**Algorithm Design and Analysis**

UNIT – I Asymptotic notations for time and space complexity, Big-Oh notation, Θ notation, Ω notation, the little-oh notation, the little-omega notation, Recurrence relations: iteration method, recursion tree method, substitution method, master method (with proof), subtract and conquer master method(with proof), Data Structures for Disjoint Sets, Medians and Order statistics. Complexity analysis, Insertion sort, Merge Sort, Quick sort. Strassen’s algorithm for Matrix Multiplications.

UNIT – II Dynamic Programming: Ingredients of Dynamic Programming, emphasis on optimal substructure , overlapping substructures, memorization. Matrix Chain Multiplication, Longest common subsequence and optimal binary search trees problems, 0-1 knapsack problem, Binomial coefficient computation through dynamic programming. Floyd Warshall algorithm.

UNIT – III Greedy Algorithms: Elements of Greedy strategy, overview of local and global optima, matroid, Activity selection problem, Fractional Knapsack problem, Huffman Codes, A task scheduling problem. Minimum Spanning Trees: Kruskal’s and Prim’s Algorithm, Single source shortest path: Dijkstra’s and Bellman Ford Algorithm(with proof of correctness of algorithms).

UNIT – IV String matching: The naïve String Matching algorithm, The Rabin-Karp Algorithm, String Matching with finite automata, The Knuth-Morris Pratt algorithm. NP-Complete Problem: Polynomial-time verification, NP-Completeness and Reducibility, NP-Completeness Proof, NP –hard ,Case study of NP-Complete problems (vertex cover problem, clique problem).

**PREVIOUS YEAR QUESTIONS**

**UNIT 1**

* Define
  + Asymptotic Notation
  + Substitution Method, Recursion Tree and Iteration Method
  + Big oh, Small Oh, Big omega, Small Omega Notations
  + Subtract and Conquer Master Theorem
* Prove 3n2 + 2n2 = **O**(n2) and that 3n can NOT be expressed as **O**(2n)
* Solve
  + T(n) = T(n/3)+T(2n/3)+n [Using Recurrence Tree]
  + T(n) = 4T(n/2)+n3 [Using Master Method]
  + T(n) = 2T(n/2)+n [Using Substitution Method]
  + T(n) = 4T(n/2)+n2 [Using Recurrence Relations]
  + T(n) = 1 if n=1

= 2T(n/2)+n if n>1 [Using Iteration Method]

* If N guests are attending a party and each guest shakes hand with other only once. How many handshakes will take place? Also write a recursive definition and Algorithm.

[Hint : treat the number of guests as Vertices and draw a complete graph, then the total handshakes will be the number of edges of the complete graph, answer should be N(N-1)/2 ]

* Write an algorithm for
  + Merge Sort
  + Insertion Sort

and find its Best, Worst and Average Case Complexities

* Explain Quick Sort and Sort the following Array Using Quick Sort (14, 15,25,28,30,32,35,40)
* Sort the Array Using Merge Sort
  + (13,19,9,5,12,8,21,11,56,19)
  + (50,10,20,30,15,70,35,55)
* Explain Data Structures for disjoint sets and its Applications
* State and Prove Master Theorem, also give an example.
* Explain Strassen’s Matrix Multiplication and give its Time Complexity.
* Numerical on Multiplication Of Matrices using Strassen’s Algorithm.

**UNIT – 2**

* What is Dynamic Programming Paradigm? Explain its characteristics.
* Define Overlapping Sub-problem, Optimal Substructure and Memoization with an Example.
* DP Vs Divide and Conquer
* Advantages of OBST over BST.
* Explain 0-1 Knapsack Problem.
* Write Floyd Warshall Algorithm and its Time Complexity, solve an example using the same.
* Find Optimal Parenthesis of a matrix chain product whose sequence of diemensions are <4,10,3,12,20,7>
* Determine LCS of
  + <10010101> and <010110110>
  + <ABCBDAB>and<BDCABA>

**UNIT – 3**

* Difference between DP and Greedy.
* Define elements of Greedy Strategy.
* What is Matroid?
* Define Fractional Knapsack Problem.
* With the help of an Example differentiate between
  + Dijkstra and Bellman Ford Algorithm
  + Prim’s and Kruskal’s Algorithm
* Write and Explain
  + Huffman’s Algorithm
  + Dijkstra Algorithm
  + Bellman Ford Algorithm
  + Prim’s Algorithm
  + Kruskal’s Algorithm
* Generate Huffman Tree for :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ques  No. | Character | A | B | C | D | E | F |
| 1 | Frequency | 50 | 35 | 50 | 10 | 40 | 20 |
| 2 | Frequency | 45 | 15 | 5 | 25 | 10 | - |
| 3 | Frequency | 2 | 3 | 5 | 8 | 13 | 21 |

* Find the optimal schedule for the following jobs with profit (p1,p2,p3,p4,p5,p6) = (3,5,17,20,6,10) and deadlines (d1,d2,d3,d4,d5,d6) = (1,3,3,4,1,2)
* Consider 5 items

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item | I1 | I2 | I3 | I4 | I5 |
| Weight | 5 | 10 | 20 | 30 | 40 |
| Value | 30 | 20 | 100 | 90 | 160 |

The capacity of Knapsack W = 60. Find the solution to Fractional Knapsack Problem.

**UNIT – 4**

* Write Algorithm and Define with Example
  + KMP Algorithm
  + String Matching with Finite Automata
  + Rabin Karp Algorithm
* Define NP, NP hard and NP Complete Problems Breifly
* Explain NP Completeness Reduction with Example
* Write and Explain 3 examples of NPC Problems