AMITY SCHOOL OF ENGINEERING & TECHNOLOGY
Algorithms Design and Analysis (ETCS-301)

Semester – V Credit - 4

ETCS 301 – ALGORITHMS DESIGN AND ANALYSIS

Lecture Schedule Total lecture classes available: 36

	TOPICS TO BE COVERED	TOTAL NUMBER OF LECTURE	TOTAL NUMBER OF TUTORIAL
1.	UNIT I - Asymptotic notations for time and space complexity, Big-Oh notation, Θ notation, Ω notation, the little-oh notation, the little-omega notation	2	1
2.	Recurrence relations: iteration method, recursion tree method, substitution method, master method (with proof)	2	
3.	Subtract and conquer master method (with proof), Data Structures for Disjoint Sets, Medians and Order statistics	2	2
4.	Complexity analysis, Insertion sort, Merge Sort, Quick sort, Strassen's algorithm for Matrix Multiplications	3	
5.	UNIT II - Dynamic Programming: Ingredients of Dynamic Programming, emphasis on optimal substructure , overlapping substructures,	2	3
6.	Memorization, Matrix Chain Multiplication Longest common subsequence, Optimal binary search trees problems	4	
7.	0-1 knapsack problem, Binomial coefficient computation through dynamic programming.	2	
8.	Floyd Warshall algorithm	1	
	END Mid TERM		
9.	UNIT III - Greedy Algorithms: Elements of Greedy strategy, overview of local and global optima, matroid	2	
10.	An activity selection problem, Fractional Knapsack problem, Huffman Codes, A task scheduling problem.	3	1
11.	Minimum Spanning Trees: Kruskal's & Prim's Algorithms	2	1
12.	Single source shortest paths: Dijkstra's & Bellman Ford Algorithm (with proof of correctness of algorithms).	2	
13.	UNIT IV - String matching: The naïve String Matching algorithm, Rabin-Karp Algorithm	2	1
14.	String Matching with finite automata, Knuth-Morris Pratt algorithm	2	
15.	NP-Complete Problem: Polynomial time verification, NP-Completeness & Reducibility	2	2
16.	NP-Completeness Proof, NP –hard ,Case study of NP-Complete, problems (vertex cover problem, clique problem)	3	

Text Books:

[T1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Ed., PHI, 2013.
[T2] Jon Klenberg,Eva Tardos,"Algorithm Design", Pearson Publications, 2014

Reference Books:

[R1] Sara Basse, "introduction to Design & analysis",Pearson
[R2] Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Computer Algorithms/C++ "Second Edition, Universities Press.
[R3] A. V. Aho, J. E. Hopcroft, J. D. Ullman, "The Design and Analysis of Computer Algorithms", Pearson Publication, 2013.
[R4] Richard Neapolitan, "Foundations of Algorithms" , Fifth Edition, Jones & Bartlett Learning.

Dr. A K Yadav

ALGORITHMS DESIGN AND ANALYSIS

Paper Code: ETCS-301	**L**	**T/P**	**C**
Paper: Algorithms Design and Analysis	**3**	**1**	**4**

INSTRUCTIONS TO PAPER SETTERS: **MAXIMUM MARKS: 75**

1. Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 25 marks.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be of 12.5 marks

***Objective**: The objective of this paper is to teach the students various problem solving strategies like divide and conquer, Greedy method, Dynamic programming and also the mathematical background for various algorithms. After doing this course, students will be able to select an appropriate problem solving strategies for real world problems. This will also help them to calculate the time, complexity and space complexity of various algorithms.*

UNIT – I

Asymptotic notations for time and space complexity, Big-Oh notation, Θ notation, Ω notation, the little-oh notation, the little-omega notation, Recurrence relations: iteration method, recursion tree method, substitution method, master method (with proof), subtract and conquer master method(with proof), Data Structures for Disjoint Sets, Medians and Order statistics. Complexity analysis, Insertion sort, Merge Sort, Quick sort. Strassen's algorithm for Matrix Multiplications.

[T1][R1][R2][No. of Hrs. 10]

UNIT – II

Dynamic Programming: Ingredients of Dynamic Programming, emphasis on optimal substructure , overlapping substructures, memorization. Matrix Chain Multiplication, Longest common subsequence and optimal binary search trees problems, 0-1 knapsack problem, Binomial coefficient computation through dynamic programming. Floyd Warshall algorithm.

