PESIT Bangalore South Campus

15CS43: DESIGN AND ANALYSIS OF ALGORITHMS

Faculty : Prof.Sandesh B J, Prof.Vandana M L

No Of Hours: 56

Course Description:
This course focuses on various techniques for designing the algorithms. It discusses various approaches to determine the algorithm performance. It includes application of algorithm design techniques to various problems

Prerequisites:
Programming in C/Java language

Course Objectives:
- Explain various computational problem solving techniques.
- Apply appropriate method to solve a given problem.
- Describe various methods of algorithm analysis.

Course Plan:

Class	Chapter Title	Topics to be covered	Book	% of portion covered	
					Cumulative
1.	Module I	**Introduction:**What is an Algorithm?	T2:1.1	20%	20%
2.		Algorithm Specification	T2:1.2		
3.		Analysis Framework	T1:2.1		
4.		**Performance Analysis**: Space complexity, Time complexity	T2:1.3		
5.		**Asymptotic Notations:**Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ), and Little-oh notation (o)	T1:2.2		
6.		Mathematical analysis of Non-Recursive Algorithms with examples	T1:2.3		
7.		Mathematical analysis of recursive Algorithms with Examples	T1:2.4		
8.		**Important Problem Types:** Sorting, Searching, String processing	T1:1.3		
9.		Graph Problems, Combinatorial Problems	T1:1.3		
10.		**Fundamental Data Structures:** Stacks,	T1:1.4		

		Queues, Graphs,			
11.		Trees, Sets and Dictionaries.	T1:1.4		
12.	**Module II**	**Divide and Conquer:**General method	T2:3.1	**20%**	**40%**
13.		Binary search	T2:3.1		
14		Recurrence equation for divide and conquer	T2:3.3		
15.		Finding the maximum and minimum	T2:3.4		
16.		Merge sort	T1:4.1		
17.		Quick sort	T1:4.2		
18.		Stassen's matrix multiplication	T2:3.8		
19.		Advantages andDisadvantages of divide and conquer	T1:5.3		
20.		**Decrease and Conquer Approach:** TopologicalSort.	T1:5.3		
21.	**Module III**	**Greedy Method:** General method, Coin Change Problem,	T2:4.1	**20%**	**60%**
22.		Knapsack Problem	T2:4.3		
23.		Job sequencing with deadlines	T2:4.5		
24.		**Minimum cost spanning trees:** Prim's Algorithm	T1:9.1		
25		Prim's Algorithm: Example	T1:9.1		
26.		Kruskal's Algorithm	T1:9.2		
27.		Kruskal's Algorithm :Example	T1:9.2		
28.		**Single source shortest paths:** Dijkstra's Algorithm	T1:9.3		
29.		Dijkstra's Algorithm : Example	T1:9.3		
30.		**Optimal Tree problem:** Huffman Trees and Codes	T1:9.4		
31.		**Transform and Conquer Approach:** Heaps and Heap Sort (**T1:6.4**).	T1:6.4.		
32.		**Transform and Conquer Approach:** Heaps and Heap Sort (**T1:6.4**).	T1:6.4.		
33.	**Module IV**	**Dynamic Programming:** General method with Examples,	T2:5.1	**20%**	**80%**
34.		Multistage Graphs	T2:5.2		
35.		**Transitive Closure:** Warshall's Algorithm	T1:8.2		
36.		**All Pairs Shortest Paths:** Floyd's Algorithm,	T1:8.2		
37.		Optimal Binary Search Trees	T1:8.3		
38.		Knapsack problem	T1:8.4		
39.		Knapsack problem & Memory Functions	T1:8.4		
40.		Bellman-Ford Algorithm	T2:5.4		
41.		Bellman-Ford Algorithm: Example	T2:5.4		
42.		Travelling Sales Person problem	T2:5.9		
43.		Reliabilitydesign	T2:5.8		
44.	**Module V**	**Backtracking:** General	T2:7.1	**20%**	**100%**

45.		N-Queens problem	T1:12.1		
46.		Sum of subsetsproblem	T1:12.1		
47.		Graph coloring	T2:7.4		
48.		Hamiltonian cycles	T2:7.5		
49.		**Branch andBound:** Assignment Problem,	T1:12.2		
50.		Travelling Sales Person problem	T1:12.2		
51.		**0/1 Knapsack problem :** LC Branch and Bound solution	T2:8.2, T1:12.2		
52.		FIFO Branch and Bound solution	T2:8.2		
53.		**NP-Complete and NP-Hard problems:** Basic Concepts	T2:11.1		
54.		non-deterministic algorithms, P, NP	T2:11.1		
55		,NP-Complete, NP-Hard classes	T2:11.1		
56		Revision			

Course Outcomes:

After studying this course, students will be able to:

- Describe computational solution to well-known problems like searching, sorting etc.
- Estimate the computational complexity of different algorithms.
- Devise an algorithm using appropriate design strategies for problem solving.

