VALLIAMMAI ENGINEERING COLLEGE

SRM Nagar, Kattankulathur-603203.

Department of Information Technology

Question Bank- Even Semester 2014-2015

CS6402 – Design Analysis and Algorithm

Handled By,

Mr.K.Ravindran, A.P (Sr.G) & Mr. M.Kavinkumar, A.P(O.G)

Unit-I

Part-A

1. What is an Algorithm?
2. Define Sequential Algorithms and Parallel Algorithms?
3. What is Exact and Approximation algorithm?
4. What is Algorithm Design Technique?
5. Explain Algorithm's Correctness
6. What is Efficiency of algorithm?
7. What is generality of an algorithm?
8. What is algorithm's Optimality?
9. What do you mean by Sorting problem?
10. What do you mean by Searching problem?
11. What do you mean by Worst case and Best case- Efficiency of an algorithm?
12. Define the Average-case efficiency of an algorithm?
13. What do you mean by Amortized efficiency?
14. How to measure the algorithm's efficiency and algorithm's running time?
15. What is called the basic operation of an algorithm?
16. Define order of growth.
17. Define Big-oh notation and prove that $100n + 5 \quad O(n^2)$?
18. Define Ω notation and prove that $n^3 \quad (n^2)$?

19. Define Θ–notation and Prove that (½) n (n-1) (n^2)
20. What is the use of Asymptotic Notations?

Part-B

1. (a) Describe the steps in analyzing & coding an algorithm.
   (b) Explain some of the problem types used in the design of algorithm.
2. (a) Discuss the fundamentals of analysis framework.
   (b)Explain the various asymptotic notations used in algorithm design.
3. (a) Explain the general framework for analyzing the efficiency of algorithm.
   (b)Explain the various Asymptotic efficiencies of an algorithm.
4. (a) Explain the basic efficiency classes.
   (b)Explain briefly the concept of algorithmic strategies.
5. Describe briefly the notions of complexity of an algorithm.
6. (a) What is Pseudo-code? Explain with an example.
   (b)Find the complexity C(n)of the algorithm for the worst case, best case and average case.(Evaluate average case complexity for n=3,where n is the number of inputs)
7. Setup & solve are currence relation for the number of key comparisons made by above pseudocode.
8. Explain mathematical analysis of non- recursive algorithm
9. Explain mathematical analysis of recursive algorithm
10. Write short notes on limits for comparing orders of growth with example

Unit-II

Part-A

1. Explain divide and conquer algorithms
2. Define Merge Sort
3. Define Binary Search
4. Is Merge Sort and Quick Sort a stable sorting algorithm
5. Define exhaustive search give examples

6. Solve the average case recurrence for quick sort
7. What can we say about the average case efficiency of binary search
8. How divide and conquer technique can be applied to binary trees?
9. Explain Internal and External Nodes
10. Define Preorder, inorder and postorder Traversal
11. Define the Internal Path Length
12. Define the External Path Length
13. Explain Traveling salesman problem"?
14. Explain Knapsack problem
15. Define Brute force algorithm
16. Write an algorithm for matrix multiplication
17. Write the strength and weakness of brute force algorithm
18. Define Closest Pair
19. Define Convex Hull
20. Define Assignment problem

Part-B

1. Write a pseudocode for divide & conquer algorithm for merging two sorted arrays in to a single sorted one. Explain with example.
2. Explain0/1knapsackproblemwithexample.
3. Discuss the solution for Travelling salesman problem using branch & bound technique
4. Explain about Knapsack Problem with example
5. Explain Merge Sort and Quick Sort algorithm
6. Write a program implementing the brute force algorithm for the convex hull problem
7. Explain Brute force Closest pair algorithm
8. Explain Binary search tree
9. Explain how exhaustive search can be applied to the sorting problem and determine the efficiency class of such an algorithm
10. Give an example of the assignment problem whose optimal solution does not include the smallest element of its cost matrix

