# +FR. Conceicao Rodrigues College Of Engineering

## Department of Computer Engineering

**S.E. (AI&DS) (semester III) (2022-2023)**

**Lesson Plan**

**Branch: AI&DS Semester:IV**

**Subject: Analysis of Algorithms Credits-3**

**Course code: CSC402**

|  |  |
| --- | --- |
| Course Title: **Analysis of Algorithms** | SEE: 3 Hours – Theory  & Oral Examination |
| Total Contact Hours: 36 Hours | Duration of SEE: 3 Hrs |
| SEE Marks: 80 (Theory) + 20 (IA) |  |
| Lesson Plan Author: | Date: |
| Checked By: | Date: |

**Pre-requisite:** Data structure concepts, Discrete structures

**Syllabus:**

|  |  |
| --- | --- |
| **Course Objectives:** The course aims: | |
| 1 | To provide mathematical approaches for Analysis of Algorithms |
| 2 | To provide mathematical approaches for Analysis of Algorithms |
| 3 | To provide mathematical approaches for Analysis of Algorithms |
| **Course Outcomes:** | |
| 1 | Students will be able to analyze the running time and space complexity of algorithms |
| 2 | Students will be able to describe, apply and analyze the complexity of divide and conquer strategy. |
| 3 | Students will be able to describe, apply and analyze the complexity of greedy strategy |
| 4 | Students will be able to describe, apply and analyze the complexity of dynamic programming strategy. |
| 5 | Students will be able to explain and apply backtracking, branch and bound. |
| 6 | Students will be able to explain and apply string matching techniques. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Module** |  | **Detailed Content** | **Hours** |
| **1** |  | **Introduction** | 8 |
|  | 1.1 | Performance analysis, space, and time complexity Growth of function, Big-Oh, Omega Theta notation Mathematical background for algorithm analysis. Complexity class: Definition of P, NP, NP-Hard, NP-Complete Analysis of selection sort, insertion sort. |  |
|  | 1.2 | Recurrences: The substitution method, Recursion tree method, Master method |  |
| **2** |  | **Divide and Conquer Approach** | 6 |
|  | 2.1 | General method, Merge sort, Quick sort, Finding minimum and maximum algorithms and their Analysis, Analysis of Binary search |  |
| **3** |  | **Greedy Method Approach** | 6 |
|  | 3.1 | General Method, Single source shortest path: Dijkstra Algorithm Fractional Knapsack problem, Job sequencing with deadlines, Minimum cost spanning trees: Kruskal and Prim‟s algorithms |  |
| **4** |  | **Dynamic Programming Approach** | 9 |
|  | 4.1 | General Method, Multistage graphs, Single source shortest path: Bellman Ford Algorithm All pair shortest path: Floyd Warshall Algorithm, Assembly-line scheduling Problem0/1 knapsack Problem, Travelling Salesperson problem, Longest common subsequence |  |
| **5** |  | **Backtracking and Branch and bound** | 6 |
|  | 5.1 | General Method, Backtracking: N-queen problem, Sum of subsets, Graph coloring |  |
|  | 5.2 | Branch and Bound: Travelling Salesperson Problem, 15 Puzzle problem |  |
| **6** |  | **String Matching Algorithms** | 4 |
|  | 6.1 | The Naïve string-matching algorithm, The Rabin Karp algorithm, The Knuth-Morris-Pratt algorithm |  |

|  |  |
| --- | --- |
| **Textbooks:** | |
| 1 | T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, “Introduction to algorithms”, 2nd Edition, PHI Publication 2005. |
| 2 | Ellis Horowitz, Sartaj Sahni, S. Rajsekaran. “Fundamentals of computer algorithms” University Press. |
| **References:** | |
| 1 | Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, “Algorithms”, Tata McGrawHill Edition. |
| 2 | S. K. Basu, “Design Methods and Analysis of Algorithm”, PHI |

## Course Outcomes:

At the end of the course students will be able to:

|  |  |  |
| --- | --- | --- |
| CSC402.1 | Analyze the running time and space complexity of algorithms. | (Analyze) |
| CSC 402.2 | Describe, apply and analyze the complexity of divide and conquer strategy | (Analyze and Application) |
| CSC 402.3 | Describe, apply and analyze the complexity of greedy strategy. | (Analyze and Application) |
| CSC 402.4 | Describe, apply and analyze the complexity of dynamic programming strategy. | (Analyze and Application) |
| CSC 402.5 | Explain and apply backtracking, branch and bound. | (Analyze and Application) |
| CSC 402.6 | Explain and apply string matching techniques. | ( Application) |

#### Program Outcomes (POs)

Engineering Graduates will be able to

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **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.
3. **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 public health and safety, and the cultural, societal, and environmental considerations.
4. **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.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling of complex engineering activities with an understanding of the limitations.
6. **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.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **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.
11. **Project Management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognized the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

#### Program Specific Outcomes (PSOs)

Student will have ability to

1. The graduate of BE in AI&DS program will have strong foundation and ability to use cutting– edge AI tools and techniques to innovate and develop new solutions.

