DESIGN AND ANALYSIS OF ALGORITHMS

COURSE OBJECTIVE:

The objective of this course is to study paradigms and approaches used to analyze and design algorithms and to appreciate the impact of algorithm design in practice. It also ensures that students understand how the worst-case time complexity of an algorithm is defined, how asymptotic notation is used to provide a rough classification of algorithms, how a number of algorithms for fundamental problems in computer science and engineering work and compare with one another, and how there are still some problems for which it is unknown whether there exist efficient algorithms, and how to design efficient algorithms.

JNTU SYLLABUS :

UNIT – I
Introduction, algorithm, pseudo code for expressing algorithms, performance analysis, space complexity, time complexity, asymptotic notation, big oh notation, omega notation, theta notation, and little oh notation, probabilistic analysis, amortized analysis.

UNIT-II
Disjoint sets, disjoint set operations, union and find algorithms, spanning trees, connected components, biconnected components

UNIT – III
Divide and conquer: general method, applications, binary search, quick sort, merge sort, strassen's matrix multiplication.

UNIT – IV
Greedy method: general method, applications, job sequencing with deadlines, 0/1 knapsack problem, minimum cost spanning trees, single source shortest path problem.

UNIT – V
Dynamic programming: general method, applications, matrix chain multiplication, optimal binary search trees, 0/1 knapsack problem, all pairs shortest path problem, travelling sales person problem, reliability design.

UNIT – VI
Backtracking: general method, applications, n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

UNIT – VII
Branch and Bound: general method, applications, travelling sales person problem, 0/1 knapsack problem. LC branch and bound solution, FIFO branch and bound solution.

UNIT – VIII
NP-hard and NP-complete problems: basic concepts, non deterministic algorithms, NP-hard and NP-complete classes, Cook's theorem.

SUGGESTED BOOKS

TEXT BOOKS:

T1. Ellis Horowitz, Satraj Sahni and S Rajasekharam, Fundamentals of Computer Algorithms, Galgotia publishers

T2. Parag Himanshu Dave, Himanshu BhalchandraDave, Design and Analysis algorithms Pearson Publication

T3. M.T. Goodrich, Robert Tamassia, Algorithm design: Foundations, Analysis and Internet examples, Wiley student Edn, John Wiley & sons

REFERENCE BOOKS:

R1. T H Cormen, C E Leiserson, and R L Rivest, Introduction to Algorithms, 2nd Edn, Pearson Education
R2. R C T Lee, Hang and TT Sai, Introduction to Design and Analysis of Algorithms, A strategic approach, TMH
R3. Allen Weiss, Data structures and Algorithm Analysis in C++, 2nd Edn, Pearson Education
R4. Aho, Ullman and Hopcroft, Design and Analysis of Algorithms, Pearson Education
R5. Richard Johnson Baugh, and Marcus Schaefer, Algorithms, Pearson Education

Session Plan

Topic	Lecture No	Topics to be covered	Text book/reference books
UNIT-I			
Understanding what is an algorithm	1	Introduction to Algorithm, characteristics	TB1: 1.1
Understanding algorithm specifications	2	Pseudo conventions	TB1: 1.2.1
Performance analysis of an algorithm	3 , 4,	Time complexity, space complexity, step count method	TB1: 1.3.1,1.3.2 & R1:4.2
Performance analysis of an algorithm	5,6	Asymptotic notation	TB1: 1.3.3 &R1:3.1-3.5
Analysis of algorithms	7, 8	Probabilitic analysis, Amortized analysis	R1: 5.4, 17.1 – 17.4 & R1:4.6
Tutorial	9	Problems on algorithms	

UNIT-II			
Disjoint sets, disjoint set operations	10	Disjoint sets, disjoint set operations	TB1:2.5.1
Union and find for sets	11,12	Weighted union and find algorithms	TB1:2.5.2
Spanning trees	13	spanning trees	TB1:4.5
Connected components	14	Biconnectedcomponents	TB1:6.4
Tutorial	15	Problems	
UNIT-III			
Understanding divide-and-conquer method	16	Divide-and-conquer general method	TB1: 3.1
Applications of Divide-and-conquer	17	Binary search	TB1: 3.2
Applications of Divide-and-conquer	18,19	Quick sort	TB1: 3.5 & R2:3.5
Applications of Divide-and-conquer	20,21	Merge sort	TB1: 3.4
Applications of Divide-and-conquer	22	Strasen's Matrix multiplication	TB1:3.7 & R2:6.2