[T1][T2][R1] [R3][No. of Hrs. 10]

UNIT – III

Greedy Algorithms: Elements of Greedy strategy, overview of local and global optima, matroid, Activity selection problem, Fractional Knapsack problem, Huffman Codes, A task scheduling problem. Minimum **Spanning Trees:** Kruskal's and Prim's Algorithm, Single source shortest path: Dijkstra's and Bellman Ford Algorithm(with proof of correctness of algorithms).

[T1][T2][R4] [No. of Hrs. 10]

UNIT – IV

String matching: The naïve String Matching algorithm, The Rabin-Karp Algorithm, String Matching with finite automata, The Knuth-Morris Pratt algorithm.

NP-Complete Problem: Polynomial-time verification, NP-Completeness and Reducibility, NP-Completeness Proof, NP –hard ,Case study of NP-Complete problems (vertex cover problem, clique problem).

[T1][R1] [No. of Hrs.: 10]

Text Books:

[T1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Ed., PHI, 2013.

[T2] Jon Klenberg,Eva Tardos,"Algorithm Design", Pearson Publications,2014

Reference Books:

[R1] Sara Basse, "introduction to Design & analysis",Pearson

[R2] Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Computer Algorithms/C++ "Second Edition, Universities Press.

[R3] A. V. Aho, J. E. Hopcroft, J. D. Ullman, "The Design and Analysis of Computer Algorithms", Pearson Publication, 2013.

[R4] Richard Neapolitan, "Foundations of Algorithms" , Fifth Edition, Jones & Bartlett Learning

AMITY SCHOOL OF ENGINEERING & TECHNOLOGY
Algorithms Design and Analysis (ETCS-301)

Semester – V Credit - 4

Tutorial – 1

1. Why do we study algorithms? What is the difference between algorithm and program?
2. What is smallest value of n such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n on the same machine?
3. Analyze the problem of arranging the students of 3rd year at ASET according to their height.
4. What are different strategies you know of solving these problems?
5. How much time would each strategy take if each comparison takes 15 seconds and 30 seconds to interchange the position? How much land area would your each strategy require if each student occupies 0.2 m^2?
6. Design an algorithm for computing $\lfloor \sqrt{n} \rfloor$ for any positive integer n. Besides assignment and comparison. Your algorithm may only use the four basic arithmetical operations.
7. Explain the time & space complexity of an algorithm.
8. Describe linear and non- linear data structure with example.

Tutorial – 2

1. Distinguish between the following:-
   a. Big O notation and Θ notation
   b. Big Ω notation and Θ notation
2. Let f(n) and g(n) be asymptotically non negative functions. Using the basic definitions of Θ - notation, prove that max (f (n), g (n)) = Θ (f (n) +g (n)).
3. Prove that for any two functions f(n) and g(n), f(n)= Θ(g(n)) iff f(n) = O(g(n)) and f(n) = Ω(g(n)).
4. Find functions f(n) and g(n) that satisfy, f(n) = Ω(g(n)) and f(n) $\neq$ O(g(n)).
5. Prove that $2^n = O(n!)$ and $n^2 = O(2^n)$. Is $3n+2=O(n^2)$ true?
6. Show that for any real constants a & b, where b>0 $(n+a)^b = \Theta(n^{b)}$
7. Prove that running time of an algorithm is Θ(g(n)), if and only if its worst case running time is O(g(n)) and its best case running time is Ω(g(n)).
8. Prove that (a) log(n!) = O(nlogn), and
   (b) $(n+a)^b = O(n^b)$

Tutorial –3

1. How disjoint sets are defined using linked representation? Explain make_set, union & find operation and give their time complexity.
2. Prove that the solution of :
   (a) T(n)=T(n/2)+1 is O(lg n)
   (b) T(n)= 2T(n/2)+n is O(nlg n)

Dr. A K Yadav

AMITY SCHOOL OF ENGINEERING & TECHNOLOGY
Algorithms Design and Analysis (ETCS-301)

Semester – V Credit - 4

(c) $T(n) = 3T(n/4)+n\log n$ using master theorem

3. Solve the following recurrences:
   a. $T(n) = 2T(n/4)+n$
   b. $T(n) = 4T(n/2)+n^2$
   c. $T(n) = 2T(n/2)+n\log n$
4. Determine a good asymptotic upper bound on the recurrence $T(n) = T(n/3) + T(2n/3) + n$ by applying a recursion tree.
5. If $f(n)=a_0+ a_1n+ \ldots a_{m-1}n^{m-1}+ a_mn^m$ then prove that $f(n)=O(n^m)$.
6. Determining the asymptotic upper bound on the recurrence $T(n) = 2T(\sqrt{n})+\log n$.
7. Give a Θ bound solution for the following recurrences:- a) $T(n)=2T(n/3)+1$; b) $T(n)=7T(n/7)+n$; c) $T(n)=9T(n/3)+ n^2$
8. Show how to remove all duplicates from the array in time $O(n\log n)$

