VIVA - VOCE

UNIT - I

Q.1. Define the term mathematical induction.

Ans. Mathematical induction is a method to prove sequence of steps.

Q.2. What are the phases of mathematical induction?

Ans. There are two phases of mathematical induction :

(1) Basis phase.

(2) Induction / Hypothesis phase.

Q.3. What is recurrence?

Ans. An equation of inequality that describes a function in terms of its value on smaller input is known as recurrence.

Q.4. What is recurrence equation?

Ans. If the algorithm call itself within its body, then the algorithm is called as recursive algorithm and complexity equation of algorithm is called as recurrence equation.

Q.5. How many methods are there for solving recurrence?

Ans. There are five methods for solving recurrence :

(1) Substitution method

(2) Iteration method

(3) Recursion tree

(4) Changing variables

(5) Master's theorem

Q.6. What is homogeneous recurrence?

Ans. If in the recurrence relation there are similar types of terms with same base "b", then the recurrence is called as homogeneous recurrence.

Q.7. What is non-homogeneous recurrence?

Ans. If in the recurrence equation there are multiple types of terms, then it is called as non-homogeneous recurrence.

Q.8. What is logarithmic recurrence?

Ans. If in a algorithm a index variable is operated by either multiplication or division operator, then the algorithm will generate recurrence called as algorithm recurrence.

UNIT - II

Q.1. What is Asymptotic notation?

Ans. Asymptotic notations describe the behavior of time or space complexity for large instance characteristics. A way of comparing functions that ignores constant factors and small input size is known as asymptotic notations.

Q.2. How many types of asymptotic notations are there?

Ans. There are five types of asymptotic notations :

(1) Big Oh (O)

(2) Big Theta (θ)

(3) Big omega (Ω)

(4) Little Oh (o)

(5) Little omega (ω)

Q.3. Define Big 'Oh'.

Ans. It is defined as function $f(n) = O(g(n))$ if there exist positive constants c and n_o such that $f(n) \leq c^* g(n)$ for all n, $n \geq n_o$.

Q.4. Define Omega.

Ans. It is defined as function $f(n) = \Omega(g(n))$ if there exist positive constants c and n_o such that $f(n) \geq c^* g(n)$ for all n, $n \geq n_o$.

Q.5. Define Theta.

Ans. It is defined as function $f(n) = \theta(g(n))$ if there exist positive constants C_1, C_2 and n_o such that $C_1 g(n) \leq f(n) \leq C_2 g(n)$ for all n, $n \geq n_o$.

Q.6. What are the properties of asymptotic notations?

Ans. The properties of asymptotic notations are as follows :

(1) Reflexivity

(2) Symmetry

(3) Transpose symmetry

(4) Transitivity

Q.7. What is algorithm?

Ans. An algorithm is a finite set of instructions that accomplishes a particular task.

Q.8. What criteria should all algorithms satisfy?

Ans. Input, Output, Definiteness, Finiteness, Effectiveness.

Q.9. What do you mean by analysis of algorithm?

Ans. Analysis of algorithm depends upon the individual instruction along with control structure. Analysis of algorithm focuses on time and space complexity.

Q.10. List the control structures for analyzing algorithms.

Ans. Control structures for analyzing algorithms are as follows :

(1) Sequencing

(2) If-then-else

(3) For loop

(4) While loop

(5) Recursion

Q.11. Define time and space complexity.

Ans. **(i) Time complexity :**

Time complexity is a function of input size 'n' referred to as the amount of time required by an algorithm to run to completion.

(ii) Space complexity :

Space complexity refers to the amount of memory required by an algorithm to run to completion.

Q.12. What is worst, best and average case complexity?

Ans. **(i) Worst case complexity :**

Worst case time complexity is the function defined by the maximum amount of time needed by an algorithm for an input of size 'n'.

(ii) Average case complexity :

Average case time complexity is the execution of an algorithm having typical input data of size 'n'.

(iii) Best case complexity :

Best case time complexity is the minimum amount of time that an algorithm requires for an input of size 'n'.

Q.13. What is amortized analysis?

Ans. Amortized analysis is the time required to perform a sequence of data structure operations is averaged over all the operations performed.

Q.14. List the methods of amortized analysis.

Ans. The methods of amortized analysis are as follows :

(1) Aggregate method

(2) Potential method

(3) Accounting method

Q.15. Give the applications of amortized analysis.

Ans. The various applications of amortized analysis are as follows :

(1) Stack operation.

(2) Binary counter.

