understand about P, NP hard and NP Complete problems.

Relationship between the Course Outcomes (COs) and Program Outcomes (POs)

Mapping between COs and POs		
	Course Outcomes (COs)	Mapped Programmed Outcomes/PSO
CO1	Apply mathematical techniques to find the Complexity of an algorithm design	PO1, PSO 3
CO2	Understand and analyze algorithms and estimate their worst and average case behavior.	PO 2, PO 3, PO5, PSO2
CO3	Study good principles of algorithm designs	PO1, PO 2, PSO 3, PO5
CO4	Design an appropriate data structure to reduce the complexity of an algorithm	PO 3, PO4, PSO1, PSO3
CO5	Implement algorithms in a programming language.	PO4, PSO2

CO6	Understand about P, NP hard and NP Complete problems.	PSO2,PSO3,PO5
-----	---	---------------

Table: Correlation of POs and PSOs v/s COs

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2														2
CO2		1	2		3									2	
CO3	1	2			2										2
CO4			3	2									2		2
CO5			2	2										2	
CO6					2									2	2

		Engineering Knowledge	Problem analysis	Design/development of solutions	Conduct investigations of complex problems	Modern tool usage	The engineer and society	Environment and sustainability	Ethics	Individual or team work	Communication	Project management and finance	Life-long Learning	Perform system and application programming using computer system concepts, concepts of Data Structures, algorithm development, problem solving and optimization techniques	Apply software development and project management methodologies using concepts of front-end and back-end development and emerging technologies and software	Apply computing knowledge to assess, design and propose cyber security solutions and perform forensic procedures on digital systems and cyber world using tools and technologies in the area of cyber security and cyber forensics
Course Code	Course Title	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CSEG 2103	Design and Analysis of Algorithms	2	2	3	2	2							2	2	2	3

1: Slight (Low)

2: Moderate (Medium)

3: Substantial (High)

The course provides the deep understanding of advanced data structures and to implement various algorithms implementation in C/C++ programming language

Subject: Design and Analysis of Algorithms Lab, Course: B.Tech-CSE-All branches

Duration: 120 Minutes

OBJECTIVES:

1. The overall objective of the modules is that the student should be able to distinguish between the different algorithms designing methods.
2. The student should be able to design and implement the algorithms using C/C++ code.
3. The student should be able analyze the performance of designing methods
4. The student should be able to apply the algorithms for implementation of applications
5. The student should be able to design an efficient algorithms for problem solving

COURSE OUTLINE

Experiment-1: Brute Force Techniques

Experiment-2: Divide And Conquer-I

Experiment-3: Divide And Conquer-Ii

Experiment-4: Greedy Method

Experiment-5: Graph Algorithms

Experiment-6: Dynamic Programming-I

Experiment-7: Dynamic Programming-Ii

Experiment-8: Sorting In Linear Time Lower Bounds

Experiment-9: Backtracking

Experiment-10: Branch and Bound

D. PEDAGOGY

1.1 PREREQUISITE : Basic knowledge of Computer Science such as fundamentals & logic for solving programs, knowledge of basic mathematics.

1.2 CORE REQUISITE: Learning aptitude, Basic knowledge of programming, Data Structure concepts.

E. COURSE COMPLETION PLAN

No. of Labs Planned/taken		No. of Experiments planned / conducted		No. of Quizzes/Viva planned / conducted		No. of tests planned / conducted		% of Syllabus completed
10		10		2		2		

F. EVALUATION & GRADING

Students will be evaluated based on the following 2 stages.

Continuous assessment - 100%

INTERNAL ASSESSMENT:

WEIGHTAGE- 100% Continuous assessment shall be based on the following:

Sl. No.	Description	%of Weightage out of 100%
1	Algorithm Design	20
2	Coding Style	40
3	Execution/Bug Finding	20
4	Records	20

Internal Assessment Record Sheet will be displayed on LMS at the end of the semester i.e. in the last week of regular classroom teaching.