TEXT BOOK:

T1. Introduction to the Design and Analysis of Algorithms, AnanyLevitin:, 2nd Edition, 2009, Pearson.

T2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014,Universities Press

REFERENCE BOOKS:

R1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest,Clifford Stein, 3rd Edition, PHI

R2. Design and Analysis of Algorithms , S. Sridhar, Oxford (Higher Education)

QUESTION BANK:

MODULE1

1. What is an algorithm? Write step by step procedure to write an algorithm.

2. What are the properties of an algorithm? Explain with an example.

3. Define asymptotic notations for worst case, best case and average case time complexities.Give examples.(OR) Explain all asymptotic notations used in algorithm analysis.(OR)Explain in brief the basic asymptotic efficiency classes.(OR) Explain the worst-case, bestcaseand average case efficiencies of an algorithm with help of an example.

4. Explain the general plan for analysing the efficiency of a recursive algorithm.

5. Explain the method of comparing the order of the growth of 2 functions using limits.Compare order of growth of (i) log2 n and sqrt(n) ii) (log2 n)2 and log2 n2 .

6. Discuss the general plan for analyzing efficiency of non recursive algorithms.

7. Write an algorithm to compute n! recursively. Set up a recurrence relation for the algorithm's basic operation count and solve it.

8. Consider the following algorithm
Algorithm Enigma (A[0· ·n -1, 0..n-1])
for i = 0 to n-2 do
for j=i+1 to n-1 do
if A[i,j]Not equal to A[j,i]
return false
end for
end for
return true
end algorithm
i) What does this algorithm compute?
ii) What is its basic operation
iii) How many times is the basic operation executed?

9. Write the algorithm to compute the sum of n numbers and indicate
(a) Natural size metric for its inputs
(b) Its basic operation
(c) Whether the basic operation count can be different for inputs of the same size.

10. Consider the following recursive algorithm for computing the sum of the first n cubes.S (n) = 13 + 23 + 33 + ... + n3
Algorithm S (n)
if(n = 1) return 1

else return (S(n-1)+n*n*n)
end algorithm

Set up and solve a recurrence relation for the number of times the algorithm's basic operation is executed.
.
11. Ift1(n) € 0 (gr (n)) and t2 (n) € 0(gz (n)),
prove that t1 (n) + t2 (n) € 0(max {gr (n), gz(n)}).

12. If M(n) denotes the, number of moves in tower of Hanoi puzzle when n disks are involved,give a recurrence relation for M(n) and solve this recurrence relation

13. With a suitable example explain the significance of Order of Growth in analyzing thealgorithm efficiency

14. What is a "Brute force "method? Under what condition does the method become desirable?

15. a. Using bubble sort algorithm arrange the letters of the word "QUESTION" in alphabetical order
b. Explain brute force method for algorithm design and analysis. Explain the brute force string matching algorithm with its efficiency.

MODULE2

16. Explain the general concept of divide - and conquer method. Show how binary searchproblem can be solved using the same method, find its average case efficiency.

17. Define Master theorem. Compute the time complexity for the following recurrence equation
using the same.
i) T (n) = 4T *(nl2)* + n, T(1) = 1; ii) T(n) = 4T *(nl2)* +n2, T(1) = 1
iii) T (n) = 4T *(nl2)* +n3, T (1) = 1; iv) T (n) = 2T *(nl2)* + Cn, T (1) = 0

18. Give the general divide and conquer recurrence and explain the same. Give the Master'stheorem.

19. Discuss the merge sort algorithm with recursive tree and its efficiency. Apply the same algorithm to sort the list {4,6,1,3,9,5}

20. Write the quick sort algorithm. Analyze its efficiency. Apply the algorithm to sort thelist 4, 1, 6, 3, 9, 2, 7, 5 .Derive worst case complexity of it.

21. Using quick sort algorithm arrange the letters of the word "Example" and "Question" inalphabetical order.

22. Write the algorithm for binary search and find the average case efficiency.

.

23. Give an algorithm (recursive) for merge sort.

24. Consider the numbers given below. Show how partitioning algorithm of quick sort willplace106 in its correct position. Show all the steps clearly.
106 117 128 134 141 91 84 63 42

25. Consider the set of 14 elements in array list:-
15,16,0,7,9,23,54,82,101,112,125,131,142,151. When binary search is applied on theseelements, find the elements which required maximum number of comparisons. Alsodetermine the average number of key comparisons for successful and unsuccessful search.

26. Explain divide and conquer technique. Write the algorithm for binary search and findaverage case efficiency.What is stable algorithm? Is quick sort stable? Explain with an example.