Q.16. What is sorting network?

Ans. Sorting network is a comparison network for which the output sequence is monotonically increasing (i.e. $b_1 \leq b_2 \leq ....b_n$) for every input.

Q.17. What is comparison network?

Ans. Comparison network is a set of comparators interconnected by wires. It has two inputs x and y, and two outputs x' and y' , Where,

$x' = \min(x, y)$

$y' = \max(x, y)$

Q.18. Define bitonic sorter.

Ans. A sequence is bitonic if it montonically increases and then monotonically decreases or can be circularly shifted to become monotonically increase and then monotonically decrease.

Q.19. What is Fibonacci heap?

Ans. A collection of min-heap ordered trees is called as Fibonacci heap.

Q.20. Explain disjoint set representation.

Ans. A disjoint set data structure maintains a collection $S = \{s_1, s_2, ... s_k\}$ of dynamic sets. Each set is identified by a representative, which is some member of the set.

If we have sets S_x and S_y, $x \neq y$ such that $S_x = \{3, 4, 5, 6, 7\}$ and $S_y = \{1, 2\}$ these sets are called disjoint set as there is no element which is common in both sets.

Q.21. List the disjoint set operations.

Ans. The disjoint set operations are as follows :

(1) MAKE-SET (x)

(2) UNION (x, y)

(3) FIND-SET (x)

Q.22. Explain red black trees.

Ans. Red black tree is a binary search tree where each node is assigned a color, where the coloring scheme helps is maintaining height as $0(\log n)$.

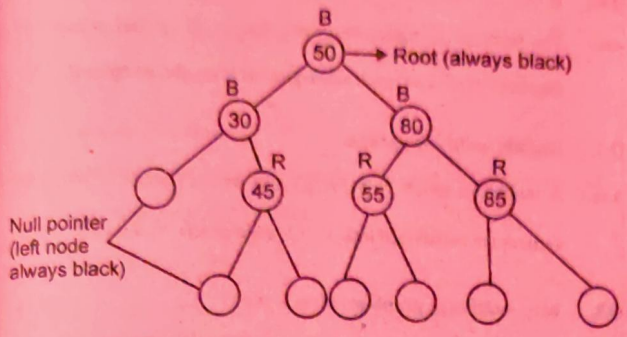

Fig. Simple red black tree

UNIT - III

Q.1. What is divide and conquer approach?

Ans. Divide and conquer is a recursive approach. This approach works as follows :

(1) Divide problem into sub problems of the same kind of problem.

(2) For sub problems that are really small solve them directly. Else solve them recursively.

(3) Combine subproblem solutions to solve the whole things.

Q.2. What is binary search?

Ans. In binary search tree every nodes value is greater than its left subtree and lesser than its right subtree.

Q.3. What is the complexity of binary search?

Ans. Complexity of binary search is $O(\log_2 n)$.

Q.4. What is quick sort?

Ans. In quick sort, the division is made into two subarrays so that the sorted subarrays do not need to be merged later.

Q.5. What is the complexity of quick sort?

Ans. Complexity of quick sort is $O(n \log n)$.

Q.6. What is merge sort?

Ans. In merge sort, the file a [1 : n] is divided at its midpoint into subarrays which were independently sorted and later merged.

Q.7. Merge sort is based on which approach?

Ans. Merge sort is based on divide and conquer approach.

Q.8. Matrix multiplication operation is associative or commutative?

Ans. Matrix multiplication operation is associative.

Q.9. Greedy algorithms are used to solve which problems?

Ans. Greedy algorithms are used to solve optimization problem.

Q.10. What is knapsack problem?

Ans. Knapsack is a block having fixed capacity. The main objective is to achieve maximum profit.

Q.11. What are the types of knapsack problem?

Ans. There are two types knapsack problem :

(1) Fractional knapsack.

(2) 0/1 knapsack.

Q.12. What is Huffman code?

Ans. Huffman code is a technique for compressing data.

Q.13. What are the methods of Huffman code?

Ans. The two methods of Huffman code are as follows :

(1) Fixed length code.

(2) Prefix code.

Q.14. What are the applications of greedy algorithm?

Ans. Applications of greedy algorithm are as follows :

(1) Ordering projects by deadlines.

(2) Compressed video using Huffman coding.

(3) Packet scheduling.

(4) Packet routing.

(5) Removing noisy data for training neural network.

Q.15. What is activity scheduling problem?

Ans. In activity scheduling problem, we have to determine the optimum numbers of activities that are scheduled to the resources.

