Marwadi University

Syllabus for Bachelor of Technology

Information and Communication Technology

Subject Code: 01CT0402

Subject Name: Problem Solving Using Python

B. Tech. Year – II (Semester IV)

Objectives:

Obtaining efficient algorithms is very important in modern computer engineering as the world wants applications to be time and space and energy efficient. This course enables to understand and analyze efficient algorithms for various applications.

Credits Earned: 04 Credits

Course Outcomes: After completion of this course, student will be able to:

1. Learn and understand asymptotic notations for performance of different algorithms
2. Derive and solve recurrences describing the performance of divide-and-conquer algorithms
3. Design optimal solution by applying various methods like dynamic Greedy methods.
4. Summarize the certain graph algorithms and their analyses.
5. Apply pattern matching algorithms.
6. Differentiate polynomial and non-polynomial problems.

Pre-requisite of course: Data Structure and proficiency in programming language, knowledge of Mathematical functions like logarithms, graphs etc.

Teaching and Examination Scheme:

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial / Practical Marks		Total Marks
				E	I		V	T	
Theory	Tutorial	Practical		ESE	IA	CSE	Viva	Term Work	
3	0	2	4	50	30	20	25	25	150

Contents:

Unit	Topics	Contact Hours
1	**Introduction to Design and Analysis of Algorithms:** What is an algorithm, Mathematics for Algorithmic Sets, Functions and Relations, Vectors and Matrices, Linear Inequalities and Linear Equations.	02
2	**Analysis of Algorithm and Review of Basic Python:** Review of basic python, data types, Control flow and Statements. The efficient algorithm, Average, Best and worst case analysis, Amortized analysis , Asymptotic Notations(Big Oh, Big Theta, Big Omega), Master Method, Sorting Algorithms and analysis: Bubble sort, Selection sort, Insertion sort, Shell sort, Heap sort, Sorting in linear time : Bucket sort, Radix sort and Counting sort	06
3	**Divide and Conquer:** Introduction, Recurrence and different methods to solve recurrence, Multiplying large Integers Problem, Problem Solving using divide and conquer algorithm - Binary Search, Max-Min problem, Sorting (Merge Sort, Quick Sort), Matrix Multiplication, Exponential.	06
4	**Dynamic Programming:** Introduction, Elements of Dynamic Programming, The Principle of Optimality, Problem Solving using Dynamic Programming – Calculating the Binomial Coefficient, Making Change Problem, Assembly Line-Scheduling, Knapsack problem, Matrix chain multiplication, Longest Common Subsequence.	06
5	**Greedy Algorithm** General Characteristics of greedy algorithms, Elements of greedy strategy, Problem solving using - Activity selection problem, Fractional Knapsack Problem, Job Scheduling Problem.	04
6	**Graph Algorithms** Representation of Undirected & Directed Graph, Traversing Graphs, Depth First Search, Breath First Search, Topological sort, Strongly Connected components. Single pair shortest path and Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm) using greedy approach, All Points Shortest path using Dynamic Programming,	06
7	**Backtracking and Branch and Bound:** Introduction, The Eight queens problem, Knapsack problem, Travelling Salesman problem, Minimax principle.	04
8	**String Matching:** Introduction, The naive string matching algorithm, The Rabin-Karp algorithm, String Matching with finite automata, The Knuth-Morris-Pratt algorithm.	04
9	**Introduction to NP-Completeness:** The class P and NP, Polynomial reduction, 2-CNF Satisfiability, 3- CNF Satisfiability, NP- Completeness Problem, NP-Hard Problems. Travelling Salesman problem, Hamiltonian problem.	04
	Total Hours	42

Suggested Text books / Reference books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, PHI.
2. Learning Python: Powerful Object-Oriented Programming, Mark Lutz, O'Reilly Media
3. Fundamental of Algorithms by Gills Brassard, Paul Bratley, PHI.
4. Introduction to Design and Analysis of Algorithms, Anany Levitin, Pearson.
5. Foundations of Algorithms, Shailesh R Sathe, Penram
6. Programming in Python 3: A Complete Introduction to the Python Language, Mark Summerfield, Addison Wesley
7. Design and Analysis of Algorithms, Dave and Dave, Pearson.

Suggested Theory distribution:

The suggested theory distribution as per Bloom's taxonomy is as per follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process.

Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyze	Evaluate	Create
10%	10%	40%	20%	10%	10%

Suggested List of Experiments:

1. Introduction to Python programming, Creating Hello world Program using Python, Understanding various steps for compiling and Running python codes.
2. Implementation and Time analysis of sorting algorithms
3. Bubble sort, Selection sort, Insertion sort, Merge sort and Quicksort
4. Implementation and Time analysis of linear and binary search algorithm.
5. Implementation of max-heap sort algorithm
6. Implementation and Time analysis of factorial program using iterative and recursive method
7. Implementation of a knapsack problem using dynamic programming.
8. Implementation of chain matrix multiplication using dynamic programming.
9. Implementation of making a change problem using dynamic programming
10. Implementation of a knapsack problem using greedy algorithm
11. Implementation of Graph and Searching (DFS and BFS).
12. Implement prim's algorithm.
13. Implement Kruskal's algorithm.
14. Implement LCS problem.
15. To implement following string matching algorithms and analyze time complexities:
    a. Naïve
    b. Rabin Karp

c. Knuth Morris Pratt

16. Write a program for Floyd-Warshal algorithm.
17. Write a program for travelling salesman problem.
18. 16. Write a program for Hamiltonian cycle problem.
19. 17. To implement Huffman coding and analyze its time complexity.
20. 18. Write a program for Strassen's Matrix Multiplication.

Instructional Method:

1. The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.
2. The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.
3. Practical examination will be conducted at the end of semester for evaluation of performance of students in laboratory.
4. Students will use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory

Supplementary Resources:

1. http://interactivepython.org/runestone/static/pythonds/index.html
2. http://www.personal.kent.edu/~rmuhamma/Algorithms/algorithm.html
3. http://nptel.ac.in/courses/106101060/
4. http://www.comp.nus.edu.sg/~cs5234/Links/Course-Links.htm
5. https://www.coursera.org/learn/algorithm-design-analysis
6. http://www.codeskulptor.org/docs.htmlhttp://www.geeksforgeeks.org
7. http://www.algolist.net
8. http://www.cprogramming.com
9. http://www.codingunit.com