## Mapping of CO and PO/PSO

Relationship of course outcomes with program outcomes: Indicate 1 (low importance), 2 (Moderate Importance) or 3 (High Importance) in respective mapping cell.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | PO1  (Engg Know) | PO2  (Analysis) | PO3  (De sign) | PO4  (inve stiga) | PO5  (tools) | PO6  (engg Soci) | PO7  (Env) | PO8  (Eth) | PO9  (ind Team) | PO10  (comm.) | PO11 (PM) | PO12  (life Long) | PSO 1 |
| CSC402.1 | 3 | 3 | 2 | 3 |  |  |  |  |  |  |  | 2 | 3 |
| CSC402.2 | 3 | 3 | 3 | 2 |  |  |  |  |  |  |  |  | 3 |
| CSC402.3 | 3 | 3 | 3 | 2 |  |  |  |  |  |  |  |  | 3 |
| CSC402.4 | 3 | 3 | 3 | 2 |  |  |  |  |  |  |  |  | 3 |
| CSC402.5 | 3 | 3 | 3 | 2 |  |  |  |  |  |  |  |  | 3 |
| CSC402.6 | 3 | 3 | 3 | 2 |  |  |  |  |  |  |  |  | 3 |
| TOTAL | 18 | 18 | 17 | 13 |  |  |  |  |  |  |  | 2 | 18 |
| Course  To PO | 3 | 3 | 3 | 3 |  |  |  |  |  |  |  | 2 | 3 |

**Justification**

PO1: CS402.1, CS402.2, CS402.3, CS402.4, CSC402.5, CSC402.6 maps to PO1 because engineering graduates will be able to use any programming approaches like greedy, dynamic and algorithms to provide a solution to complex engineering problems.

PO2: CS402.1, CS402.2, CS402.3, CS402.4, CSC402.5, CSC402.6 also maps to PO2 as engineering graduates will be able to analyze a problem to make a decision about which approach, algorithm, searching or sorting techniques are appropriate to solve engineering problems and evaluate their solutions.

PO3: CS402.1, CS402.2, CS402.3, CS402.4, CSC402.5, CSC402.6 also maps to PO3 as engineering graduates will be able to analyze and design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.

PO4: CS402.1, CS402.2, CS402.3, CS402.4, CSC402.5, CSC402.6 are maps to PO4 since students will be able to use research-based knowledge and research methods including design of experiments, analysis of algorithm to provide valid conclusions.

PO12: CS402.1 are maps to PO12 since students will be able to recognized the need for, and have the preparation and ability to engage in independent and life-long learning using time and space complexity in the broadest context of technological change.

PSO1: Slightly the student will study of fundamental concepts of Algorithms to analyses and develop their own algorithms and implement them using high-level programming languages.

**CO Assessment tools:**

## CSC402.1 Analyze the running time and space complexity of algorithms. Target: 2.5

|  |  |  |  |
| --- | --- | --- | --- |
| **Direct Method** | **Weightage** | **Target** | **Marks** |
| Unit Test 1 | 0.2 | 60% of students will minimum score 80% marks | 13 M |
| University Exam (Theory ) | 0.2(TH) | 60% of students will minimum score 60% marks | 80M |
| University Exam (Practical ) | 0.25(PR) | 60% of students will minimum score 80% marks | 25M |
| Assignment 1 | 0.1 | 75% of students will minimum score 75% marks | 10M |
| Lab Performance | 0.2 | 75% of students will minimum score 70% marks | 100M |
| **Indirect Method** | **Weightage** | **Target** |  |
| Course Exit Survey | 1 | 85% of students strongly agree and agree |  |

1. **CSC402.1:**
2. **Direct Methods (80%):** Unit Test 1+UniExam+ Assignment+ Lab performance