Tutorial	23	Tutorial	

UNIT-IV			
Understanding greedy method	24	greedy general method	TB1: 4.1
Applications of greedy method	25	Job sequencing with deadlines	TB1: 4.4 & R1:6.5
Applications of greedy method	26	Knapsack problem	TB1:4.2
understanding spanning trees	27	Minimum cost spanning trees, Prim's algorithm,	TB1: 4.5 , 4.5.1 & R1:6.3.1-6.3.2
spanning trees	28	Kruskal's algorithm	TB1:4.5.2
shortest path problem	29	Single source shortest path	TB1: 4.8
	30	Tutorial	
UNIT-V			
Understanding dynamic programming	31	General method	TB1: 5.1
Applications of dynamic programming	32	Matrix chain multiplication	TB2:15.6
Applications of dynamic programming	33,34	Optimal binary search trees	TB1: 5.5, & R2 : 4.5
Applications of dynamic programming	35	Knapsack problem	TB1:5.7
Applications of dynamic programming	36	All pair shortest path	TB1:5.3
Applications of dynamic programming	37	Travelling salesman problem	TB1:5.9
Applications of dynamic programming	38,39	Reliability design	TB1:5.8
	40	Tutorial	
	41	Revision	
UNIT-VI			
Understanding backtracking	42	General method	TB1: 7.1 & R1:9.6
Applications of Backtracking	43	8-Queens problem,	TB1: 7.2 & R1:9.6.2
Applications of Backtracking	44	sum of subsets,	TB1: 7.3
Applications of Backtracking	45	Hamiltonian cycles	TB1: 7.5
Applications of Backtracking	46	graph coloring	TB1: 7.4
	47	Tutorial	
	48	Revision	
UNIT-VII			
Branch and bound	49	General method	TB1: 8.1
Applications of branch and bound	50	Knapsack problem	TB1: 8.2
Applications of branch and bound	51,52	LC branch and bound solution,	TB1: 8.2.1

Applications of branch and bound	53	FIFO branch and bound solution	TB1: 8.2.2
Applications of branch and bound	54,55	Traveling salesperson problem	TB1: 8.3
	56	Tutorial	
	57	Revision	
UNIT-VIII			
Understanding NP hard and NP complete problems	58,59	Basic concepts, nondeterministic algorithms	TB1: 11.1, 11.1.1 & R1:12.5
Applications of NP hard problems	60	NP hard class	TB1: 11.1.2
Applications of NP hard problems	61	NP complete class	TB1: 11.1.2
Applications of NP hard problems	62,63	Cook's theorem	TB1: 11.2
	64	Tutorial	
	65	Revision	

Books referred by faculty:

1) Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and S Rajasekharan, Galgotia publishers
2) Design and Analysis of Algorithms, Aho, Ullman and Hopcroft, Pearson Education

Websites:

http://www.ki.inf.tu-dresden.de/~hans/www-adr/alg_course.html --String Matching, Sorting, Linear Programming

http://www.algorithmist.com/index.php/Main_Page-it contains dynamic programming, greedy, ,Graph Theory, sorting, Data Structures

http://www-2.cs.cmu.edu/~guyb/realworld.html- it contains Data Compression, Indexing and Search engines, linear Programming, Pattern matching

For lecture notes
http://www.cs.princeton.edu/~wayne/cs423/lectures.html --it contains Beyond worst case complexity, Dynamic programming, Reductions, Greedy algorithms, Data Structures

http://www.personal.kent.edu/~rmuhamma/Algorithms/algorithm.html -- it contains Binary search trees, Amortized Anaysis, Graph Algorithms, Greedy Algorithms, Hash Table

JOURNALS:

1) Journal of Graph Algorithms and Applications

url:http://www.emis.de/journals/JGAA/home.html

2) **The Journal of Discrete Algorithms (JDA)**
Aims to facilitate communication between researchers in theoretical computer science who are concerned with the design and analysis of discrete algorithms. The Journal of Discrete Algorithms welcomes original papers in all areas of finite and discrete algorithm design

url:http://www.elsevier.com/wps/find/journaldescription.cws_home/672711/description#description

3)Algorithmica –
A journal about the design of algorithms in many applied and fundamental areas
http://www.springerlink.com

Student seminar topics:

1. An Efficient Dynamic Algorithm for Maintaining All-Pairs Shortest Paths in Stochastic Networks,
   S Misra and B J Oommen, IEEE Transactions on Computers, vol 55, no 6, Jun 06.

2. Risk-Resilient Heuristics and Genetic Algorithms for Security Assured Grid Job Scheduling,
   S Song, K Hwang, and Y K Kwok, IEEE Transactions on Computers, vol 55, no 6, Jun 06.

3. The science of breeding and its application to the breeder genetic algorithm (BGA), H Muhlenbein and D Schlierkamp, Evolutionary Computing, vol 1, no 4, pp 335-360, 94

4. The Random Walk Construction of Uniform Spanning Trees and Uniform Labelled Trees

5. Using intelligent backtracking to improve branch-and-bound methods : an application to open-shop problems.

6. An Efficient Branch-and-bound Algorithm for Finding a Maximum Clique with Computational Experiments," Journal of Global Optimization, vol. 37, pp. 95-112, 2007;

7. On the crossing numbers of Cartesian products with paths. A random NP-complete problem for inversion of 2D cellular automata

8. Fast Numerical Algorithms

Guest Lecture Topics:

1. Introducing formal derivation into the design and analysis of algorithms.
2. Communication Issues in the Design and Analysis of Parallel Algorithms
3. Matrix methods for the design and analysis of recurrent algorithms.
4. Design and Analysis of Irregular Algorithms on the Cell Processor