Q.16. What is minimum cost spanning tree?

Ans. Given a connected undirected graph G = (v, e) and a length junction w such that w(e) is the (positive) length of edge e, find a subset of the edges that connects all the vertices together and has minimum total length.

Q.17. What are the various methods to solve MST?

Ans. There are two methods to solve MST :

(1) Kruskal

(2) Prim's

Q.18. Give applications of MST.

Ans. Applications of MST are as follows :

(1) Designing physical networks.

(2) Cluster analysis.

(3) Approximate solutions to NP-hard problems.

Q.19. What is the difference between Prims and Kruskal algorithm?

Ans.

Sr. No.	Kruskal	Prims
(1)	Edge is selected on minimum value.	Edge is selected on minimum value and connectivity.
(2)	Graph generated will be isolated.	Graph generated will be connected.

Q.20. What is optimal merge pattern?

Ans. Optimal merge pattern consists of two steps :

(1) Merge the given file optimally after the greedy choice to obtain optimal pattern.

(2) At each step merge the two smallest size file together.

UNIT - IV

Q.1. What do we understand by dynamic programming?

Ans. Dynamic programming like the divide and conquer method which solves problems by combining the solutions to subproblems.

Q.2. What is the principle of dynamic programming?

Ans. Dynamic programming is a stage-wise search method suitable for optimization problem whose solutions may be viewed as the result of a sequence of decisions.

Q.3. Which approach is used in dynamic programming?

Ans. Bottom up approach is used in dynamic programming.

Q.4. What is the difference between greedy method and dynamic programming?

Ans. In the greedy method only one decision sequence is ever generated, whereas in dynamic programming many decision sequences may be gene-rated.

Q.5. Write any one feature of dynamic programming.

Ans. Optimal solutions to subproblems are retained so as to avoid recomputing their values.

Q.6. What is the principle of optimality ?

Ans. The principle of optimality states that in an optimal sequence of decisions or choices, each subsequences must also be optimal.

Q.7. Explain multistage graph.

Ans. A multistage graph G = (V, E) is a directed graph in which the vertices are partitioned into $K \geq 2$ disjoint sets $V_i, 1 \leq i \leq k$.

Q.8. Why multistage graph problem is used?

Ans. Multistage graph problem is used to find minimum cost path from one node to other.

Q.9. Define optimal solution.

Ans. A feasible solution that either maximizes or minimizes a given objective function, this solution is called an optimal solution.

Q.10. What are the methods of single source shortest path?

Ans. The methods of single source shortest path are as follows :

(1) Dijkstra's algorithm.

(2) Bellman Ford algorithm.

Q.11. What is optimal binary search tree?

Ans. Optimal binary search tree is a tree used to design a structure such that total cost of the tree is minimum.

Q.12. What is the main objective of traveling salesman problem?

Ans. The main objective of traveling salesman problem is to find shortest path from a source vertex to all the vertices and traveling back to source.

Q.13. What is the complexity of traveling salesman problem?

Ans. The complexity of traveling salesman problem is $O(n^2 2^n)$

Q.14. Define longest common subsequence.

Ans. LCS problem is defined as follows :

Given two sequences $x = \{x_1, x_2, \ldots, x_m\}$ and $y = \{y_1, \ldots, y_n\}$ find the longest subsequence $z = \{z_1, \ldots, z_k\}$ that is common to x and y.

Q.15. Give an applications of longest common subsequence.

Ans. The application of longest common subsequence is DNA matching.

UNIT - V

Q.1. What is traversal methods?

Ans. The techniques which involves examining every node in the given data object instance is referred to as traversal methods.

Q.2. What is search methods?

Ans. The techniques which are applicable to graphs and may not examine all vertices referred to as search methods.

Q.3. What is traversal?

Ans. When the search necessarily involves the examination of every vertex in the object being searched, it is called a traversal.

Q.4. What is BFS?

Ans. In BFS, a vertex is said to have been explored by an algorithm when the algorithm has visited all vertices adjacent from it.

Q.5. What is DFS?

Ans. In DFS, the exploration of a vertex V is suspended as soon as a new vertex is reached.

Q.6. How is a graph represented?

Ans. Graph is represented as :

(1) Adjacency list.

(2) Adjacency matrix.

Q.7. How long does adjacency list representation of directed graph take to compute the out degree of every vertex?

Ans. The adjacency list representation of a graph the out and in degree of every node can easily be computed in O (E + V) time.

Q.8. How many elementary algorithms are there for tree searching?

