|  |  |
| --- | --- |
|  | **PIR MEHR ALI SHAH ARID AGRICULTURE UNIVERSITY**  **University Institute of Information Technology** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CS 542: Analysis of Algorithm** | | | | |
| **Credit Hours:** | **3(3-0)** | **Prerequisites:** | CS – 443 Data Structure and Algorithms | |
| **Course Learning Outcomes (CLOs)** | | | | |
| At the end of course the students will be able to: | | | **Domain** | **BT Level\*** |
| 1. Understand key components in the field of Analysis of Algorithms | | | C | 4 |
| 1. Analyze classical algorithms | | | C | 3 |
| 1. Learn advance algorithms used for solving different real word problems. | | | C | 4 |
| \*BT- Bloom’s Taxonomy, C=Cognitive domain, P=Psychomotor domain, A=Affective domain | | | | |

|  |
| --- |
| **Course Contents:** |
| Introduction, importance of analysis with real world application, empirical analysis, analytical analysis, complexity of algorithms/programs. Asymptotic analysis, recurrences and solving recurrences, sorting and searching algorithms. Graphs, greedy and dynamic programing. |
| **Course Objective:** |
| The main objectives of the course are   * To highlight the significance of performance of an algorithm and thus its complexity * To understand the importance of the choice of an appropriate data structure for an optimized overall performance * To learn various algorithm design techniques and their applications on different problems in various domains * To learn to solve new problems using the design and analysis procedures. |
| **Teaching Methodology:** |
| Lectures, Assignments, Presentations and practice. Algorithms will be executed on input examples for better understanding. |
| **Courses Assessment:** |
| Exams, Assignments, Quizzes, Presentations. Course will be assessed with written problem solving examples and examinations. |
| **Reference Materials:** |
| * **Introduction to Algorithms**, 2nd Edition by Thomas H. Cormen  1. **Algorithm Design**, by Jon Kleinberg and Eva Tardos, 2013 (Pearson) 2. Introduction to the Design & Analysis of Algorithms, **by Anany Levitin, Second Edition, 2012 (Pearson)** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Week/Lecture #** | | **Theory** | **Practical** |
| Week 1 | Lecture-I | **Introduction to the Course**  Important Computing Problem Types  Quick Review of Necessary Data Structure Concepts  Analysis of algorithm  Empirical analysis  Analytical analysis |  |
| Lecture-II | **Analysis**  Multiple Solutions for one problem  Design Strategies  Significance of Algorithm Performance and Complexity  Some Simple Examples |  |
| Week 2 | Lecture-I | **Analyzing algorithms**  Sequential algorithms  Conditional (if else and switch) |  |
| Lecture-II | Functions  Loop |  |
| Week 3 | Lecture-I | Nested loops complexity analysis |  |
| Lecture-II | Set of examples with different variation |  |
| Week 4 | Lecture-I | **Algorithm Analysis Framework**  Asymptotic Order of Growth  Asymptotic Order of Growth |  |
| Lecture-II | Standard Notations and Common Functions  Mapping using Limits |  |
| Week 5 | Lecture-I | **Mathematical Analysis of Non- recursive Algorithms**  Mathematical Analysis of Recursive Algorithms  Iterative Algorithm |  |
| Lecture-II | Recurrences and Solutions  Master Theorem |  |
| Week 6 | Lecture-I | **Brute Force Algorithms**  Selection Sort  Exhaustive Search |  |
| Lecture-II | Merge Sort |  |
| Week 7 | Lecture-I | Quick Sort  Count Sort |  |
| Lecture-II | Heap Sort |  |
| Week 8 | Lecture-I | **Graph Algorithms**  Depth First Search (DFS) |  |
| Lecture-II | Breadth First Search (BFS) |  |
| **Midterm Exam** | | | |
| Week 10 | Lecture-I | **Greedy Algorithms**  Element of greedy strategy  Dijkstra |  |
| Lecture-II | Prims  Kruskul |  |
| Week 11 | Lecture-I | **Dynamic Programming**  Binomial Coefficient |  |
| Lecture-II | Fibonannci series |  |
| Week 12 | Lecture-I | Elements of dynamic programming  All Pair shortest distance( Floyed Warshal) |  |
| Lecture-II | Floyed Warshal) |  |
| Week 13 | Lecture-I | Matrix-chain multiplication |  |
| Lecture-II | Matrix-chain multiplication |  |
| Week 14 | Lecture-I | **String Matching**  The naïve string-matching algorithm |  |
| Lecture-II | String matching with finite automata |  |
| Week 15 | Lecture-I | The Rabin-Karp algorithm |  |
| Lecture-II | The Knuth-Morris-Pratt algorithm |  |
| Week 16 | Lecture-I | Huffman coding |  |
| Lecture-II | Tower of hunai |  |
| **Finalterm Exam** | | | |