#### CSC402.1 dm = 0.2UT1 +0.1Q1+ 0.2UniexamTH+0.25UniexamPR+ 0.1Assignment +0.2Lab

1. **InDirect Methods(20%):** Course exit survey

**CSC402.1*idm***

## CSC402.1 = 0.8\* CSC402.1dm + 0.2\* CSC402.1idm

1. **CSC402.2** Describe, apply and analyze the complexity of divide and conquer strategy. **Target: 2.5**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Direct Method** | | **Weightage** | | **Target** | | **Marks** | |
|  | Unit Test 1 | | 0.2 | | 60% of students will minimum score 80% marks | | 16 M | |
|  | University Exam (Theory ) | | 0.25(TH) | | 60% of students will minimum score 80% marks | | 80M | |
|  | University Exam (Practical ) | | 0.25(PR) | | 60% of students will minimum score 60% marks | | 25M | |
|  | Lab Performance | | 0.2 | | 75% of students will minimum score 70% marks | | 20M | |
|  | Assignment 1 | | 0.1 | | 75% of students will minimum score 75% marks | | 10M | |
|  | |  | |  | |  | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Indirect Method** | **Weightage** | **Target** |  |
| Course Exit Survey | 1 | 85% of students strongly agree and agree |  |

**CSC**402**.2:**

1. **Direct Methods (80%):** Unit Test 1+UniExam+ Assignment+ Lab performance

#### CSC402.2 dm = 0.2UT1 + 0.25UniexamTH+0.25UniexamPR+ 0.1Assignment +0.2Lab

1. **InDirect Methods(20%):** Course exit survey

**CSC402.1*idm***

## CSC402.2 = 0.8\* CSC402.2dm + 0.2\* CSC402.2idm

**3) CSC**402**.3.** Describe, apply and analyze the complexity of greedy strategy..  **Target: 2.7**

|  |  |  |  |
| --- | --- | --- | --- |
| **Direct Method** | **Weightage** | **Target** | **Marks** |
| Unit Test 1 | 0.1 | 60% of students will minimum score 80% marks | 08 M |
| Assignment 2 | 0.3 | 60% of students will minimum score 80% marks | 20M |
| University Exam (Theory ) | 0.25(TH) | 60% of students will minimum score 60% marks | 80M |
| University Exam (Practical ) | 0.25(PR) | 60% of students will minimum score 70% marks | 25M |
| Lab Performance | 0.2 | 75% of students will minimum score 70% marks | 20M |
| **Indirect Method** | **Weightage** | **Target** |  |
| Course Exit Survey | 1 | 85% of students strongly agree and agree |  |

**CSC**402**.3:** Describe, apply and analyze the complexity of greedy strategy..

* 1. **Direct Methods (80%):** Unit Test 1+ Uni Exam+ Case study+ Lab

performance

#### CSC402.3dm = 0.1UT1 + 0.25UniexamTH+ 0.25UniexamPr +0.2Lab

* 1. **InDirect Methods(20%):** Course exit survey

**CSC402.3*idm***

## CSC402.3 = 0.8\* CSC402.3dm + 0.2\* CSC402.3idm

## CSC402.4 Describe, apply and analyze the complexity of dynamic programming strategy. Target: 2.8

|  |  |  |  |
| --- | --- | --- | --- |
| **Direct Method** | **Weightage** | **Target** | **Marks** |
| Unit Test 1 | 0.2 | 60% of students will minimum score 80% marks | 20 M |
| University Exam (Theory ) | 0.2(TH) | 60% of students will minimum score 60% marks | 80M |
| University Exam (Practical ) | 0.25(PR) | 60% of students will minimum score 80% marks | 25M |
| Assignment 2 | 0.1 | 75% of students will minimum score 75% marks | 10M |
| **Indirect Method** | **Weightage** | **Target** |  |
| Course Exit Survey | 1 | 85% of students strongly agree and agree |  |

**CSC**402**.4:**

1. **Direct Methods (80%): Unit Test 1+ UniExam+ Assignment**

**CSC402.4dm = 0.2UT1 + 0.25UniexamTH+0.25UniexamPR+ 0.1Assignment**

**b) InDirect Methods(20%): Course exit survey**

**CSC402.4*idm***

## CSC402.4 = 0.8\* CSC402.4dm + 0.2\* CSC402.4idm

## CSC402.5 Explain and apply backtracking, branch and bound

**Target: 2.8**

|  |  |  |  |
| --- | --- | --- | --- |
| **Direct Method** | **Weightage** | **Target** | **Marks** |
| Unit Test 2 | 0.1 | 60% of students will minimum score 80% marks | 04 M |
| Assignment 2 | 0.3 | 60% of students will minimum score 80% marks | 20M |
| University Exam (Theory ) | 0.2(TH) | 60% of students will minimum score 60% marks | 80M |
| University Exam (Practical ) | 0.2(PR) | 60% of students will minimum score 70% marks | 25M |
| Lab Performance | 0.2 | 75% of students will minimum score 70% marks | 20M |
| **Indirect Method** | **Weightage** | **Target** |  |
| Course Exit Survey | 1 | 85% of students strongly agree and agree |  |