Ans. There are two elementary algorithms for tree searching :

(1) Breadth first search (BFS)

(2) Depth first search (DFS)

Q.9. What is the running time of BFS?

Ans. The running time of BFS is O (V+ E)

Q.10. What is the running time of DFS?

Ans. The running time of DFS is θ (V + E).

Q.11. How the edges are classified in DFS?

Ans. DFS can be used to classify the edges of input graph G = (V , E) as

(1) Tree edge

(2) Back edge.

(3) Forward edge.

(4) cross edge.

Q.12. Which algorithm efficiently computes the strongly connected components of a directed graph?

Ans. Kosaraju's algorithm efficiently computes the strongly connected components of a directed graph.

Q.13. What is backtracking?

Ans. Backtracking is a methodical way of trying out various sequences of decisions, until we find that works.

Q.14. What is explict constraint?

Ans. Explict constraint are rules which restrict each vector element to be chosen from the given set.

Q.15. What is implict constraint?

Ans. Implict constraints are rules, which determine which of the tuples in the solution space actually satisfy the criterion function.

Q.16. What is an example of backtracking algorithm?

Ans. Recursive maze algorithm is one of the good example for backtracking algorithm.

Q.17. Define N-Queen problem.

Ans. N-Queen problem is to place N-Queens in such a manner on an n * n chessboard such that no two queens attack each other by being in the same row, column or diagonal.

Q.18. What is sum of subset problem?

Ans. In sum of subset problem we have to find a subset S of the given set $S = \{S_1, S_2 .... S_n\}$ where the elements of the set are n positive integers in such a manner that $S' \in S$ and sum of the elements of subset 'S' is equal to some positive integer "x".

Q.19. Define graph coloring.

Ans. Graph coloring is an assignment of colors to certain objects in a graph such that no two colors should be of same color at the adjacent node.

Q.20. Define Hamiltonian circuit.

Ans. Hamiltonian circuit is a path in an undirected graph which visits each vertex exactly once.

Q.21. Define approximation algorithm.

Ans. An approximation algorithm for P is an algorithm that generates approximate solutions for P.

Q.22. What are the example of approximate algorithm?

Ans. The examples of approximate algorithm are as follows :

(1) Vertex cover problem.

(2) Traveling salesman problem.

UNIT - VI

Q.1. What are the three classes of problems?

Ans. The three classes of problems are P, NP and NPC.

Q.2. What is class NP?

Ans. The class which consists of problems that are 'verifiable' in polynomial time is known as class NP.

Q.3. Define class P.

Ans. P is the set of decision problems with a yes-no answer that is polynomial bounded.

Q.4. What is NP-completeness?

Ans. NP-completeness deals with the finding of an efficient algorithm for certain problem.

Q.5. What is polynomial time computation?

Ans. If there exist a polynomial-time algorithm A that when given any input $x \in \{0,1\}^*$ produces as output $f(x)$, then function $f: \{0,1\}^* \rightarrow \{0,1\}^*$ is polynomial-time computable.

Q.6. What is decision problem?

Ans. The problems under the class P have the single bit output which shows 0 or 1 i.e., the answer for the problem is either zero or one are decision problems.

Q.7. Define complexity class NP.

Ans. The complexity class NP includes the complexity class P but allows for the languages that are not present in P.

Q.8. Define cooks statement.

Ans. Cooks statement is as follows :

A problem is in NP if and only if it can be reduced to circuit SAT.

Q.9. Define clique.

Ans. A clique in a graph G is a subset S of vertices such that for each V_1 and V_2 in S, with $V_1 \neq V_2$ (V_1, V_2) is an edge. This means there exists an edge between every distinct pair of vertices in S.

Q.10. Define randomized algorithm.

Ans. Randomized algorithm are those that achieves an approximation ratio of P (n).

Q.11. How many methods are of randomized algorithm?

Ans. The methods of randomized algorithm are as follows :

(1) Numerical algorithm.

(2) Monte-casio algorithm.

(3) Las vegas algorithm.

(4) Sherwood algorithm.

Q.12. What are the strengths of computational geometry?

Ans. Strength of computational geometry are as follows :

(1) Development of geometric tools.

(2) Emphasis on provable efficiency.

(3) Emphasis on correctness.

(4) Linkage to discrete combinational geometry.

Q.13. What are the limitations of computational geometry?

Ans. Limitations of computational geometry are as follows :

(1) Emphasis on discrete geometry.

(2) Emphasis on flat objects.

(3) Emphasis on low-dimensional space.