Unit-III

Part-A

1. Define Dynamic Programming
2. Define Binomial Coefficient
3. Define Transitive closure
4. Explain Warshalls algorithm
5. Explain All-pair shortest-paths problem
6. Explain Floyd's algorithm
7. What does Floyd’s algorithm do?
8. Explain principle of Optimality
9. Explain Optimal Binary Search Trees
10. Explain Knapsack problem
11. Explain the Memory Function technique
12. Explain about greedy technique
13. Define Spanning Tree
14. Define Minimum Spanning Tree
15. Define min-heap
16. Define Kruskal's Algorithm
17. Define Prim's Algorithm
18. Define Binary Tree
19. Explain Dijkstra's Algorithm
20. Define Huffman tree and Huffman code

Part-B

1. Construct a minimum spanning tree using Kruskal’s algorithm with your own example.
2. Explain Kruskal’s algorithm.
3. Explain about Knapsack Problem with example
4. Explain Dijikstra algorithm
5. Define Spanning tree. Discuss design steps in Prim’s algorithm to construct Minimum spanning tree with an example.

6. Explain about binary search with example.
7. Explain Warshall's & Floyd's Algorithm.
8. Explain about Huffman trees graphs with example.
9. Explain 0/1 knapsack problem with example
10. Define optimal binary search trees with example.

Unit-IV

Part-A

1. Explain Maximum Flow time complexity
2. Explain maximum flow problem
3. What is a cut
4. How will you find minimum cut
5. Define simplex method
6. Define a bipartite graph
7. How will you check the stability
8. Define the term stable pair
9. Define the term fixed pair
10. Define altering path
11. Define augmenting path
12. Define Hall's theorem
13. Difference between maximum flow and minimum cut
14. What is the capacity of a cut
15. Explain bipartite perfect matching polytope
16. Define Maximum Bipartite Matching
17. What is the running time of Ford-Fulkerson
18. State the minimum weight perfect matching problem
19. What do you mean by dual linear program?
20. What do you mean by residual network?

Part-B

1. Write short notes on simplex method
2. Write short notes on maximum flow problem
3. Explain the maximum flow algorithm
4. Explain briefly on bipartite graph
5. Explain briefly on stable marriage problem
6. Explain briefly on Fold-Fulkerson algorithm
7. Explain briefly on reducing bipartite graph to net flow
8. Write short notes on Fold-Fulkerson
9. Explain briefly on minimum weight perfect matching algorithm
10. Explain briefly on bipartite perfect matching polytope

Unit-V

Part-A

1. Explain Backtracking
2. Explain State Space Tree
3. Explain promising and non promising node
4. Explain n-Queens problem
5. Explain Subset-Sum Problem
6. Explain Branch and Bound Technique
7. Define Feasible Solution and Optimal solution
8. Explain Graph coloring problem.
9. Mention two reasons to terminate a search path at the current node in a state-space tree of a branch and bound algorithm.
10. What is articulation point?
11. List out the techniques for traversals in graph.
12. Define Traversals.
13. Explain NP-Hard problems
14. Define a Heuristic
15. When a decision problem is said to be polynomially reducible
16. Explain NP-complete problems

17. Explain class NP problems
18. Explain the halting problem
19. Explain undecidable problems
20. Explain class P problems

Part-B

1. Explain the 8-Queen's problem & discuss the possible solutions
2. Solve the following instance of the knapsack problem by the branch & bound algorithm.
3. Apply backtracking technique to solve the following instance of subset sum problem: S={1,3,4,5} and d=11(16)
4. Explain subset sum problem & discuss the possible solution strategies using backtracking.
5. Explain Graph coloring with example.
6. Explain about Knapsack Problem using backtracking with example.
7. Explain about biconnected components with example.
8. Briefly explain NP-Hard and NP-Completeness with examples.
9. Explain about 0/1 Knapsack Problem using branch and bound with example.
10. Explain briefly about Hamiltonian Circuit problem