|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Mod. No.*** | | ***Module Name*** | |  | | ***Topics to be covered*** | | | | ***No. of lectures*** | | ***Lecture serial nos.*** | |
| ***1.*** | | ***Introduction*** | | ***⚫ ⚫ ⚫***  ***⚫ ⚫ ⚫*** | | ***Concepts in algorithm analysis & design***  ***- motivation***  ***Complexity of an algorithm (Space and time Complexity)***  ***Analysis of time complexity of Insertion Sort by step count method***  ***Growth of functions***  ***Asymptotic notations (O, Θ)***  ***Solving recurrences (Iterative/Substitution/ Recurrence Tree/Master theorem/Change of variable method)*** | | | | ***8*** | | ***1-8*** | |
|  | |  | |  | | ***Tutorials / Activity*** | | | |  | |  | |
| ***2.*** | ***Divide and Conquer Method, Heap*** | | ***⚫ ⚫***  ***⚫ ⚫ ⚫ ⚫ ⚫ ⚫ ⚫*** | | ***Structure of Divide-and-Conquer algorithm***  ***Analysis of divide-and-conquer run time recurrence relations of***  ***Binary Search***  ***Merge Sort***  ***Quick Sort***  ***Randomized Quick Sort***  ***Building a heap***  ***Heap sort algorithm***  ***Priority queue*** | | | | ***9*** | | ***9-17*** | |
|  |  | |  | | | |  |
|  | | ***Tutorials / Activity*** | | | |  | |  | |
| ***3.*** | ***Greedy Method*** | | ***⚫ ⚫ ⚫ ⚫*** | | ***Overview of Greedy paradigm Fractional knapsack problem Activity selection problem Huffman’s code*** | | | | ***4*** | | ***18-21*** | |
|  |  | |  | | ***Tutorials / Activity*** | | | |  | |  | |
| ***4.*** | ***Dynamic Programming*** | | ***⚫ ⚫***  ***⚫ ⚫*** | | ***Overview of Dynamic Programming paradigm***  ***Difference between Divide and Conquer and Dynamic Programming***  ***Matrix Chain Multiplication***  ***Longest Common Subsequence*** | | | | ***4*** | | ***22-25*** | |
|  |  | |  | | ***Tutorials / Activity*** | | | |  | |  | |
| ***5.*** | ***Graph Algorithms*** | | ***⚫ ⚫ ⚫***  ***⚫***  ***⚫ ⚫*** | | ***Dis-joint Set Data Structure Representation Of Graph Graph Traversals :: BFS DFS Single Source Shortest Path***   * ***Dijkstra’s Algorithm*** * ***Bellman-Ford Algorithm***   ***All Pair Shortest Path***  ***Floyd-Warshall Algorithm Minimum Cost Spanning Tree***   * ***Kruskal’s Algorithm*** * ***Prim’s Algorithm*** | | | | ***10*** | | ***26-35*** | |
|  |  | |  | | ***Tutorials / Activity*** | | | |  | |  | |
| ***6.*** | ***Computationa l Complexity*** | | ***⚫*** | | ***Complexity Classes: P, NP, NP-Hard and NP-Complete*** | | | | ***2*** | | ***36-37*** | |
|  | | ***Tutorials / Activity*** | | | |