# Syllabus

**Subject Name :DAA** **Subject Code: 15CS43**

| **Module 1** |
|---|
| **Introduction:** What is an Algorithm? **(T2:1.1),** Algorithm Specification **(T2:1.2),** Analysis Framework **(T1:2.1), Performance Analysis**: Space complexity, Time complexity **(T2:1.3)**. **Asymptotic Notations:** Big-Oh notation (*O*), Omega notation (Ω), Theta notation (□), and Little-oh notation (*o*), Mathematical analysis of Non-Recursive and recursive Algorithms with Examples **(T1:2.2, 2.3, 2.4). Important Problem Types:** Sorting, Searching, String processing, Graph Problems, Combinatorial Problems. **Fundamental Data Structures:** Stacks, Queues, Graphs, Trees, Sets and Dictionaries **(T1:1.3,1.4)** |
| **Module 2** |
| **Divide and Conquer**: General method, Binary search, Recurrence equation for divide and conquer, Finding the maximum and minimum **(T2:3.1, 3.3, 3.4)**, Merge sort, Quick sort **(T1:4.1, 4.2)**, Strassen's matrix multiplication **(T2:3.8)**, Advantages and Disadvantages of divide and conquer. **Decrease and Conquer Approach:** Topological Sort. **(T1:5.3)** |
| **Module 3** |
| **Greedy Method:** General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines **(T2:4.1, 4.3, 4.5). Minimum cost spanning trees:** Prim's Algorithm, Kruskal's Algorithm **(T1:9.1, 9.2)**. **Single source shortest paths:** Dijkstra's Algorithm **(T1:9.3)**. **Optimal Tree problem:** Huffman Trees and Codes **(T1:9.4)**. **Transform and Conquer Approach:** Heaps and Heap Sort **(T1:6.4)**. |
| **Module 4** |
| **Dynamic Programming:** General method with Examples, Multistage Graphs (T2:5.1,5.2). **Transitive Closure:** Warshall's algorithm, **All Pairs shortest Paths:** Floyd's algorithm, Optimal Binary Search trees, Knapsack problem **((T1:8.2, 8.3, 8.4), Bellman**-ford algorithm **(T2:5.4),** Travelling Sales Person Problem **(T2:5.9)**, Reliability design**.(T2:5.8).** |
| **Module 5** |
| **Backtracking:** General method **(T2:7.1)**, N-Queens problem **(T1:12.1)**, Sum of subsets problem **(T1:12.1)**, Graph coloring **(T2:7.4)**, Hamiltonian cycles **(T2:7.5)**. **Branch and Bound:** Assignment Problem, Travelling Sales Person problem **(T1:12.2)**, **0/1 Knapsack problem (T2:8.2, T1:12.2):** LC Branch and Bound solution **(T2:8.2)**, FIFO Branch and Bound solution **(T2:8.2)**. **NP-Complete and NP-Hard problems:** Basic concepts, non-deterministic algorithms, P, NP, NP-Complete, and NP-Hard classes |

**(T2:11.1)**.

# MODULE-1

**Introduction:** What is an Algorithm? **(T2:1.1),** Algorithm Specification **(T2:1.2),** Analysis Framework **(T1:2.1), Performance Analysis**: Space complexity, Time complexity **(T2:1.3)**. **Asymptotic Notations:** Big-Oh notation (*O*), Omega notation (Ω), Theta notation (*Θ*), and Little-oh notation (*o*), Mathematical analysis of Non-Recursive and recursive Algorithms with Examples **(T1:2.2, 2.3, 2.4). Important Problem Types:** Sorting, Searching, String processing, Graph Problems, Combinatorial Problems. **Fundamental Data Structures:** Stacks, Queues, Graphs, Trees, Sets and Dictionaries

**Questions:**

1. Define algorithm. Explain asymptotic notations, Big O, big Omega, big theta notations. (08 Marks June/July 2017)
2. Explain general plan of mathematical analysis of nonrecursive algorithms with example. (08 Marks June/July 2017)
3. Define time and space complexity.Explain important problem types. (08 Marks June/July 2017)
4. Define three asymptotic notations and express the following assertions using three asymptotic notations with proof from its definition i)n(n-1)/2 ii)6*2n+n iii)100n+5 (6m) [dec16-jan-17]
5. Give general plan of analyzing recursive algorithm. Mathematically analyze the tower of Hanoi problem and find its complexity. (8m) [dec16-jan-17] (08 Marks June/July 2017)
6. Prove that: If t1(n)□O(g1(n)) and t2(n)□O(g2(n)), Then t1(n)+t2(n))□O(max{ g1(n), g2(n)})(6m)[June 2012]
7. Define asymptotic notation for analyzing algorithm. Give at least one example for each.(06M)[dec15-jan-16]
8. Define the asymptotic notations used for best case average case and worst case(08M)[dec16-jan-17]
9. Write an algorithm for finding maximum element of an array , perform best , worst and average case complexity with appropriate order notations.(06M)[dec15-jan-16]
10. Write an algorithm to find mean and variance of an array perform best, worst and average
11. Explain the various criteria used for analyzing algorithms.(6m)[June 2013]
12. List the properties of various asymptotic notations.(06M)[dec14-jan-15]
13. Explain the necessary steps for analyzing the efficiency of recursive algorithms.(6m)[June 2012]
14. Write short notes on algorithm visualization.(06M)[dec11-jan-12]
15. Describe briefly the notions of complexity of an algorithm.(06M)[dec11-jan-12]
16. What is pseudo-code? Explain with an example(6m)[June 2010]
17. 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 number of inputs)(8m)[June

