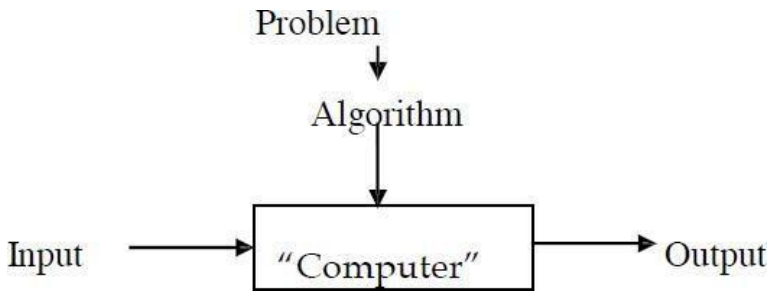
UNIT I

PART A QUESTIONS

1. What is an algorithm? (UQ April 2012 & APRIL 2013/2015/2017)

An algorithm is a sequence of unambiguous instructions for solving a problem, i.e., for obtaining a required output for any legitimate input in finite amount of time.

2. Give the diagram representation of Notion of algorithm.



3. What is the formula used in Euclid's algorithm for finding the greatest common divisor of two numbers?(Ap 2016)

Euclid's algorithm is based on repeatedly applying the equality

Gcd(m,n)=gcd(n,m mod n)

until m mod n is equal to 0, since gcd(m,0)=m.

4. What are the three different algorithms used to find the gcd of two numbers?(AP 2017)

The three algorithms used to find the gcd of two numbers are .

- Euclid's algorithm
- Consecutive integer checking algorithm
- Middle school procedure

5. What are the fundamental steps involved in algorithmic problem solving?

The fundamental steps are

1. Understanding the problem.
2. Ascertain the capabilities of computational device
3. Choose between exact and approximate problem solving.
4. Decide on appropriate data structures.
5. Algorithm design techniques
6. Methods for specifying the algorithm
7. Proving an algorithms correctness
8. Analyzing an algorithm.
9. Coding an algorithm

8. What is pseudo code?

A pseudo code is a mixture of a natural language and programming language constructs to specify an algorithm. A pseudo code is more precise than a natural language and its usage often yields more concise algorithm descriptions.

9. What are the types of algorithm efficiencies?

The two types of algorithm efficiencies are

Time efficiency: indicates how fast the algorithm runs.

Space efficiency: indicates how much extra memory the algorithm needs

10. What are the steps involved in the analysis framework?

The various steps are as follows

- Measuring the input's size.
- Units for measuring running time.
- Orders of growth.
- Worst case, best case and average case efficiencies

11. What is worst-case efficiency?(UQ APRIL 2015)

The worst-case efficiency of an algorithm is its efficiency for the worst-case input of size n,

which is an input or inputs of size n for which the algorithm runs the longest among all possible inputs of that size.

18. What is best-case efficiency? (UQ APRIL 2013/2015)

The best-case efficiency of an algorithm is its efficiency for the best-case input of size n, which is an input or inputs for which the algorithm runs the fastest among all possible inputs of that size.

19. What is average case efficiency?

The average case efficiency of an algorithm is its efficiency for an average case input of size n. It provides information about an algorithm behavior on a "typical" or "random" input.

20. What is amortized efficiency?

In some situations a single operation can be expensive, but the total time for the entire sequence of n such operations is always significantly better that the worst case efficiency of that single operation multiplied by n. this is called amortized efficiency.

21. Define notation?

A function t(n) is said to be in (g(n)), denoted by t(n) (g(n)), if t(n) is bounded below by some constant multiple of g(n) for all large n, i.e., if there exists some positive constant c and some nonnegative integer n0 such that

T(n) >= cg(n) for all n >=. n0

22. What is the recurrence relation to find out the number of multiplications and the initial condition for finding the n-th factorial number?

The recurrence relation and initial condition for the number of multiplications is

M(n)=M(n-1)+1 for n>0

M(0)=0

23. Write the general plan for analyzing the efficiency for recursive algorithms.

The various steps include

- Decide on a parameter indicating input's size.
- Identify the algorithms basic operation. .

- Check whether the number of times the basic operation is executed
- Depends on size of input. If it depends on some additional property the worst, average and best-case efficiencies have to be investigated separately.
- Set up a recurrence relation with the appropriate initial condition, for the number of times the basic operation is executed.
- Solve the recurrence or at least ascertain the orders of growth of its solution.

24. What is time complexity?

The time complexity of an algorithm is the amount of computer time it needs to run to completion.