Assignment Questions:

UNIT –1

1. Define algorithm. Explain the characteristics of the algorithm
2. What are best case, average case, and worst-case performance? Explain
3. Compute time complexity of recursive Fibonancci procedures where F(n)= F(n-1) + F(n-2)
4. Explain how time complexity of an algorithm is computed
5. Compare the 2 functions N2 and 2n/4 for various values of n?
6. analyse the computing time of selection sort.?
7. What do you mean by performance analysis of an algorithm? Explain
8. Write an algorithm for matrix multiplication and find step count to calculate complexity
9. Write the algorithm for addition and obtain run times for n=1,10,20,30.
10. Write the non recursive algorithm for finding the Fibonacci sequence and derive its time complexity.
11. a) Devise an algorithm, which accepts a number in decimal and produces the equivalent number in binary. b) What is its time complexity?
12. Compare the two functions n2 and 2n/4 for various values of n. Determine when the second becomes larger than first?
13. (a)Show that f(n)+g(n)=O (n2) where f(n) = 3n2− n + 4 and g(n)=nlogn+5
(b)Explain how time complexity of an algorithm is computed.
14. Consider the following claims
i. O(n+km) = O(nm), where k and m are constants
ii. 2n+1 = O(2n)
iii. 22n+1 = O(2n)
Which of the following claims are correct
i. I and II ii. I and III iii. II and III iv. I, II and
15. Determine the frequency counts for all statements in the following two algorithm segments

i.
```
for I:=1 to n do
for j:=1 to I do
for k:=1 to j do
x:=x+1;
```

ii.
```
I:=1;
While (I_ n) do
{
 x:=x+1;
 I:=I+1;
}
```

UNIT-II

1. Write the algorithm for simple union?
2. Write the algorithm for simple find?
3. Write the algorithm for weighted union?
4. Write the algorithm for collapsing find?
5. Write the prims algorithm? And find its time complexity?
6. Write the kruskal's algorithm? and find its time complexity?
7. Draw the tree representation of two set union?
8. Draw the data representation for sets?
9. Explain array representation for sets?
10. Draw the tree for weighted union?
11. Draw the tree for collapsing find?
12. Draw the tree for achieving for worst case bound weighted union?
13. Write the algorithm for biconnected components?
14. Show that if u unions are performed then no set contains more than u+1 elements?
15. Show that if u unions are performed then at leastmax{n-2u,0}single ton sets remain?

UNIT –III

1. Run the recursive and iterative versions of binary search and compare times.

2. Write and explain the control abstraction for Divide and conquer

3. Prove by induction the relationship E=i+2n where E and I are external and internal path length respectively.

4. Write an algorithm of quick sort and explain in detail.

5. Trace the quick sort algorithm to sort the list C, O, L, L, E, G, E in alphabetical order

6. Explain the Strassen's matrix multiplication concept with an example

7. The worst-case time of procedure MERGESORT is O(nlogn). What is its time in the best case? Can we say that the time for merge sort is O(nlogn)?

8. What is a STABLE SORTING method? Is merge sort a stable sorting method?
9. Calculate the time complexity of merge sort and quick sort
10. Show how quick sort sorts the following sequence of keys 1,1,1,1,1,1,1 and 5,5,8,3,4,3,2.
11. The worst-case time of procedure MergeSort is O(nlogn).what is its best-case?

12. Why is it necessary to have the auxiliary array b[low:high] in function Merge? Give an example that shows why in-place merging is inefficient?

13. (a) Suppose a binary tree has leaves ℓ1ℓ2.....lm at depths d1, d2....dm respectively prove that mPi=12−di _ 1 and determine when the equality is true.
(b) Write and explain the control abstraction algorithm of divide and conquer

14. Suggest refinements to merge sort to make it in-place.

15. Show how quick sort sorts the following sequences of keys in ascending order. 22,55,33,11,99,77,55,66,54,21,32

UNIT-IV

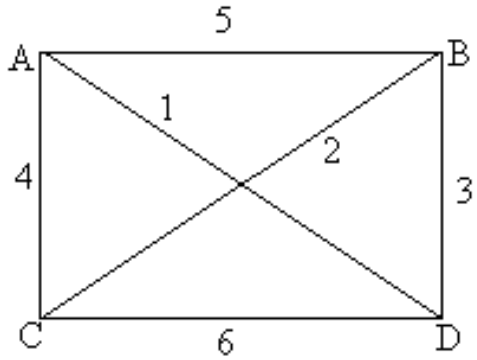
1. Explain the Kruskal's algorithm with an example
2. Explain the control abstraction of greedy method.
3. Explain the 0/1 knapsack problem algorithm with greedy concept.
4. Applying the greedy strategy find the solution for optimal storage on tapes problem instance n = 3, (l1, l2, l3) = (5, 10, 3).
5. Prove that Kruskal's algorithm generates a minimum-cost spanning tree for every connected undirected graph G.
6. Explain the algorithm for Job sequencing with deadlines. Applying the same, find the solution for the instance n = 4, (p1.....p4)=(100,10,15,27) and (d1.....d4)=(2,1,2,1).
7. Give an algorithm to assign program to tapes.
8. What are depth first spanning trees? Explain.
9. Write a Greedy algorithm to generate shortest path.
10. Explain the Prims' algorithm with an example. Analyze the time complexity of the algorithm
11. Formulate an algorithm for a spanning tree problem in terms of a sequence of set operations in which take G as the undirected graph; S as the undirected tree; V as the number of vortices; E as the number of edges; T as a set used to collect the edges of the final minimum spanning tree; C as the cost function for the graph G given by C (e) for the sub graph $G^1=(V^1,E^1)$ of G. Use set VS for the vertex set of the trees in the spanning forest to write the minimum cost spanning tree algorithm.