**CSC**402**.5:**

1. **Direct Methods (80%): Unit Test 2+ UniExam+ Assignment+ Lab performance**

**CSC402.5dm = 0.2UT1 + 0.25UniexamTH+0.25UniexamPR+ 0.1Assignment + 0.2LabPerf**

**b) InDirect Methods(20%): Course exit survey**

**CSC402.5*idm***

## CSC402.5 = 0.8\* CSC402.5dm + 0.2\* CSC402.5idm

## CSC402.6 Explain and apply string matching techniques. Target: 2.8

|  |  |  |  |
| --- | --- | --- | --- |
| **Direct Method** | **Weightage** | **Target** | **Marks** |
| Demonstration of Mini Project | 0.75 | 75% of students will minimum score 70% marks | 20M |
| Report | 0.25 | 75% of students will minimum  score 70% marks |  |
| **Indirect Method** | **Weightage** | **Target** |  |
| Course Exit Survey | 1 | 75% of students strongly agree and agree |  |

**CSC**402**.6:**

* 1. **Direct Methods (80%):** Mini Project + Report

#### CSC402.6dm = 0.75MiniProject+0.25Report

* 1. **InDirect Methods(20%):** Course exit survey

**CSC402.6*idm***

## CSC402.6 = 0.8\* CSC402.6dm + 0.2\* CSC402.6idm

**Course Outcomes Target:**

|  |  |
| --- | --- |
| CSC402.1 | Analyze the running time and space complexity of algorithms.**Target level: 2.5** |
| CSC402.2 | Describe, apply and analyze the complexity of divide and conquer strategy.**Target level: 2.5** |
| CSC402.3 | Describe, apply and analyze the complexity of dynamic programming strategy.**Target level: 2.7** |
| CSC402.4 | Describe, apply and analyze the complexity of dynamic programming strategy. **Target level: 2.8** |
| CSC402.5 | Explain and apply backtracking, branch and bound. **Target level: 2.8** |
| CSC402.6 | Explain and apply string matching techniques. **Target level: 2.8** |

**CO Attainment of previous years**

|  |  |
| --- | --- |
| Course Outcomes | 2021-22 |
| CSC402.1 Analyze the running time and space complexity of algorithms. | 2.48 |
| CSC 402.2 Describe, apply and analyze the complexity of divide and conquer strategy. | 2.28 |
| CSC 402.3 Describe, apply and analyze the complexity of greedy strategy. | 2.28 |
| CSC 402.4 Describe, apply and analyze the complexity of dynamic programming strategy.. | 2.28 |
| CSC 402.5 Explain and apply backtracking, branch and bound. | 2.28 |
| CSC 402.6 Explain and apply string matching techniques. | 2.28 |

**Rubrics for Lab Experiments:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No** | **Performance Indicator** | **Excellent** | **Good** | **Satisfactory** | **Unsatisfactory** |
| 1) | Completeness and correctness [4] | Well commented and formatted, program functions correctly for all input cases. [4M] | Comparatively less use of comments, Inconsistent formatting. Program functions correctly for all input cases.[3M] | Inconsistent comments and formatting.  Program functions correctly for most of the input cases.[2M] | Improper formatting, No comments. Program functions correctly for very limited cases [1M] |
| 2) | Efficiency [3] | The code could be reused as a whole or each routine could be reused. It is readable and easy to understand [3M] | Most of the code could be reused in other programs. It is fairly readable and easy to understand  [2M] | Only Some parts of the code could be reused in other programs. The code is unnecessarily long and repeated. [1M] | The code lacks reusability. It is huge and repeated at many places[0M] |
| 3) | Post Lab Questions [2] | Answers to all questions are correct and explained in depth.  [2M] | Answers to most of the questions are correct but not explained in much depth.  [2-1.5M] | Answers of few questions are incorrect and lacks sufficient depth  [0-1M] | Answers to most of the questions are incorrect and not explained in depth.  [0 mark] |
| 4) | Promptness [1] | The laboratory report is submitted on time  [1 mark] | The laboratory report is submitted next day.  [0.5 marks] | The laboratory report is submitted in next practical session.  [0 marks] |  |

**Rubrics for Assignments:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance Indicator** | **Excellent** | **Good** | **Below average** |
| Timeline(2) | submitted on time or early  (2) | Submitted next day (1) | Submitted in same week  (0.5) |
| Organization (2) | Well organized,  neat and clear handwriting, easy to read.(2) | Organized to some extent, handwriting is neat(1) | Poorly organized(0.5) |
| Level of content (4) | All points are covered and answered accurately (4) | Some important points are omitted /addressed minimally (3) | Many important points are missing and the answers are not accurate.  (2) |
| Depth and breadth discussion (2) | Each point is illustrated in depth with proper justification wherever  required (2) | Few points are not illustrated in depth and have minimal justification (1) | Many points are not illustrated in depth and justification missing.(0.5) |