2009]

## MODULE -2

**Divide and Conquer**: General method, Binary search, Recurrence equation for divide and conquer, Finding the maximum and minimum **(T2:3.1, 3.3, 3.4)**, Merge sort, Quick sort **(T1:4.1, 4.2)**, Strassen’s matrix multiplication **(T2:3.8)**, Advantages and Disadvantages of divide and conquer. **Decrease and Conquer Approach:** Topological Sort. **(T1:5.3)**

**Questions:**

1. Explain concept of divide and conquer. Write merge sort algorithm. (08 Marks June/July 2017)
2. Write a recursive algorithm for binary search and also bring out its efficiency. (08 Marks June/July 2017)
3. Illustrate the tracing of quick sort algorithm for the following set of numbers: 25, 10, 72, 18, 40, 11, 64, 58, 32, 9 (08 Marks June/July 2017)
4. List out the advantages and disadvantages of divide and conquer method and illustrate the topological sorting for the following graph. (08 Marks June/July 2017)

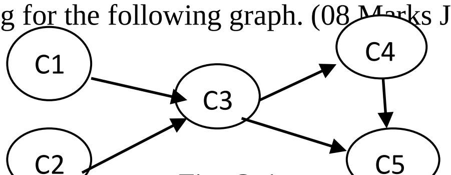

Fig: Q.4

5. The general form of divide and conquer recurrence relation and explain how you can solve it using Master's theorem. (6m)[June 2015]
6. Give a suitable sorting algorithm that uses divide and conquer a technique which divides problem size by considering values in the list. Analyse it for best and worst case efficiencies. (08M)[dec16-jan-17]
7. Give recursive binary search algorithm and write binary decision tree for the following n=14 elements(-15,-6,0,7,9,23,54,82,101,112,125,131,142,151) (08M)[dec16-jan-17]
8. Sort the following set of elements using Quick Sort. (24,8,71, 4,23,6)(6m)[June 2013]
9. Give a detailed note on divide and conquer techniques.(06M)[dec14-jan-15]
10. Write an algorithm for searching an element using binary search Method, Give an example.(6m)[June 2012]
11. Compare and contrast BFS and DFS.(6m)[June 2014]
12. Explain the merge sort.(08M)[dec13-jan-14]
13. Explain the method of finding the minimum spanning tree for a connected graph using Prim's algorithm.(6m)[June 2011]
14. Discuss the 0/1 knapsack problem(08M)[dec10-jan-11]

## MODULE -3

**Greedy Method:** General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines **(T2:4.1, 4.3, 4.5). Minimum cost spanning trees:** Prim’s Algorithm, Kruskal’s Algorithm **(T1:9.1, 9.2)**. **Single source shortest paths:** Dijkstra's Algorithm **(T1:9.3)**. **Optimal Tree problem:** Huffman Trees and Codes **(T1:9.4)**. **Transform and Conquer Approach:** Heaps and Heap Sort **(T1:6.4)**.

**Questions:**

1. Explain Greedy criterion. Write a Prim's algorithm to find minimum cost spanning tree. (08 Marks June/July 2017)
2. Sort the given list of numbers using heap sort: 2, 9,7,6,5,8 (08 Marks June/July 2017)
3. Write an algorithm to find single source shortest path. (08 Marks June/July 2017)
4. Construct a Huffman tree and resulting code word for the following:

| Character | A | B | C | D | - |
|---|---|---|---|---|---|
| Probability | 0.35 | 0.1 | 0.2 | 0.2 | 0.15 |

Encode the words DAD and ADD. (08 Marks June/July 2017)

5. Describe the travelling salesman problem and discuss how to solve it using dynamic programming(6m)[June 2015]
6. Discuss the use of Greedy method in solving Knapsack problem and subset-sum Programming problem.(08M)[dec16-jan-17]
7. Write an algorithm to sort a set of " M "numbers using insertion sort.(06M)[dec14-jan-15]
8. How will you find the shortest path between two given vertices using Dijkstra's algorithm? (6m)[June 2014]
9. Explain multistage graphs and give it example.(6m)[June 2011]