12. Consider the Knapsack instance n=6, m=165, (p1, p2, ..., p6) = (w1,w2, ...,w6) = (100,50,20,10,7,3). Generate the Si sets containing the pair (pi,wi) and thus find the optimal solution.
13. Prove that Kruskal's algorithm generates a minimum-cost spanning tree for every connected undirected graph G.
14. What are breadth first spanning trees? Explain.
15. Write the algorithm for optimal storage on tapes.

UNIT-V

1. Find one problem for which the principle of optimality does not hold. Explain why the principle does not hold.
2. Write an algorithm of matrix chain multiplication.
3. Design a three stage system with device types D1,D2 and D3. The costs are Rs.30, Rs.15 and Rs.20 respectively. The cost of the system is to be no more than Rs.105. the reliability of each device type is 0.9, 0.8 and 0.5 respectively.
4. Explain in detail the reliability design problem.
5. Define a binary search tree.
6. What do you mean by forward and backward approach of problem solving in Dynamic Programming
7 What are the differences between Greedy and dynamic programming method of problem solving techniques?
8. Show that the computing time of algorithm OBST is O(n2).
9. Write an algorithm to construct the optimal binary search tree T given the roots R(i, j), 0 d" I d" J d'
10. Using a dynamic programming approach coupled with the set generation approach, show how to obtain an 0(2n/2) algorithm for the 0/1 knapsack problem.

11. Find the shortest path between all pairs of nodes in the following graph.



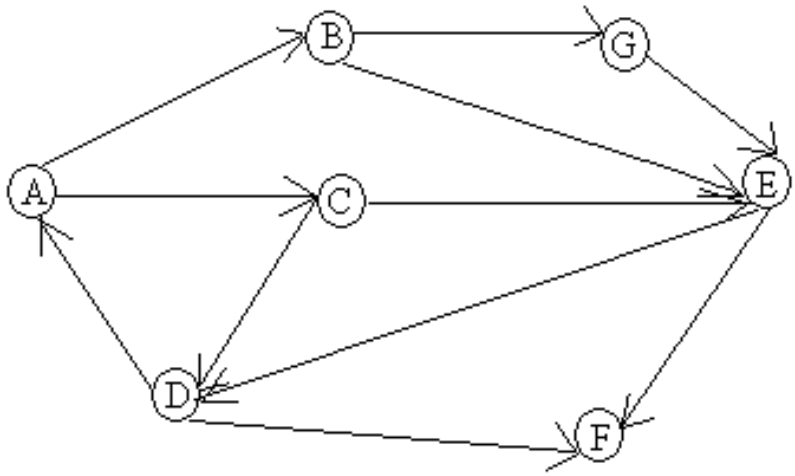
12. Show that the computing time of algorithm OBST is O(n2).
13. For the Travelling sales person algorithm show that the (time complexity is 0(n22n) and space complexity is O(n2n).

14. What are the differences between Greedy and dynamic programming method of problem solving techniques?

15. Prove that algorithm for construction of an optimal binary search tree requires O(n3) time.

UNIT-VI

1. Explain the BFS algorithm with an example
2. Write an algorithm of Biconnected components and also analyze its time complexity .
3. Write a detail note on depth first graph traversal algorithm
4. (a) Find the strongly connected components in the graph of below figure.



(b) Write a non-recursive algorithm of Pre-order traversal of a tree and also analyze its time complexity

5. Write an algorithm of Biconnected components and also analyze its time Complexity

6. a) Write a recursive backtracking algorithm for sum of subsets problem.
   b) Draw the search tree to color the graph with the three colors: red, blue, green

7. (a) Write an algorithm of finding all m-colorings of a graph.
   (b) Describe the 4-queens problem using backtracking.

8. (a) Let w = {15, 7, 20, 5, 18, 10, 12} and m=35. Find all possible subsets of w that sum to m. Draw the portion of the state space tree that is generated.
   (b) Write the control abstraction of backtracking.

9. (a) Write an algorithm to generate next color in M coloring problem.
   (b) Draw the tree organization of the 4-queen solution space. Number the nodes using DFS.