CONTINUOUS ASSESSMENT: Based on the weekly evaluation of the experiments actually performed by the students in the Laboratory and submitted on the same day or on the very next turn. A group project will be submitted and that will be evaluated.

The continuous Assessment will be displayed on LMS/ICOS on monthly basis i.e. on the last two or three working days of every month.

MANDATORY: A group project assignment will be submitted by the students. Project progress / VIVA: Progress of the project work will be discussed by the students twice the term each time a viva based exercise will be followed. Those who fail to do so shall be marked as absent and shall lose their marks.

The marks obtained by the students will be displayed on LMS after evaluation.

GENERAL DISCIPLINE: Based on student's regularity, punctuality, sincerity and behavior in the class.

The marks obtained by the students will be displayed on LMS at the end of semester.

GRADING: The overall marks obtained at the end of the semester comprising the above two mentioned shall be converted to a grade.

G. DETAILED SESSION PLAN

EXPERIMENT1: Title: BRUTE FORCE TECHNIQUES

1. Sort a given set of elements using bubble and selection sort and hence find the time required to sort elements
2. Perform linear search and find the time required to search an element.
3. Given a string called TEXT with 'n' characters and another string called PATTERN with 'm' characters ($m \leq n$). Write a program which implements brute force string matching to search for a given pattern in the text. If the pattern is present then find the position of first occurrences of Pattern in that Text.

EXPERIMENT-2: Title: DIVIDE AND CONQUER-I

1. Implement Binary search and linear search and determine the time required to search an element. Repeat the experiment for different values of n, the number of elements in the list to be searched and plot a graph of the time taken versus n.

2. Search a elements using the Binary search method and determine the time required to search the element. Repeat the experiment for different values of n, to search for the element in the list and plot a graph of the time taken versus n.
3. Sort a given set of elements using the Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.

EXPERIMENT-3: Title: DIVIDE AND CONQUER-II

1. Sort a given set of elements using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
2. Sort a given set of elements using the insertion Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
3. Implement Strassen's matrix multiplication and compare the complexity with normal matrix multiplication

EXPERIMENT-4: Title: GREEDY METHOD

1. Implement the activity-selection problem (You are given n activities with their start and finish times. Select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a time.

Example: Consider the following 6 activities.

Start [] = {1, 3, 0, 5, 8, 5}; finish [] = {2, 4, 6, 7, 9, 9};

The maximum set of activities that can be executed by a single person is {0, 1, 3, 4}).

2. Consider the following scheduling problem. You are given n jobs. Job i is specified by an earliest start time s_i , and a processing time p_i . We consider a preemptive version of

the problem where a job's execution can be suspended at any time and then completed later. For example if $n = 2$ and the input is $s_1 = 2, p_1 = 5$ and $s_2 = 0, p_2 = 3$, then a legal preemptive schedule is one in which job 2 runs from time 0 to 2 and is then suspended. Then job 1 runs from time 2 to 7 and secondly, job 2 is completed from time 7 to 8. The goal is to output a schedule that minimizes $\sum C_j = 1$, where C_j is the time when job j is completed and j runs from 1 to n . In the example schedule given above, $C_1 = 7$ and $C_2 = 8$.

3. To find Optimal solution for a Knapsack Problem using Greedy Method
4. Implement the file or code compression using Huffman's algorithm.

EXPERIMENT-5: Title: Graph Algorithms

1. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm
2. Find Minimum Cost Spanning Tree of a given undirected graph using prim's algorithm
3. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm
4. From a given vertex in a weighted connected graph, find shortest paths to other vertices negative weights (using Bellman-Ford algorithm).
5. Print all the nodes reachable from a given starting node in a digraph using BFS method. Check whether a given graph is connected or not using DFS method.