Tutorial –4

1. Find an optimal parenthesis of a matrix chain product whose sequence of dimension is 2, 3,4,5,6.
2. Find the LCS from the given two sequence of characters:
   P=(a,a,b,c,d,e,f) Q=(a,b,c,d,e,f)
3. Find the m^{th} order statistics using minimum time & find out the complexity of that algorithm also.
4. Use recursion tree and master theorem to find best case and worst case running time for quick sort.
5. Why the complexity of randomized Quick sort is less then the complexity of normal Quick Sort..
6. How would you construct an optimal binary search tree for a set of n keys if all the keys are equally likely to be searched for? What will be the average number of comparisons in a successful search in such a tree of $n = 2^k$
7. What does dynamic programming have in common with divide & conquer? What is the principal difference between the two techniques?
8. Find the multiplication of two matrix using Strassen's algorithm

Tutorial –5

1. What are the four steps of dynamic programming?
2. Write LCS algorithm and do analysis of its computational complexity.
3. Illustrate the operation of HEAP SORT on the array A = <5,13,2,25,7,17,20,8,4>
4. Suppose the dimensions of matrices A, B, C, D are 20x2, 2x15, 15x40, and 40x4 respectively. What will be the number of scalar multiplications?
5. What is the minimum and maximum number of elements in a heap of height h?
6. Find an optimal parenthesization of a matrix chain multiplication whose sequence of dimensions are 2, 6,3,5,4.

AMITY SCHOOL OF ENGINEERING & TECHNOLOGY
Algorithms Design and Analysis (ETCS-301)

Semester – V Credit - 4

7. Show that the running time of Quick Sort is $\Theta(n^2)$ when the array A contains distinct elements and is sorted in decreasing order.
8. Determine the cost & structure of an OBST for a set of n=7 keys with the following probabilities:-

i	0	1	2	3	4	5	6	7
p_i		0.06	0.04	0.08	0.02	0.1	0.12	0.14
q_i	0.05	0.04	0.18	0.05	0.04	0.06	0.07	0.04

Tutorial –6

1. Distinguish between dynamic programming and greedy approach of solving problem with the help of suitable example.
2. Given ten activities along with their start and finish time as $S=(A_1,A_2,A_3,A_4,A_5,A_6,A_7,A_8,A_9,A_{10})$,$F_i=(3,5,4,7,10,9,11,13,12,14)$ $S_i=(1,2,3,4,7,8,9,9,11.12)$. Compute a schedule where the largest numbers of activities take place.
3. What is the optimal Huffman Code for the following set of frequencies a:20, b:30, c:30, d:5, e:5, f:10.
4. Explain following terms: Matroids, Memoization
5. Find optimal scheduling and total incurred penalty for following given problem: $a_i=(1,2,3,4,5,6,7)$ $d_i=(4,2,4,3,1,4,6)$ $w_i=(70,60,50,40,30,20,10)$.
6. How does dynamic programming differ from memoization?
7. Find out the Huffman Code for the five-character alphabet {a,b,c,d,e} with the following occurrence probabilities-a:0.35, b:0.1, c:0.2, d:0.2, e:0.15.
8. Find the optimal schedule for the following problem- $a_i=(1,2,3,4,5,6,7)$, $d_i=(1,3,3,4,1,2,1)$, $p_i=(3,5,18,20,6,1,38)$

Tutorial –7

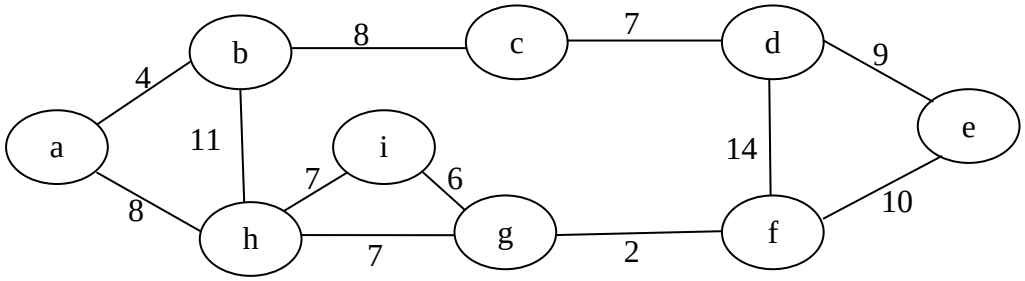
1. How one can prove correctness of Huffman Code Algorithm?
2. There are 5 jobs whose profits $(p_1...p_5)=(20,15,10,1,18)$ and deadlines $(d_1...d_5)=(2,2,1,3,3)$. Find the optimal solution that maximizes profits on these scheduling these jobs.
3. What is Knapsack problem? Explain different types of it with example.
4. Consider five items along their respective weight and values $I=(I_1, I_2, I_3, I_4, I_5)$, w=(5,10,20,30,40), v=(40,20,120,90,160). The capacity of knapsack W=60. Find the solution of the problem with fractional weight and without fractional weight.