10. Discuss the relevance of backtracking technique to m-coloring graph. Explain with an example.

11. What is graph coloring? Present an algorithm, which finds m-coloring of a graph

12. Draw the state space for m-closing graph using an suable graph
13. Suggest a solution for 8 queen's problem.
14. Compare and contrast Bruteforce approach Vs backtracking.
15. prove that the size of the set of all subsets of n elements is 2n.

UNIT-VII

1. (a) Explain the general method of Branch and Bound.
   (b) Explain the principles of LIFO Branch and Bound
2. Draw the portion of the state space tree generated by FIFO Branch and Bound for the following instances.
   n=5, m=12, (P1 – – – P5) = (10,15,6,8,4)
   (w1 – – – –w5) = (4,6,3,4,2)
3. Draw the portion of the statespace tree generated by LCBB for the following instances.
   n=4, m=15, (P1 – – – –P4)=(10, 10, 12,18)
   (w1 – – – w4)=(2,4,6,9).

4. (a) What is Bounding? Explain how these bound are useful in Branch and Bound methods.
   (b) Describe the TSP in Branch and Bound.

5. Write a complete LC - branch and bound algorithm for knapsack problem.
6. Explain Traveling sales person problem with an example.
7. Present a program schema for a FIFO branch and bound search for a least–cost answer node
8. Write a program schema for a LIFO branch and bound search for a least–cost answer node.
9. Using a state space tree formulation and cost function c, obtain the portion of the state space tree that will be generated by LCBB. Label each node by its value. Write out the reduced matrices corresponding to each of these nodes.

10. Give the algorithm for FIFOBB for Knapsack problem.
11. Write a backtracking algorithm using the static tree formulation.
12. Draw the portion of a state space tree generated by FIFOBB, LCBB and LIFOBB for the job sequencing with deadlines instance n=5, (p1, p2, ., p5)=(6,3,4,8,5), (t1, t2, .., t5)=(2,1,2,1,1) and, (d1, d2, .., d5) = (3,1,4,2,4). What is the penalty corresponding to an optimal solution? Use a variable tuple size formulation and c (.) and u (.)
13. Explain the solution to the Traveling sales person problem using LCBB.
14. Is the above technique applicable for a non-symmetric distance matrix? Substantiate.
15. What is the data structure used to keep track of live nodes in the best first of Branch and Bound? Why?

UNIT-VIII

1. Explain NP hard problems.
2 Explain modular arithmetic.
3. Given two sets S1 and S2, the disjoint sets problem is to check whether the sets have a common element. Present an O(1) time non-deterministic algorithm for this problem.
4. Given a sequence of n numbers, the distinct elements problem is to check if there are equal numbers.Give an O(1) time non-deterministic algorithm for this problem.
5. Obtain a non-deterministic algorithm of complexity O(n) to determine whether there is a subset of n numbers a(i), 1<=i<=n, that sums to m.

6. Show that the clique optimization problem reduces to the clique decision problem.
7. Show how to encode the following instructions as CNF formulas:
   (a) for (b) while
8. (a) What is meant by Halting problem explain with an example.
   (b) Explain the classes of P and NP.
9. (a) State and Explain cook's theorem.
   (b) Explain the satisfiability problem
10. a) Explain the satisfiability problem and write the algorithm for the same.
    (b) Differentiate between NP-complete and NP-Hard.
11.. (a) Explain the satisfiability problem and write the algorithm for the same.
    (b) Differentiate between NP-complete and NP-Hard. [10+6]
12. What is interpolation? Explain lagrange interpolation algorithm and newtonian interpolation algorithm.
13. A polynomial of degree n>0 has n derivatives, each one obtained by taking the derivative of the previous one. Devise an algorithm, which produces the values of a polynomial and its n derivatives.?
14. Explain the operations on polynomials of evaluation and interpolation?
15. Explain the transformation technique for polynomial products.

QUESTION BANK

UNIT- I

1. What is meant by time complexity? Define different time complexity notations. Give examples one for each. (Jan 10)

2. a) Define algorithm. Explain the characteristics of the algorithm.
b) Write an algorithm to find the largest of N numbers (Jan 10)

3. a) Write an algorithm to evaluate a polynomial using Horner's rule.
b) Present an algorithm that searches for the element x in unsorted array a[1:n]. If x occurs, then return a position in the array; else return zero. Evaluate its time complexity.
(june 09)

4. (a) Define time complexity. Describe different notations used to represent there complexities.
(b) Derive the function f(n)= 12n2+6n is 0(n3) and w(n). (Apr 09)

5. (a)Compute time complexity of recursive Fibonancci procedures where F(n)= F(n-1) + F(n-2)

(b) What does the following procedure outputs? Compute F1(1), F1(2), F1(3), F1(4).

```
Procedure F1(n)
if n < 2 then return (n)
else return (F2(2,n,1,1)
endif
end F1

procedure F2(I, n, x, y)
if I<= n
thecallF2 (I+1, n, y, x+y)
endif
return(y)
end F2
```

(June 09)