25. What is space complexity?

The space complexity of an algorithm is the amount of memory it needs to run to completion.

PART B QUESTIONS

1. Discuss in detail about fundamentals of algorithmic problem solving?

2. Explain the important problem types in detail ***(APR 2015)***

3. Explain the necessary steps for analyzing the efficiency of recursive algorithms ***(APR 2017)***

4. Explain the general framework for analyzing the efficiency of algorithm. ***(Apr 2016)***

5. Write the asymptotic notations used for best case ,average case and worst case analysis of algorithms and Write an algorithm for finding maximum element of an array perform best , worst and average case complexity with appropriate order notations ***(APR 2017)***

6. Explain the method of solving recurrence equations with suitable example.

7. Explain the method of solving Non recursive equations with suitable examples

i)Describe the basic efficiency classes in detail.

ii) Write an algorithm for Fibonacci numbers generation and compute the following

a)How many times is the basic operation executed

b)What is the efficiency class of this algorithm

8.Compute complexity of element uniqueness problem, Tower of Hanoi, no.of digits in binary representation.(DEC 2016)

9.Evaluate the following recurrences completely

(i)T(n) = T(n/2) +1, where n=2k for all k≥0

(ii) T(n) = T(n/3) + T(2n/3) + cn, where 'c' is a constant and 'n' is the input size.

(iii) Solve the recurrence relation using substitution method

T(n)= { T(1) n=1

aT(n/b)+f(n) n>1 where a=5, b=4 and $f(n)=cn^2$

10. Analyze the following equalities are correct:

i) 5n2-6n=Θ(n2)

ii) n!=O(nn)

iii) iii)$n^3+10^6n^2=\Theta(n^3)$

iv) iv)$2n^22^n+n\ \log n=\Theta(n^22^n)$

UNIT II

PART-A

1. Define the divide and conquer method. ***(UQ APRIL'13 & APRIL'12/JUNE 2016)***

Given a function to compute on 'n' inputs the divide-and-conquer strategy suggests splitting the inputs including 'k' sub problems. The sub problems must be solved, andthen a method must be found to combine sub solutions into a solution of the whole. If thesub problems are still relatively large, then the divide-and conquer strategy can possiblybe reapplied.

2. Define control abstraction.

A control abstraction we mean a procedure whose flow of control is clear but whose primary operations are by other procedures whose precise meanings are left undefined.

3. What is the substitution method?

One of the methods for solving any such recurrence relation is called the substitution method.

4.* What is the binary search? *(UQ APRIL'12/APRIL 2015)

If 'q' is always chosen such that 'aq' is the middle element (that is, q=[(n+1)/2), then the resulting search algorithm is known as binary search.

5. What is the maximum and minimum problem?

The problem is to find the maximum and minimum items in a set of 'n' elements. Though this problem may look so simple as to be contrived, it allows us to demonstrate divide and-conquer in simple setting.

6.* What is the Quick sort? *(UQ Nov'10& APRIL'12)

The n numbers in quick sort, the division into sub arrays is made so that the sorted sub arrays do not need to be merged later.

7.* Is insertion sort better than the merge sort? *(UQ APRIL'13)

Insertion sort works exceedingly fast on arrays of less than 16 elements, though for large 'n' its computing time is O(n2).

8. Give the recurrence relation of divide-and-conquer?

The recurrence relation is

$T(n) = g(n)$

$T(n1) + T(n2) + ---- + T(nk) + f(n)$

9. What is meant by feasible solution?

Given n inputs and we are required to form a subset such that it satisfies some given constraints then such a subset is called feasible solution.

10. Define optimal solution?

A feasible solution either maximizes or minimizes the given objective function is called as optimal solution

11.* What is greedy method? *(UQ:Apr/May'14)

Greedy method is the most important design technique, which makes a choice that looks best at that moment. A given 'n' inputs are required us to obtain a subset that satisfies

some constraints that is the feasible solution. A greedy method suggests that one can device an algorithm that works in stages considering one input at a time.

12. What is the difference between quicksort and mergesort?

Both quicksort and mergesort use the divide-andconquer technique in which the given array is partitioned into subarrays and solved. The difference lies in the technique that the arrays are partitioned. For mergesort the arrays are partitioned according to their position and in quicksort they are partitioned according to the element values.

13. List out the 4 steps in Strassen's Method?

1. Divide the input matrices A and B into n/2 * n/2 submatrices, as in equation (1).

2.Using Θ(n2) scalar additions and subtractions, compute 14 n/2 * n/2 matrices A1, B1,A2, B2, A7, B7.