**Lesson Plan: AOA**

**Semester IV Year: 2022-23**

**Modes of Content Delivery:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| I | Class Room Teaching | v | Self-Learning Online Resources | Ix | Industry Visit |
| Ii | Tutorial | vi | Slides | X | Group  Discussion |
| Iii | Remedial Coaching | vii | Simulations/Demonstrations | xi | Seminar |
| Iv | Lab Experiment | viii | Expert Lecture | xii | Case Study |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Planned Date** | **Actual Date** | **Topic** | **Delivery Mechanisms** |
| 1 | 9-1-23 | 9-1-23 | Performance analysis, space, and time complexity Growth of function, Big-Oh, Omega | Classroom teaching |
| 2 | 10-1-23 | 10-1-23 | Performance analysis, space, and time complexity Growth of function, Big-Oh, Omega | PPT presentation in classroom |
| 3 | 11-1-23 | 11-1-23 | Complexity class: Definition of P, NP, NP-Hard | PPT presentation in classroom |
| 4 | 16-1-23 | 16-1-23 | NP-Complete Analysis of selection sort, | Classroom teaching, Self-learning online resources |
| 5 | 17-1-23 | 17-1-23 | Analysis of insertion sort | PPT presentation in classroom |
| 6 | 18-1-23 | 18-1-23 | Recursion tree method, Master method | Demonstration of program using Code Blocks in classroom |
| 7 | 23-1-23 | 23-1-23 | Recursion tree method, Master method | Demonstration of program using CodeBlocks in classroom |
| 8 | 24-1-23 | 24-1-23 | Recursion tree method, Master method | Demonstration of program using CodeBlocks in classroom |
| 9 | 25-1-23 | 25-1-23 | General method, Merge sort, Quick sort | PPT presentation in classroom |
| 10 | 30-1-23 | 30-1-23 | Finding minimum and maximum algorithms | Classroom teaching, Self learning online resources |
| 11 | 31-1-23 | 31-1-23 | Finding minimum and maximum algorithms and their Analysis | Demonstration of program using CodeBlocks in classroom |
| 12 | 1-2-23 | 1-2-23 | Analysis of Binary search. | PPT presentation in classroom |
| 13 | 6-2-23 | 6-2-23 | Analysis of Binary search. | Classroom teaching |
| 14 | 7-2-23 | 7-2-23 | General Method | PPT presentation in classroom |
| 15 | 8-2-23 | 8-2-23 | Single source shortest path: Dijkstra Algorithm | PPT presentation in classroom |
| 16 | 13-2-23 | 13-2-23 | Fractional Knapsack problem | Classroom teaching, Self learning online resources |
| 17 | 14-2-23 | 14-2-23 | Job sequencing with deadlines, | Demonstration of program using CodeBlocks in classroom |
| 18 | 15-2-23 | 15-2-23 | Minimum cost spanning trees: Kruskal and Prim‟s algorithms | PPT presentation in classroom |
| 19 | 20-2-23 | 20-2-23 | Minimum cost spanning trees: Kruskal and Prim‟s algorithms | Classroom teaching |
| 20 | 21-2-23 | 21-2-23 | General Method, Multistage graphs | PPT presentation in classroom |
| 21 | 22-2-23 | 22-2-23 | General Method, Multistage graphs | PPT presentation in classroom |
| 22 | 27-2-23 | 27-2-23 | General Method, Multistage graphs | Classroom teaching, Self learning online resources |
| 23 | 6-3-23 | 6-3-23 | All pair shortest path: Floyd Warshall Algorithm | Demonstration of program using CodeBlocks in classroom |
| 24 | 8-3-23 | 8-3-23 | Assembly-line scheduling Problem0/1 knapsack Problem | PPT presentation in classroom |
| 25 | 13-3-23 | 13-3-23 | Assembly-line scheduling Problem0/1 knapsack Problem | Classroom teaching |
| 26 | 14-3-23 | 14-3-23 | Travelling Salesperson problem, Longest common subsequence | PPT presentation in classroom |
| 27 | 15-3-23 | 15-3-23 | Travelling Salesperson problem, Longest common subsequence | PPT presentation in classroom |
| 28 | 20-3-23 |  | Backtracking and Branch and bound General Method | Classroom teaching, Self learning online resources |
| 29 | 21-3-23 |  | Backtracking: N-queen problem, | Demonstration of program using CodeBlocks in classroom |
| 30 | 22-3-23 |  | Backtracking: N-queen problem, | PPT presentation in classroom |
| 31 | 27-3-23 |  | Sum of subsets, | Classroom teaching |
| 32 | 28-3-23 |  | Graph coloring | PPT presentation in classroom |
| 33 | 3-4-23 |  | Branch and Bound: Travelling Salesperson Problem | PPT presentation in classroom |
| 34 | 5-4-23 |  | 15 Puzzle problem | Classroom teaching, Self learning online resources |
| 35 | 10-4-23 |  | String Matching Algorithms | Demonstration of program using CodeBlocks in classroom |
| 36 | 11-4-23 |  | String Matching Algorithms | PPT presentation in classroom |
| 37 | 12-4-23 |  | The Rabin Karp algorithm | Classroom teaching |
| 38 | 17-4-23 |  | The Rabin Karp algorithm | PPT presentation in classroom |
| 39 | 18-4-23 |  | The Knuth-Morris-Pratt algorithm | Classroom teaching using black board |
| 40 | 19-4-23 |  | The Knuth-Morris-Pratt algorithm | Classroom teaching using black board |
| Remedial classes | | | | |
| 41 | 20-4-23 |  | Revision | Online class in the evening |
| 42 | 21-4-23 |  | Revision | Online class in the evening |