6. a) Show that f(n)+g(n)=O (n2) where f(n) = 3n2− n + 4 and g(n)=nlogn+5
b) Explain how time complexity of an algorithm is computed. (May 07,08)

7. Write the non recursive algorithm for finding the Fibonacci sequence and Show that n3logn is w(n3) derive its time complexity (May 08)

Unit II

1. a)What are the breadth first spanning trees? Explain.
b) Obtain the binary tree form for the following infix expressions.
i) (a + b) / (c * d)
ii) a + (b + (c + d)) (jan 10)

2. a) Write an algorithm to count the number of leaf nodes in a binary tree T.
b) Write a non-recursive procedure for pre-order traversal of a tree. (Jan 10)

3. a) Show that DFS visits all vertices in G reachable from v.
(b) Show that the number of leaves in a binary tree is more than the number of nodes of degree two.
(jan 10)

4. Use an AVL tree as the basis of an algorithm to execute MIN, UNION, and DELETE on sets consisting of integers 1 through n, using O(log n) steps per operation
(June 09)

5. What is preorder traversal. Write and explain a procedure for preorder traversal of a binary tree with an example in detail. Analyze the time & space complexity of your procedure (June 09)

6. a) Show that the inorder and post order sequences of a binary tree uniquely define the binary tree.
(b) Write a detailed note on depth-first traversal. (june 09)

7. a) Write an algorithm that uses BFS traversal to determine if the undirected and directed graphs are cyclic?
b) Show that the time complexity of the above algorithm is of the order n. (june 09)

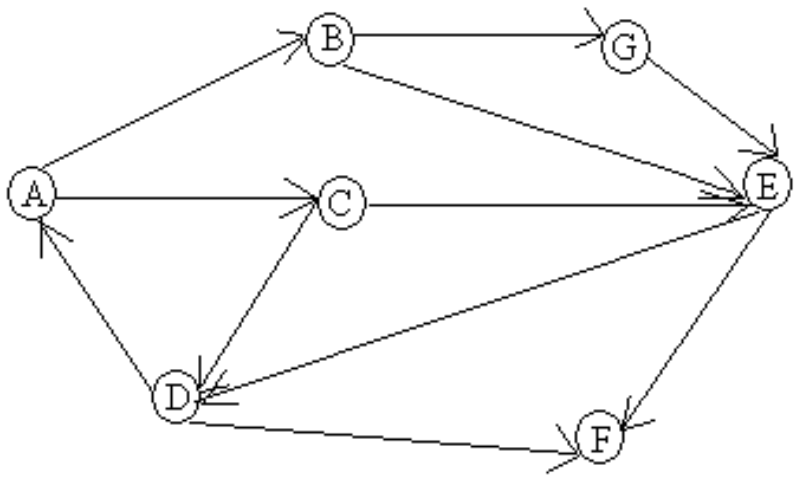
8. Explain the set representation using tree and develop algorithms for UNION and FIND using Weighing and Collapsing rules. (June 09)

9. a) Explain the union algorithm for weighting rule with an example and also analyze its complexity.
b) Write the deletion algorithm of heap and also analyze its complexity (Nov 09)

10. a) Explain the properties of strongly connected components.
(b) Write a non-recursive algorithm of In-order traversal of a tree and also analyze its time complexity.
(Apr 09)

11. Formulate an algorithm for a spanning tree problem in terms of a sequence of set operations in which take G as the undirected graph; S as the undirected tree; V as the number of vortices; E as the number of edges; T as a set used to collect the edges of the final minimum spanning tree; C as the cost function for the graph G given by C (e) for the sub graph $G^1=(V^1,E^1)$ of G. Use set VS for the vertex set of the trees in the spanning forest to write the minimum cost spanning tree algorithm. (Apr 06, Nov 08)

12. (a) Find the strongly connected components in the graph of figure 5a.



(b) Write a non-recursive algorithm of Pre-order traversal of a tree and also analyze its time complexity. (may 07)

Unit III

1. a) Explain the control abstraction for divide – and – conquer strategy.
b) Sort the records with the following index values in the ascending order using quick sort algorithm.
45, 24, 36, 59, 68, 7, 99, 17 (Jan 10)

2. a) Devise a version of Merge sort algorithm which performs sorting in-place.
b) Explain the control abstraction of divide and conquer strategy (Jan 10)

3. Show how quick sort sorts the following sequences of keys in ascending order.
22,55,33,11,99,77,55,66,54,21,32 (June 09)

4. Design a Divide and Conquer algorithm for computing the number of levels in a binary tree. Compute the complexity of the above algorithm (June 09)

5. Give the relative merits and demerits of Divide and Conquer when compared to the Greedy method. Give the time complexity of the generic Divide and Conquer algorithm
(june 09)

6. Determine the running time of Quick sort for
a. sorted Input
b. reverse-ordered input
c. random input.
(b) Design Divide and conquer algorithm for computing the no of levels in a binary Tree.
(May 09)

7. (a) Write an algorithm of Quick sort and explain in detail.
(b) Suggest refinements to Merge sort to make it in-place. (May 09)

