CS6402/ DESIGN AND ANALYSIS OF ALGORITHMS
IMPORTATANT 16 MARK QUESTIONS

UNIT - 1

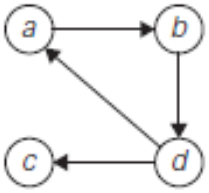
1. Describe briefly the notions of complexity of an algorithm. ***(June 07)***
2. Explain the fundamentals of algorithmic problem solving. (OR) Discuss briefly the sequence of steps in designing and analyzing an algorithm. ***(Dec 06)***
3. Explain the various asymptotic notations of an algorithm in detail. ***(Dec 07)***
4. Elaborate on how space and time complexities are calculated. Give Examples. **(*May 13)***
5. Explain the general framework for analyzing the efficiency of algorithm. (or) Discuss the fundamentals of analysis framework. ***(Dec 06, 07)***
6. Explain some of the important problem types used in the design of algorithm with an example. ***(Dec 06, 07)***
7. Give a short account of basic efficiency classes ***(Dec 06)***
8. Design a non recursive algorithm for computing the product of two n*n matrices and also find time efficiency of algorithm***.(Dec 06)***
9. Design a recursive algorithm to find the number of moves in tower of Hanoi problem and find the time of complexity. ***(Dec 013)***
10. Design a recursive algorithm to compute the factorial function F (n) =n! and derive the recurrence relation***.(Dec 06)***

UNIT – 2

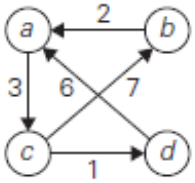
1. Explain how brute force approach is applied to solve closest-Pair and convex-Hull problem.
2. Explain the concept of travelling salesman problem, knapsack problem and assignment problem using exhaustive search.
3. Explain the concept of knapsack problem using exhaustive search.
4. Explain in detail about merge sort. Illustrate the algorithm with a numeric example***.(Dec 06)***
5. Devise an algorithm to sort the following elements using Merge sort technique 286, 45,278,368,475,389,656,788,503,126. ***(Dec 13)***
6. Write an algorithm for merge sorting. Show the intermediate steps when the numbers 310, 285, 179, 652, 351, 423, 861, 254, 450, 520 are sorted using Merge Sort***.(Dec 14)***
7. Distinguish between quick sort and merge sort, and arrange the following numbers in increasing order using merge sort.(18,29,68,32,43,37,87,24,47,50). ***(May 13)***
8. Write an algorithm for binary search using divide and conquer and analyze the time complexity***.(June 06,Dec 14)***
9. Write Strassen's matrix multiplication algorithm. Is there any time efficiency improvement compared to ordinary matrix multiplication?
10. Explain how divide and conquer method is applied to solve closest-Pair and convex-Hull problem.

UNIT – 3

1. Explain how dynamic programming technique is applied to compute binomial coefficient.
2. Explain Warshall's algorithm to find the transitive closure with an example.
3. Explain the pseudo code for Warshall's algorithm and apply for the following diagraph.



4. Explain an algorithm to find optimal binary search tree with example***.(Dec 07,14)***
5. Write and explain Floyd's algorithm for the all-pairs shortest path problem. Using this find the length of the shortest path between all pairs of vertices of the following graph. ***(Dec 06,07)***

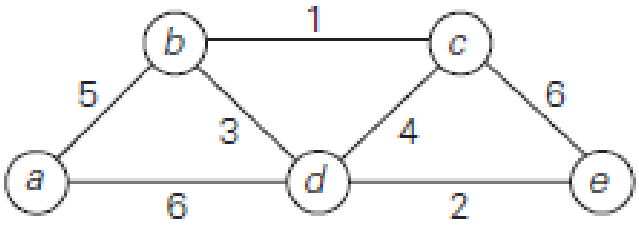


6. Write an memory function algorithm to solve the following knapsack problem. ***(Dec 07)***

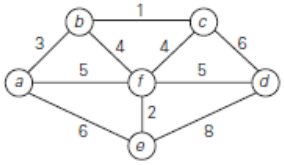
item	weight	value
1	2	$12
2	1	$10
3	3	$20
4	2	$15

Knapsack capacity W= 5

7. Explain the method for finding the minimum spanning tree for a connected graph using Prim's algorithm with an example. ***(June 06,Dec 14)***
8. Apply Kruskal's algorithm to find a minimum spanning tree of the following graph. ***(Dec 07)***



9. How will find the shortest path between two given vertices using Dijikstra's algorithm? Explain the pseudo code with an example. ***(June 06,07)***
10. Explain the pseudo code for prim's algorithm and apply the same to minimum spanning tree for the following graph. ***(Dec 07)***



UNIT – 4

1. Solve the instance of the stable marriage problem given by the ranking matrix and find the stable and unstable matching.

	Ann	Lea	Sue
Bob	2,3	1,2	3,3
Jim	3,1	1,3	2,1
Tom	3,2	2,1	1,2

2. Write the pseudo code for maximum bipartite matching, and explain with an examples.
3. Explain maximum matching problem in bipartite graphs.
4. Write the pseudo code for Shortest Augmenting Path method and explain in detail.
5. Explain the maximum flow problem with an example.
6. Use simplex method to solve the LPP

   maximize $Z = 4x + 10y$
   subject to $2x + y \leq 50$
   $2x + 5y \leq 100$
   $2x + 3y \leq 90$
   $x \geq 0, y \geq 0.$
7. Solve the following problem using simplex method:

   maximize $Z = 3x + 5y$
   subject to $x + y \leq 4$
   $x + 3y \leq 6$
   $x \geq 0, y \geq 0.$
8. Explain the steps to solve simplex method using linear programming with an example.

UNIT – 5

1. Apply the branch and bound algorithm to solve the following knapsack problem and explain in detail.
   (Dec 06)

Item	Weight	value
1	2	1
2	3	2
3	4	5

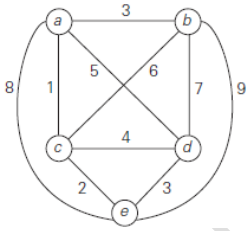
   The knapsack capacity W is 6.

2. Apply the branch and bound algorithm to solve the following knapsack problem and explain in detail.
(Dec 06)

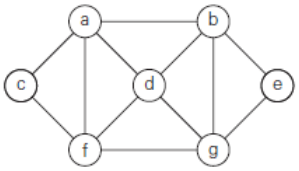
Item	Weight	value
1	4	$40
2	7	$42
3	5	$25
4	3	$12

The knapsack capacity W is 10.

3. Explain in detail about branch and bound technique with an example. ***(June 07)***
4. Solve the following instance of the travelling salesman problem by branch and bound method and explain in detail. ***(Dec 07)***



5. Explain knapsack problem using branch and bound technique. ***(Dec 07)***
6. What is backtracking? Write the template of general backtracking algorithm and explain in detail with suitable example. ***(June 07)***
7. Explain n-queen's problem. Draw a portion of the state space tree and perform backtracking search for a solution to 4-queens problem. ***(June 06,Dec07,14)***
8. Explain how backtracking method is applied to solve subset sum problem with suitable example***.(June 0,Dec 07)***
9. Write a pseudo code for backtracking algorithm and apply backtracking to solve the following instances of the subset sum problem: S= {1, 3, 4,5} d=11 and d=8. ***(Dec 06)***
10. Explain Hamiltonian circuit in a graph. Use backtracking to get a Hamiltonian circuit of following the graph..



11. Explain knapsack problem using branch and bound technique. ***(Dec 07)***