## Subject: Analysis of Algorithms Lab Credits-1

**Course code: CSL401**

**Lab Objectives:**

CSL401.1 To introduce the methods of designing and analyzing algorithms

CSL401.2 Design and implement efficient algorithms for a specified application

CSL401.3 Strengthen the ability to identify and apply the suitable algorithm for the given real-world problem.

CSL401.4 Analyze worst-case running time of algorithms and understand fundamental algorithmic problems.

**Syllabus:**

### Implementation of Selection sort. \*

1. Implementation of Insertion sort. \*
2. Implementation of Finding Minimum and Maximum
3. Implementation of Merge sort.
4. Implementation of Quick sort
5. Implementation of Binary search \*
6. Implementation of Single source shortest path- Dijkstra.
7. Implementation of Fractional Knapsack problem
8. Implementation of Job sequencing with deadlines.
9. Implementation of Minimum cost spanning trees-Kruskal and Prim‟s algorithm
10. Implementation of Single source shortest path- Bellman Ford.
11. Implementation of Single source shortest path- Bellman Ford.
12. Implementation of 0/1 knapsack
13. Implementation of Travelling salesperson problem
14. Implementation of Longest common subsequence
15. Implementation of N-queen problem.
16. Implementation of Sum of subsets.\*
17. Implementation of Graph coloring.\*
18. Implementation of Naïve string-matching Algorithms.
19. Implementation of Rabin Karp algorithm.
20. Implementation of Knuth-Morris-Pratt algorithm

**Term Work (25M): Lab Experiments (15M) +Assignment(5M)**

**List of Practicals and Lab Plan**

**Sub: Data Structures                                                                 Year (2020-21)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Aim** | **CO Mapping** | **Weekly Plan** |
| 1. | 1.1 Implementation of Selection sort. \*  1.2 Implementation of Insertion sort. \* | CSL401.1 | Third Week  Fourth Week |
| 2. | **Divide and Conquer Approach**  2.1 Finding Minimum and Maximum  2.2 Merge sort  2.3 Quick sort  2.4 Binary search | CSL401.1 | Fifth Week  Sixth Week |
| 3. | **Greedy Method Approach**  3.1 Single source shortest path- Dijkstra  3.2 Fractional Knapsack problem  3.3 Job sequencing with deadlines  3.4 Minimum cost spanning trees-Kruskal and Prim’s algorithm | CSL401.2 | Eighth Week  NinethWeek |
| 4. | **Dynamic Programming Approach**  4.1 Single source shortest path- Bellman Ford  4.2 All pair shortest path- Floyd Warshall  4.3 0/1 knapsack  4.4 Travelling salesperson problem  4.5 Longest common subsequence | CSL401.2 | TenthWeek |
| 5. | **Backtracking and Branch and bound**  5.1 N-queen problem  5.2 Sum of subsets  5.3 Graph coloring | CSL401.2 | Eleventh Week |
| 6. | **String Matching Algorithms**  6.1 The Naïve string-matching Algorithms  6.2 The Rabin Karp algorithm  6.3 The Knuth-Morris-Pratt algorithm | CSL401.2 | Twelth Week |
| 7 | Assignment 1 | CSC402.2  CSC402.2  CSC402.2 | Sixth week |
| 8 | Assignment 2 | CSC402.2  CSC402.2  CSC402.2 | Tenth week |

#### Course Outcomes (given in syllabus):

Students will be able to:

1. Implement the algorithms using different approaches.
2. Analyze the complexities of various algorithms
3. Compare the complexity of the algorithms for specific problem.