8. Give an instance, where the quick sort algorithm has worst-case time complexity (Nov 08)

9. Write the non-recursive binary search algorithm. (May 05, Nov 08)

UNIT-IV

1. a)Find the solution for the instance n = 7
   (p1,p2,p3,p4,p5,p6,p7) = (3, 5, 20, 18, 1, 6, 30) and (D1, D2,D3,D4,D5,D6,D7) = (1, 3, 4, 3, 2, 1,2) by using Job sequencing with Deadlines.
   b) Write an algorithm of prim's minimum spanning tree. (Jan 10)
2. a) If objects are selected in order of decreasing vi/wi then prove that the algorithm Knapsack finds an optimal solution.
   (b) Write an algorithm of prim's minimum spanning tree. (June 09)
3. a) Explain the control abstraction of Greedy method compare this with Dynamic programming.
   b) Applying the Greedy strategy find the solution for optimal storage on tapes problem instance n = 3, (l1, l2, l3) = (5, 10, 3).
   (c) Explain the 0/1 knapsack problem algorithm with Greedy concept. (june 09)
4. (a) How many comparisons of edge weights will be done by the minimum spanning tree algorithm, in total, if the input is a complete undirected graph with n vertices and vi is the start vertex.
   (b) Design a linear-time algorithm for solving the single source shortest path algorithm for directed a cyclic graphs represented by their adjacency linked lists. (may 09)
5. Explain the Kruskal's algorithm with an example and analyze its time complexity. Prove that Prim's algorithm method generates the minimum cost spanning tree. (may 09)
6. Can Kruskal's and Prim's algorithms be applied to graph with negative edge weights? Explain. (may 09)
7. Prove that Kruskal's algorithm generates a minimum cost spanning tree for every connected undirected graph G. (Nov 05, Nov 08)
8. Explain the algorithm for Job sequencing with deadlines. Applying the same, find the solution for the instance n = 4, (p1.....p4)=(100,10,15,27) and (d1.....d4)=(2,1,2,1).
   (Nov 05,08)
9. Applying the Greedy strategy find the solution for optimal storage on tapes problem instance n = 3 $(l_1, l_2, l_3) = (5, 10, 3)$. (May 08)

UNIT – V

1. What is Travelling sales person problem and what are its applications. (Jan 10)
2. For the Travelling sales person algorithm show that the time complexity is $0(n^2 2^n)$ and space complexity is $O(n2^n)$. (may 07, jan 10)
3. Discus the dynamic programming solutions for the problems
   Reliability design and b) Traveling sales person. (june 10)
4. The edge length of a directed graph is given by any matrix of order 5*5. Using the traveling salesperson algorithm, calculate the optimal tour. (june 09)
5. Determine the cost and structure of an optimal Binary search tree for a set of n=5 keys with the following probabilities.
   p1 = p2 = 0.125, p3 = p4 = 0.0625, P5 = 0.1875
   q0 = 0, q1 = 0.1875, q2 = q3 = q4 = q5 = 0.0625
   The keys are a1 < a2 < a3 < a4 < a5. (may 09)

6. Write a pseudocode of the dynamic programming algorithm for solving Optimal Binary search tree and determine its time and space efficiencies. (may 09)
7. Find one problem for which the principle of optimality does not hold. Explain why the principle does not hold.
8. Find the shortest path between all pairs of nodes in graph with example. (may 09)
9. Find an OBST for a, b, ..., h if the elements in order have the probabilities {0.1, 0.2, 0.05, 0.1, 0.3, 0.05, 0.15, 0.05} and all the other elements have zero probability. (b) Write the algorithm for OBST. (june 08)
10. Discuss briefly the solution to the traveling salesperson problem using dynamic programming. Can it be solved by using divide and conquer method? (Nov 05,08)

UNIT VI

1 Explain the Kruskal's algorithm with an example and analyze its time complexity. (Jan- 2010**)**
2 Define multistage graphs problem. Name the algorithms, which solve the problem. Write one of the algorithms and explain its working with the help of an example (Apr/may 09)
3. a) Write an algorithm of estimating the efficiency of backtracking.
b) Let w = { 5, 10, 12, 13, 15, 18 } and m = 30. Find all possible subsets of w that sum to m. Draw the portion of the state space tree that is generated. (Jan 2010)

5. a) Draw and explain the portion of the tree for 4-queens problem that is generated during backtracking.
b) Explain the applications of Backtracking. (May 09)

6 a) Write an algorithm of finding all m-colorings of a graph.
b) Describe the 4-queens problem using backtracking. (May 09)

7 Write the control abstraction of backtracking write backtracking algorithm for 8- queen problem. (Apr/May 09)

8 Explain in detail how the technique of backtracking can be applied to solve the 8-queens problem. Write an algorithm for this and explain. (Jan 2010)
9. a) Write an algorithm for solving n-queens problem. Also analyze its complexity
b) Solve the following recurrence relation by using substitution method

$$T(n) = 1 \qquad Where \quad n \le 4$$

$$= T(\sqrt{n}) + c \qquad Where \quad n > 4$$

(Nov 09)

