

# GUJARAT TECHNOLOGICAL UNIVERSITY

## COMPUTER ENGINEERING (SOFTWARE ENGINEERING) (02)

COMPUTER ALGORITHM

SUBJECT CODE: 2710201

SEMESTER: I

**Type of course:** Foundation/Core

**Prerequisite:** Data structure and algorithm(primitive, nonprimitive, linear data structure(stack, queue, linked list, nonlinear data structure(tree,graph), hashing, File structure)

**Rationale:** 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.

### Teaching and Examination Scheme:

Teaching Scheme			Credits C	Examination Marks						Total Marks
L	T	P		Theory Marks		Practical Marks				
			ESE (E)	PA (M)	PA (V)		PA (I)			
					ESE	OEP	PA	RP		
3	2#	2	5	70	30	20	10	10	10	150

### Content:

Sr. No.	Topics	Teaching Hrs.	Module Weightage
1	From problems to programs, set theory, functions and relations Insertion sort, analyzing algorithms, designing algorithms, asymptotic notation.	5	10
2	Divide and conquer, Strassen's algorithm for matrix multiplication, The substitution method for solving recurrences, The recursion tree method for solving recurrences, master method	4	10
3	Dynamic programming, Making Change, The principal of optimality, the knapsack problem, Floyd's algorithm for shortest paths	4	10
4	Greedy Algorithms, making change, Knapsack problem, Shortest Path – Dijkstra's algorithm, Huffman codes	5	10
5	Amortized analysis- aggregate analysis, accounting method, potential method	4	10
6	Single source shortest paths. Bellman Ford, directed acyclic graphs, Floyd Warshall algorithm	3	10
7	Number theoretic algorithms, Greatest common divisor, Modular arithmetic	1	5
8	String matching, the naïve string matching, Rabin Karp algorithm, Boyer Moore pattern matching, Knuth Moriss Pratt algorithm	4	15
9	Introduction to NP completeness, The class P and NP, polynomial reductions, NP complete problems	3	10
10	Heuristic algorithm – the travelling salesperson, approximate algorithms-knapsack problem	3	10

## Reference Books:

- 1) Introduction to Algorithms. Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein. PHI publication
- 2) Fundamentals of Algorithms. Gilles Brassard, Paul Bratley. PHI publication
- 3) Advanced data structure. Peter Brass. Cambridge University Press.
- 4) Data structures and Algorithms, Allfred Aho, Jeffrey Ullman, John Hopcroft. Pearson Education.
- 5) Data Structures and Algorithms by G.A.V. Pai, 2009, TMH.
- 6) Fundamentals of Computer Algorithms by Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, 2nd edition, University Press.
- 7) Classic Data Structures by D. Samanta, 2005, PHI
- 8) Design and Analysis of Computer Algorithms by Aho, Hopcraft, Ullman 1998, PEA.
- 9) Introduction to the Design and Analysis of Algorithms by Goodman, Hedetniemi, TMG.
- 10) Design and Analysis of Algorithms by E. Horowitz, S. Sahani, 3rd Edition, Galgotia.
- 11) Data Structures and Algorithms in C++ by Drozdek 2nd Edition, Thomson

## Course Outcome:

After learning the course the students should be able to:

1. Perform best case and worst case analysis of algorithm.
2. Solve recurrence problem.
3. Find optimal solution by applying various methods.
4. Apply pattern matching algorithms to find particular pattern.
5. Differentiate polynomial and nonpolynomial problems.
6. Find shortest path for directed and undirected graph.

## List of Experiments:

1. List the factors that may influence the space complexity of a program. Write a recursive and nonrecursive function to compute  $n!$  Compare the space requirements of nonrecursive function with those of recursive version.
2. The array  $a[0:9]=[4,2,6,7,1,0,9,8,5,3]$  is to be sorted using insertion sort. Show the best case, average case and worst case analysis.
3. Write a program to determine whether or not a character string has an unmatched parenthesis. Use a stack. What is the time complexity of your program? Can we replace the stack with a queue?
4. Write a program that implements change making solution. Assume that the cashier has currency notes in the denominations Rs. 100, Rs. 50, Rs. 20, Rs. 10, Rs. 5 and Rs. 1 in addition to coins . Program should include a method to input the purchase amount and the amount given by the customer as well as method to output the amount of change and a breakdown by denomination. Apply greedy algorithm at the cashier side that is give less number of coins if sufficient currency of that denomination available.
5. Write a program for 0/1 knapsack problem using this heuristic : Pack the knapsack in nonincreasing order of profit density.
6. Write a program that implement divide and conquer method to find the maximum and minimum of  $n$  elements. Use recursion to implement the divide and conquer scheme.
7. Implement the Rabin – Karp matcher and Boyer Moore string matching algorithm. Give analysis – For pattern matching in FIREWALL which algorithm is best suited?
8. Implement Bellman ford algorithm. Find an application that can best be solved by Bellman Ford algorithm.
9. Implement Dijkstra’s algorithm. Find an application that can best be solved by Dijkstra’s algorithm.
10. Consider the subnetting in router. How a source node must be sending a packet to destination node. Does it use greedy, divide and conquer or dynamic programming? Perform analysis.
11. Consider your E-mail account. What data structure can be used to make the search faster if we want subject wise or sender wise search.

### **Open Ended Problems:**

1. An anagram is a type of word play, the result of rearranging the letters of a word or phrase to produce a new word or phrase, using all the original letters exactly once; for example “Doctor Who” can be rearranged into Torchwood.[Reference <http://en.wikipedia.org/wiki/Anagram>]. Given a file with 1,00,000 words , design a data structure to find anagrams.

There are 100000 lines in a file. Out of these only 2 lines are identical. Rests are unique. Each line is so long that it cannot fit in memory. Find an optimal solution to find identical lines.

2. Usually routing protocols use fixed number of fields. Design a routing protocol with variable data structure.

### **Major Equipments:**

Desktop,laptop

### **List of Open Source Software/learning website:**

- 1) [www.nptel.ac.in](http://www.nptel.ac.in)
- 2) <http://www.cise.ufl.edu/~sahni/cop3530/>