## MODULE -4

**Dynamic Programming:** General method with Examples, Multistage Graphs (T2:5.1,5.2). **Transitive Closure:** Warshall's algorithm, **All Pairs shortest Paths:** Floyd's algorithm, Optimal Binary Search trees, Knapsack problem **((T1:8.2, 8.3, 8.4), Bellman**-ford algorithm **(T2:5.4),** Travelling Sales Person Problem **(T2:5.9)**, Reliability design**.(T2:5.8).**

**Questions:**

1. Explain the concept of dynamic programming, with example. (08 Marks June/July 2017)
2. Trace the following graph using Warshall's algorithm. (08 Marks June/July 2017)

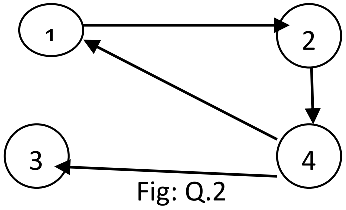

Fig: Q.2

3. Explain multistage graphs with example. Write multistage graph algorithm to forward approach. (08 Marks June/July 2017)
4. Solve the following instance of Knapsack problem using dynamic programming. Kanpsack capacity is 5. (08 Marks June/July 2017)

| Item | Weight | Value |
|---|---|---|
| 1 | 2 | $12 |

| | | |
|---|---|---|
| 2 | 1 | $10 |
| 3 | 3 | $20 |
| 4 | 2 | $15 |

5. What are the factors that influence the efficiency of the backtracking algorithm? (08M)[dec16-jan-17]
6. Define Branch-and-Bound method.(6m)[June 2015]
7. Find transitive closure for the graph shown below using dynamic programming.(08M)[dec14-jan-15]
8. Solve the travelling sales person problem using dynamic programming (6m)[June 2015]
9. Obtain the optimal solution for the given assignment problem as a matrix shown below using branch and bound method (08M)[dec12-jan-13]
10. Find the optimal solution for the following instance of knapsack problem using dynamic programming Capacity W=5(6m)[June 2010]
11. Apply the branch and bound algorithm to solve the travelling salesman problem (08M)[dec09-jan-10]

## MODULE-5

**Backtracking:** General method **(T2:7.1)**, N-Queens problem **(T1:12.1)**, Sum of subsets problem **(T1:12.1)**, Graph coloring **(T2:7.4)**, Hamiltonian cycles **(T2:7.5)**. **Branch and Bound:** Assignment Problem, Travelling Sales Person problem **(T1:12.2)**, **0/1 Knapsack problem (T2:8.2, T1:12.2):** LC Branch and Bound solution **(T2:8.2)**, FIFO Branch and Bound solution **(T2:8.2)**. **NP-Complete and NP-Hard problems:** Basic concepts, non-deterministic algorithms, P, NP, NP-Complete, and NP-Hard classes **(T2:11.1)**.

**Questions:**

1. Explain backtracking concept. Illustrate N queens problem using backtracking to solve 4-Queens problem. (08 Marks June/July 2017)
2. What are the searching techniques that are commonly used in Branch-and-Bound method. (08M)[dec16-jan-17]
3. Solve subset sum problem for the following example, S={3,5,6,7} and d=15.construct a state space tree. (08 Marks June/July 2017)
4. Explain the concept of brach and bound and solve assignment problem for the following and obtain optimal solution. (08 Marks June/July 2017)

| | | Job1 | Job2 | Job3 | Job4 |
|---|---|---|---|---|---|
| Person | a | 9 | 2 | 7 | 8 |
| | b | 6 | 4 | 3 | 7 |
| | c | 5 | 8 | 1 | 8 |
| | d | 7 | 6 | 9 | 4 |

5. The searching techniques that are commonly used in Branch-and-Bound method are:i. FIFO ii. LIFO iii. LC iv. Heuristic search(6m)[June 2015]
6. State 8 – Queens problem.(08M)[dec115-jan-16]
7. Explain LC Branch and Bound and FIFO branch and bound. (08 Marks June/July 2017)

8. The problem is to place eight queens on a 8 x 8 chessboard so that no two queen “attack” that is, so that no two of them are on the same row, column or on the diagonal.(6m)[June 2014]
9. State Sum of Subsets problem.(6m)[June 2013]
10. Given n distinct positive numbers usually called as weights , the problem calls for finding all the combinations of these numbers whose sums are m.(08M)[dec12-jan-13]
11. State m – colorability decision problem.(6m)[June 2009]
12. Let G be a graph and m be a given positive integer. We want to discover whether the nodes of G can be colored in such a way that no two adjacent nodes have the same color yet only mcolors are used.(6m)[June 2010]
13. Define chromatic number of the graph.(08M)[dec09-jan-10]
14. Define a planar graph.(6m)[June 2009]
15. What are NP- hard and Np-complete problems?(08M)[dec08-jan-09]
16. What is a decision problem? (6m)[June 2008]
17. What is maxclique problem? (08M)[dec07-jan-8]
18. what is approximate solution? (08M)[dec07-jan-08]