AMITY SCHOOL OF ENGINEERING & TECHNOLOGY
Algorithms Design and Analysis (ETCS-301)

Semester – V Credit - 4

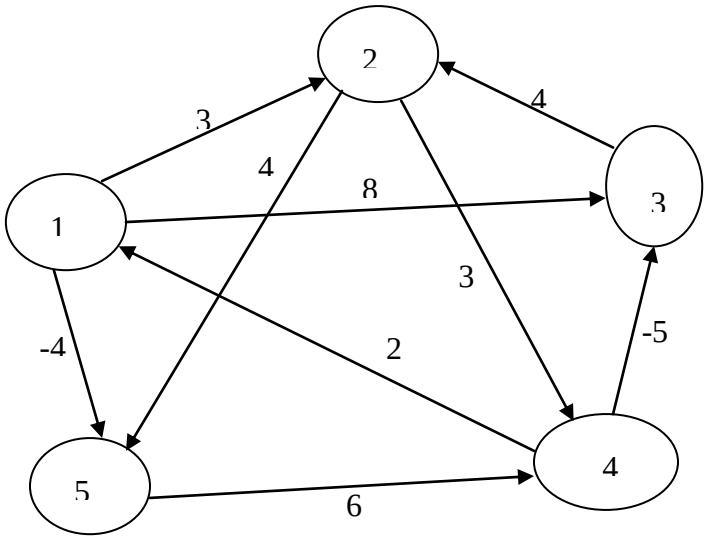
Tutorial –8

1. What do you mean by MST?
2. Write and explain Prim's Algorithm for finding MST.
3. Distinguish between Kruskal and Prim's algorithms with the help of following graph



Graph: G3

4. Explain Bellman Ford algorithm for solving single source shortest path problem.
5. Run the Bellman Ford algorithm on the below given directed graph.



6. Find the all pair shortest path using Floyd Warshall algorithms for the given graph. (use above graph with all positive weights)

AMITY SCHOOL OF ENGINEERING & TECHNOLOGY
Algorithms Design and Analysis (ETCS-301)

Semester – V Credit - 4

Tutorial –9

1. Prove that the running time of Floyd Warshall is $\Theta(V^3)$.
2. How can the output of Floyd Warshall algorithm be used to detect the presence of a negative weight cycle?
3. Write an algorithm to compute transitive closure of G.
4. What is the space complexity for the Floyd Warshall algorithm?
5. What is the time complexity of Dijkstra's algorithm?
6. Prove that Dijkstra's algorithm is correct.
7. Let (u,v) be a minimum-weight edge in a graph G. Show that (u,v) belongs to some minimum spanning tree of G.
8. Write an efficient algorithm to count the total number of paths in a directed acyclic graph & analyze your algorithm.

Tutorial –10

1. Give algorithm and explain principle for finite automation string matching technique.
2. Discuss Rabin-Karp algorithm for following given string text T = 3141592653589793 and pattern P = 26.
3. Show the comparisons the naïve string matcher makes for the pattern P=(0 0 0 1) in the text T=(0 0 0 0 1 0 0 0 1 0 1 0 0 0 1).
4. Construct the string matching automation for the pattern p=(a a b a b) and illustrate its operation on the text string T=(a a a b a b a a b a a b a a b)
5. Using knuth Marries Pratt algorithm find whether the pattern P=(0 0 1 0) is in the text T=(1 1 0 0 0 1 1 0 1 0 0 0 1 0 1 0) or not.
6. What are NP Complete problems? Define NP completeness and reducibility of problems.
7. Give the Knuth-Morris-Pratt algorithms for pattern matching. Discuss its performance and failure function.
8. Describe Hamiltonian Cycle in NP.

Dr. A K Yadav