Content beyond syllabus:

* [Advanced Lists](https://www.geeksforgeeks.org/advanced-data-structures/#AdvancedLists)
* [n-ary Tree](https://www.geeksforgeeks.org/advanced-data-structures/#n-arytree)
* [Self-Balancing BSTs](https://www.geeksforgeeks.org/advanced-data-structures/#SelfbalancingBSTs)
* [Trie](https://www.geeksforgeeks.org/advanced-data-structures/#trie)

## Assignment 1

**Sub:** Analysis of Algorithms

**Class: SE AI&DS**

**Assignment Date: 15th Feb 2023 9.00 AM**

**Date of submission: 15th Feb 2023 2.00 PM**

Course outcome:

**CSC402.1: CO1: Analyze the running time and space complexity of algorithms**

**CSC402.2: Describe, apply and analyze the complexity of divide and conquer strategy.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q. No.** | **Question No.** | **CO** | **BL** | **PI** |
| **1** | **Trace the quick sort algorithm to sort the array**  **[11, 25,9,3,5,0,20,28,7]**  **into ascending order. Use the array**  **implementation exactly as described in the class. Also list the calls to quicksort and partition in the order they occur. Assume that the last element is chosen as pivot.** | CSC402.1 | 3 | 1.4.1 |
| **2** | **Compare all sorting algorithms on their running time complexities for best, average and worst case in tabular form.** | CSC402.1 | 3 | 1.4.1 |
| **3** | **Determine the running time analysis on quick sort, merge sort and insertion sort a. Do the runs on an already sorted input array. Order the algorithms according to how  their execution time grows with n.**  **b. Do all the runs on an input array sorted in reverse order. Order the algorithms  according to how their execution time grows with n.**  **c. Do the runs on an input array where all elements have the same value. Order the  algorithms on how their execution time grows with n.**  **d. Which algorithms have a running-time behavior only dependent on the input size n,  (i.e. which algorithms are not sensitive for the distribution of input elements?)** | CSC402.1 | 3 | 1.4.1 |
| **4** | **Compute the worst case complexity of the following program segment:**  **void fun(int n, int arr[]){**  **int i = 0, j = 0;**  **for(; i < n; ++i)**  **while(j < n && arr[i] < arr[j])**  **j++;**  **}** | CSC402.1 | 3 | 1.4.1 |
| **5** | **Define Master theorem. Solve the following using Master Method.**  **T=8T(n/2)+n2** | CSC402.1 | 3 | 1.4.1 |
| **6** | **Write an algorithm for finding minimum and maximum using Divide and Conquer. Explain with one  example. Also derive its complexity.** | CSC402.1 | 3 | 1.4.1 |

ASSIGNMENT NO. 2

Analysis of Algorithms YEAR 2022-23

CLASS: S.E. (AI&DS). (semester III)

DATE: 16/04/2023

Deadline: 23/04/2022

Rubrics for Assignment Grading:

|  |  |  |  |
| --- | --- | --- | --- |
| Indicator |  |  |  |
| Timeline (2) | More than one  session late (0) | One sessions late (1) | On time (2) |
| Level of content (4) | Major points are  addressed minimally (2) | Only major topics are  covered(3) | Most major and some minor criteria are  Included. Information is  Adequate (4) |
| Reading and  Understanding (4) | Superficial  at most (2) | Understood concepts but no related topics (3) | Understood concepts and related topics (4) |

Evaluation:

|  |  |  |  |
| --- | --- | --- | --- |
| Timeline | Level of Content | Reading and  Understand | Total |
|  |  |  |  |

Maps to

CO4 Describe, apply and analyse the complexity of dynamic programming strategy. CO5 Explain and apply backtracking, branch and bound.

CO6 Explain and apply string matching techniques.

 Signature of the Faculty

**FR. CONCEICAO RODRIGUES COLLEGE OF ENGG.**

**Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.**

**1.** Apply Dynamic Programming strategy to find the longest common subsequence for the  following two strings. X=ABACABB Y=BABCAB

**2.** Analyse the time complexity of the LCS.

**3.** Apply dynamic programming strategy to compute all pair shortest path for the following graph.  Explain step by step.

**4.** Analyse the time complexity of Floyd Warshall’s Algorithm.

**5.** Explain and apply backtracking to solve 4-queen problem. Show the state space tree. **6.** Explain and apply backtracking to solve sum of subset problem for following. Show the  state space tree. N=6, W={3,5,7,8,9,15} & M=20.

**7.** Explain and apply backtracking to solve sum of subset problem for following. Show the  state space tree. N=7, W={5,7,10,12,15,18, 20} & M=35. Find all possible subsets of W  that sum to M.

**8.** Explain various string-matching algorithms with one example each. Comment on  complexity of each algorithm.

**9.** A thief enters a house for robbing it. He can carry a maximal weight of 12 kg into his bag. There  are 5 items in the house with the following weights and values. What items should thief take if  he either takes the item completely or leaves it completely so that he gets maximum profit?