10. a) What is graph coloring? Present an algorithm which finds m-coloring of a graph.
b) Explain the recursive Backtracking algorithm. (June 09)

UNIT VII

1. a) Write a complete LC - branch and bound algorithm for knapsack problem.
b) Explain traveling sales person problem with an example. (JAN- 2010**)**

2. (a) Define the term branch and bound technique Explain it with an Example.
(b) Explain Properties of LC - Search. (May/Jun 2009)

3. Define the following terms: state space, explicit constraints, implicit constraints, problem state, solution states, answer states, live node, E-node, dead node, bounding functions. (Apr/May 2009)

4. Apply the branch and bound algorithm to solve the TSP for the following cost matrix.

$$\begin{bmatrix} \infty & 11 & 10 & 9 & 6 \\ 8 & \infty & 7 & 3 & 4 \\ 8 & 4 & \infty & 4 & 8 \\ 11 & 10 & 5 & \infty & 5 \\ 6 & 9 & 5 & 5 & \infty \end{bmatrix}$$

(Jan 2010)

5. a) Describe problem state, solution state and answer state with an example.
   b) Explain the general method of Branch and Bound. (May 09)

6 (a) Write an algorithm to solve the Knapsack problem with the Branch and Bound
   (b) Differentiate between Dynamic Knapsack and Branch and Bound Knapsack problem. (May 09)

7 (a) Explain the method of reduction to solve TSP problem using Branch and Bound.
   (b) Explain the principles of FIFO Branch and Bound. (May 09)

8. Draw the portion of the statespace tree generated by FIFO Branch and Bound for the following instances n=4, m=15, (P1.....P4)= (10, 10, 12, 18), (w1.....w4) = (2, 4, 6, 9). (May 09)

9. Draw the portion of the statespace tree generated by FIFO Branch and Bound for the following instances. N=5, M=15, (p1.....p5) =(W1.....W5) =(4, 4, 5, 8, 9). (May 09)

10. a) Write LC branch and bound to find minimum cost answer node algorithm.
    b) Write a program schema DFBB for a LIFO branch and bound search for a least cost answer node. (Jan 10)

11. Solve the following instance by using FIFOBS n = 4, m = 40, P =(11, 21, 31, 33) w = (2, 11, 22, 15) Also draw the state space tree.
(Nov 09)

12. a)Explain the Knapsack problem.
    b) Find an optimal solution to the Knapsack instance n=7, m=15, and (p1, p2,p3,p4 p5 p6, p7)=(10,5,15,7,6,18,3) and w1,w2,w3,w4,w5,w6,w7)=(2,3,5,7,1,4,1) (Nov 09)

13. Present a program schema for a FIFO Branch & Bound search for a Least-Cost answer node. (June 09)

14 The edge length of a directed graph are given by the below matrix. Using the traveling salesperson algorithm, calculate the optimal tour.

$$\begin{bmatrix} 0 & 20 & 30 & 10 & 11 \\ 15 & 0 & 16 & 4 & 2 \\ 3 & 5 & 0 & 2 & 4 \\ 19 & 6 & 18 & 0 & 3 \\ 16 & 4 & 7 & 16 & 0 \end{bmatrix}$$

(Nov 09)

15 a) Explain the solution to the Traveling sales person problem using LCBB.
   b) Is the above technique applicable for a non-symmetric distance matrix? Substantiate.

(June 09)

16 Explain the principles of

a) Control Abstraction for LC-search
b) Bounding
c) FIFO Branch & Bound
d) LIFO Branch & Bound

(June 09)

UNIT VIII

1. Show the tuples which would result by representing the polynomials, $5x^2 + 3x + 10$ and $7x + 4$ at the values x = 0, 1, 2, 3, 4, 5, 6. What set of tuples are sufficient to represent the product of these two polynomials. (JAN- 2010**)**

2. a) State and Explain cook's theorem.
b) Explain the satisfiability problem. (May/Jun 2009)

3. What is interpolation? Explain Lagrange interpolation algorithm & Newtonian Interpolation algorithm (Apr/May 2009)

4. a) Explain the classes of NP-hard and NP-complete.
b) Write a nondeterministic sorting algorithm. (Jan 2010)

5. a) What is meant by Halting problem explain with an example.
b) Explain the classes of P and NP. (May 09)

6 a) Explain the classes of P and NP.
b) Write a nondeterministic Knapsack algorithm. (May 09)

7 a) Explain about cook's theorem.
b) Explain the strategy to prove that a problem is NP hard. (May 09)

8 a) Write the Nondeterministic sorting algorithm. Also Analyze its complexity.
b) Distinguish between deterministic and non-deterministic algorithms. (Apr/May 09)

9. a) Explain the satisfiability problem and write the algorithm for the same.
b) Differentiate between NP-complete and NP-Hard. (Apr/May 09)

10 a) Explain the nim game.
b) Generate the complete game tree for nim with n=6. (June 09)