EXPERIMENT-6: Title: DYNAMIC PROGRAMMING-I

1. Implement 0/1 Knapsack problem using Dynamic Programming
2. Given a rod of length n inches and an array of prices that contains prices of all pieces of size smaller than n . Determine the maximum value obtainable by cutting up the rod and selling the pieces. For example, if length of the rod is 8 and the values of different pieces are given as following, then the maximum obtainable value is 22 (by cutting in two pieces of lengths 2 and 6)

Length	1	2	3	4	5	6	7	8

Price	1	5	8	9	10	17	17	20

3. The order in which we parenthesize the product affects the number of simple arithmetic operations needed to compute the product, or the *efficiency* .(using matrix chain multiplication)

EXPERIMENT-7: Title: DYNAMIC PROGRAMMING-II

1. Implement Floyd's algorithm for the All-Pairs-Shortest-Paths problem.
2. Compute the transitive closure of a given directed graph using Warshall's algorithm
3. Given two sequences, find the length of longest subsequence present in both of them. A subsequence is a sequence that appears in the same relative order, but not necessarily contiguous. For example, "abc", "abg", "bdf", "aeg", "acefg" etc are subsequences of "abcdefg".

EXPERIMENT-8: Title: SORTING IN LINEAR TIME LOWER BOUNDS

1. Write a program to perform count sort
2. Write a program to perform radix sort

EXPERIMENT-9: Title: BACKTRACKING

1. Find a subset of a given set $S = \{s_1, s_2, \dots, s_n\}$ of n positive integers whose sum is equal to a given positive integer d . For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$ there are two solutions $\{1, 2, 6\}$ and $\{1, 8\}$. A suitable message is to be displayed if the given problem instance doesn't have a solution.
2. Implement N Queen's problem using Back Tracking.
3. Implementation of GRAPH COLORING
4. Implement Sudoku problem
5. Implement the Hamilton cycle problem.

EXPERIMENT-10: Title: Branch and Bound

1. implement the 1/0 knapsack problem using branch and bound

2. The Hamiltonian cycle problem is NP-complete.
3. The sum of subset problem using NP-complete.

H. SUGGESTED READINGS:

Text Book

T1: Introduction to Algorithms, Third Edition, By **Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, 2009.**

Reference Books:

R1: Introduction to the Design and Analysis of Algorithms, **Dr. Anany Levitin,**

Video Resources:

- [V1]. <https://www.youtube.com/playlist?list=PL08885AEAE85EA836>
[V2]. <https://nptel.ac.in/downloads/106106131/>

GUIDELINES

Cell Phones and other Electronic Communication Devices: Cell phones and other electronic communication devices (such as Blackberries/Laptops) are not permitted in classes during Tests or the Mid/Final Examination. Such devices **MUST** be turned off in the class room.

E-Mail and LMS: Each student in the class should have an e-mail id and a pass word to access the LMS system regularly. Regularly, important information – Date of conducting class tests, guest lectures, via LMS. The best way to arrange meetings with us or ask specific questions is by email and prior appointment. All the assignments preferably should be uploaded on LMS. Various research papers/reference material will be mailed/uploaded on LMS time to time.

Attendance: Students are required to have **minimum attendance of 75%** in each subject. Students with less than said percentage shall **NOT** be allowed to appear in the end semester examination.

Passing criterion: Student has to secure minimum 35 marks individually in both the ‘Total Marks’ in order to pass in that paper.

- Passing Criterion for B. Tech: minimum 35 marks .
- Passing Criterion for M. Tech: minimum 40% of the highest marks in the class

Course outcome assessment: To assess the fulfilment of course outcomes two different approaches have been decided. Degree of fulfillment of course outcomes will be assessed in different ways through direct assessment and indirect assessment. In Direct



Assessment, it is measured through quizzes, tests, assignment, Mid-term and/or End-term examinations. It is suggested that each examination is designed in such a way that it can address one or two outcomes (depending upon the course completion). Indirect assessment is done through the student survey which needs to be designed by the faculty (sample format is given below) and it shall be conducted towards the end of course completion. The evaluation of the achievement of the Course Outcomes shall be done by analyzing the inputs received through Direct and Indirect Assessments and then corrective actions suggested for further improvement.