**10.** Analyse the time complexity of 0/1 Knapsack problem using Dynamic Programming.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| **FR. CONCEICAO RODRIGUES COLLEGE OF ENGG.**  **Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.**  **UNIT TEST 1**  **SEMESTER / BRANCH: III/AI&DS**  **SUBJECT: Analysis of Algorithm (AOA) MAX. MARKS: 20**  **DATE: 03/03/2023 TIMING:1.00pm to 2.00 pm**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Student should be able to**   |  |  | | --- | --- | | **CSC402.1** | Analyze the running time and space complexity of algorithms. | | **CSC402.2** | Describe, apply and analyze the complexity of divide and conquer strategy. | | **CSC402.3** | Describe, apply and analyze the complexity of greedy strategy. | | | | | | |
| Q.No. | | Questions | Marks | CO | BL | PI | |
| Q.1 A | | Solve the recurrence equation  **T(n)= T(n-1)+n**  Using the Backward substitution method.  **OR**  Solve the following recurrence using the Master method. **Where n>1**  **T(n) = 9T(n/3) + n3logn** | 04 | CSC  402.1 | Apply | 2.8.1 | |
| Q1 B | | Differentiate between NP – Hard and NP- Complete algorithms. (Any 4 points) | 02 | CSC  402.1 | Remember | 2.6.4 | |
| Q1 C | | **Define O , Ω , and Ө notations.** | 03 | CSC  402.1 | Understand | 1.7.1 | |
| Q.2 | | Explain insertion sort and derive its complexity. | 03 | CSC  402.2 | Remember | 1.4.1 | |
| **OR** | | | | | | | |
| Q.2 | | Explain the binary search algorithm with the help of following example;  **Find element 33 from array A{11,22,33,44,55,66,77,88}** | 03 | CSC  402.2 | Apply | 2.6.3 | |
| Q.3 | | Solve fractional Knapsack problem for the following:  **N = 6**  **P = {18, 5, 9, 10, 12, 7}**  **W = { 7, 2, 3, 5, 3, 2}**  **Max sack capacity M = 13** | 04 | CSC  402.3 | Apply | 2.6.3 | |
| **OR** | | | | | | | |
| Q.3 | | Explain Job sequencing with deadline for the given instance  **N=5**  **{P1, P2, P3, P4, P5} = {20, 15, 10, 5, 3}**  **{D1, D2, D3, D4, D5} = {2, 2, 1, 3, 3}** | 04 | CSC  402.3 | Apply | 2.6.3 | |
| Q.4 | | Write a Kruskal’s Algorithm to show its working by taking a suitable example of a graph with 6 vertices. | 04 | **CSC**  **402.3** | Understand | 1.7.1 | |
| **OR** | | | | | | | |
| Q.4 | | Write a short note on Prim’s Algorithm to find a minimum spanning tree with an example. | 04 | **CSC**  **402.3** | Understand | 1.7.1 | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **FR. CONCEICAO RODRIGUES COLLEGE OF ENGG**.  Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.  **UNIT TEST 1I**  SEMESTER / BRANCH: **IV/AI&DS**  SUBJECT: **Analysis of Algorithm (AOA)** MAX. MARKS: 20  DATE: **17/04/2023** TIMING:1.30 pm to 2.30 pm  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Student should be able to   |  |  | | --- | --- | | **CSC402.4** | Describe, apply and analyze the complexity of dynamic programming strategy. | | **CSC402.5** | Explain and apply backtracking, branch and bound. | | **CSC402.6** | Explain and apply string matching techniques. | | | | | | |
| Q.No. | Questions | Marks | CO | BL | PI |
| Q.1 | Describe the travelling sales person problem and discuss how to solve it using Dynamic Programming with example.  **OR**  What is principle of Optimality in Dynamic programming? Give a suitable example for this property of optimality in dynamic programming. | 04 | CSC  402.4 | Understand | 1.7.1 |
| Q2 | Find LCS for following String: X = ABACABB And Y = BABCAB | 04 | CSC  402.4 | Apply | 2.6.3 |
| Q3 | Discuss the Branch and Bound strategy for solving a problem. | 04 | CSC  402.5 | Understand | 1.7.1 |
| Q.4 | Explain Backtracking with n-queen problem.  **OR**  Write a short note on Graph Colouring Algorithm. | 04 | CSC  402.5 | Remember | 1.4.1 |
| Q.5 | Write a short note on (Any One)   1. Robin Karp Algorithm. 2. Knuth-Morris-Pratt algorithm. | 04 | CSC  402.6 | Remember | 2.1.3 |

**Action Taken for weak students:**

● Two remedial classes were conducted offline for weak students to teach the important topics and solve their doubts.