27.Explain the algorithm for Strassen's Matrix Multiplication

28.Explain DFS based and source Removal algorithm for topological Sort

MODULE3

29. What is greedy technique? Explain in detail the general method.

30. What is spanning tree? What is minimum spanning tree?

31. Justify the statement "Prim's algorithm always yields minimum cost spanning tree". Givethe prim's algorithm and discuss its time complexity.

32 . Give kruskal's algorithm and discuss its time complexity.

33. Give the Dijkstra's algorithm. What is its complexity? Discuss with a simple example.

34. Use the kruskal's algorithm to solve the min cost spanning tree problem

35. Use the prim's algorithm to solve the min cost spanning tree

36.ExplainDijkstra's algorithm to solve single source shortest path problem

37.Solve the following knapsack problem with given capacity W= 5 using Greedy method.

item	weight	value
1	2	$12
2	1	$10
3	3	$20
4	2	$15

38.Solve the following knapsack problem with given capacity W= 10 using Greedy method

item	weight	value
1	4	$40
2	7	$42
3	5	$25
4	3	$12

39. Obtain the optimal solution for the job sequencing problem with deadlines where n=4(no ofjobs), profits (p1,p2,p3,p4) = (100,10,15,27) and deadlines (d1,d2,d3,d4) = (2,1,2,1)

40. Let J be at set of K Jobs and Sigma=i1, i2, i3, …..,ik be a permutation of jobs in J such thatdi1<di2< di3<……. di1<dik. Prove that J is a feasible solution if and only if the jobs in J can beProcessed in the order sigma without violating any deadline.

41. Define i) Optimal solution ii) Feasible solution. Will greedy method yield an optimalsolution always?

42. Explain Heap Sort with an example

43. Explain algorithm to construct Huffman Tree

MODULE 4

44. Explain Dynamic programming.

45. Write the formula to find the shortest path using Floyd's approach. Use Floyd's method tosolve the below all-pairs shortest paths problem.
(_ stands for infinity)

0 _ 3 _
2 0 _ _
_ 7 0 1
6 _ _ 0

46. State all pair shortest path algorithm. Solve the all pairs shortest path problem for thediagraph with the weight matrix.
(_ stands for infinity)

0 2 _ 1 8
6 0 3 2 _
_ _ 0 4 8
_ _ 2 0 3
3 _ _ _ 0

47. Explain Warshall's Algorithm in detail.

48. Using dynamic programming, solve the following knapsack instance:
n=3,[w1,w2,w3]= [1, 2,2] and [p1,P2,P3] =[18, 16,6] and M=4

49. Explain algorithm to solve traveling sales person problem, using dynamic programming

50. List out the difference between divide and conquer and dynamic programming.

51. Using Warshall's algorithm, obtain the transitive closure of the matrix given below:
0 0 0 0
0 0 0 1
0 0 0 0
1 0 1 0

52. Outline an exhaustive search algorithm to solve traveling salesman problem

53. Write a note on multistage graph

54.write a note on reliability design

55.Explain Bellman ford algorithm with an example

MODULE 5

56. Define 1.backtracking 2. Implicit constraints 3.Explicit constraints 4. State/solution

57.state/solution space 6.state /solution space tree

58. Draw the state – space tree to generate solutions to 4 – Queen's problem. (OR) Explain howbacktracking is used for solving 4- queens problem. Show the state space tree.

59. Explain approximation algorithms for NP-hard problems in general. Also discuss Approximation algorithms for knapsack problem.

60. State subset sum Problem. Use backtracking, obtain a solution to the subset sum problemby taking (i) S={6, 8, 2, 14} and d=16. (ii) S={5,10,12,13,15,18} d=30
61. What is branch – and - bound algorithm? How it is different from backtracking?

62. Write the steps and apply nearest neighbor approximation algorithm on the TSP problem with the starting vertex a, and calculate the accuracy ratio of approximation.

63. Write short notes on: a) Travelling Sales person Problem b) Input Enhancement in String Matching c) Decision Tree. d) Challenges of numerical algorithms.

—
_
64.Write a note on Graph coloring algorithm

65.Explain an algorithm to determine Hamiltonian Cycle

66. Solve the following instance of Assignment problem using Branch-Bound technique
Job1 Job2 Job3 Job4
9 2 7 8
6 4 3 7
5 8 1 8
7 6 9 4

67. Solve the knapsack problem using branch and bound given the following data: M=10 totalitem=4, Weights= 4,7,5,3 Value = 40,42,25,12 value/weight = 10,6,5,4

68. Apply branch and bound algorithm to solve the travelling salesman problem

69.Apply Branch and Bound method to solve Knapsack problem

70. Explain the concepts of P, NP, and NP – complete problems. **(OR)** Write short notes on P,NP and NP-complete problems **(OR)** Define P, NP and NP complete problems

71. Define the following: i) Tractable Problems, ii) Class p, iii) Class NP, iv) PolynomialReduction, v) NP Complete