3. Recursively compute the seven matrix products Pi =AiBi for i =1, 2, 7.

4. Compute the desired submatrices r, s, t, u of the result matrix C by adding and/or subtracting various combinations of the Pi matrices, using only Θ(n2) scalar additions and subtractions.

14. Define mergesort.

Mergesort sorts a given array A[0..n-1] by dividing it into two halves a[0..(n/2)-1] and A[n/2..n-1] sorting each of them recursively and then merging the two smaller sorted arrays into a single sorted one.

15. What is brute force algorithm?

A straightforward approach, usually based directly on the problem‘s statement and definitions of the concepts involved.

16. What is exhaustive search?

A brute force solution to a problem involving search for an element with a special property, usually among combinatorial objects such as permutations, combinations, or subsets of a set.

17. Define the decrease and conquer method.

Decrease & conquer is a general algorithm design strategy based on exploiting the relationship between a solution to a given instance of a problem and a solution to a smaller instance of the same problem. The exploitation can be either top-down (recursive) or bottom-up (non-recursive).

18. Define the Transform-and-Conquer

The Transform and conquer technique is a way of solving problems by breaking them down into smaller sub problems, solving the smaller sub problems, and then combining the solutions to the sub problems to solve the original problem.

19. What is a heap ?

Heaps are tree-based data structures constrained by a heap property**.** Heaps are used in many famous algorithms such as Dijkstra's algorithm for finding the shortest path, the heap sort sorting algorithm, implementing priority queues, and more. Essentially, heaps are the data structure you want to use when you want to be able to access the maximum or minimum element very quickly.

20. Define heap sort.

Heap sort is a comparison-based sorting technique based on Binary Heap data structure. It is similar to the selection sort where we first find the minimum element and place the minimum element at the beginning. Repeat the same process for the remaining elements.

PART B- QUESTIONS

1. Solve using Brute force approach to evaluate and find whether the given string follows the specified pattern and return 0 or 1 accordingly. APRIL / MAY 2017

   Examples:

   1)Pattern "abba" input: "redblueredblue" should return 1

   2)Pattern "aaaa" input: "asdasdasdasd" should return 1

   3)Pattern "aabb" input: "xyzabcxyzabc" " should return 0

2. Explain in detail about Travelling Salesperson Problem using exhaustive search.
3. Explain Merge sort, and arrange the following numbers in increasing order using merge sort. (18, 29, 68, 32, 43,37, 87, 24, 47, 50)

4. Write the quick sort algorithm and explain it with an example. Derive the Worst case and average case time complexity.
5. Explain the method for performing multiplication of two large integers. Explain how divide conquer method can be used to solve the same. APRIL / MAY 2017
6. Explain closest pair problem and how will you solve it using divide and conquer technique. NOV/DEC 2019
6. Explain convex hull problem and how will you solve it using divide and conquer technique
7. Explain Multiplication of Large integers and Strassen's Matrix multiplication. APR/MAY 2018
8. Explain decrease and conquer method.
9. Discuss in brief about Transform-and-Conquer.
10. Explain String matching algorithm in detail
11. Apply quick sort algorithm to sort the list. E, X, A, M, P, L, E in alphabetical order.

UNIT III

PART A QUESTIONS

1 **Define dynamic programming**. ***(UQ APRIL'12/APRIL 2015)***

Dynamic programming is an algorithm design method that can be used when a solution to the problem is viewed as the result of sequence of decisions.

2 **What are the features of dynamic programming?**

Optimal solutions to sub problems are retained so as to avoid re-computing their values. Decision sequences containing subsequences that are sub optimal are not considered. It definitely gives the optimal solution always.

3 **Write the general procedure of dynamic programming. *(UQ APRIL'13)***

The development of dynamic programming algorithm can be broken into a sequence of 4 steps.

1. Characterize the structure of an optimal solution.

2. Recursively defines the value of the optimal solution.

3. Compute the value of an optimal solution in the bottom-up fashion.

4. Construct an optimal solution from the computed information.

4. Define principle of optimality.

It states that an optimal sequence of decisions has the property that whenever the initial stage or decisions must constitute an optimal sequence with regard to stage resulting from the first decision.

5. Give an example of dynamic programming and explain.

An example of dynamic programming is knapsack problem. The solution to the knapsack problem can be viewed as a result of sequence of decisions. We have to decide the value of xi for 1<i<n. First we make a decision on x1 and then on x2 and so on. An optimal sequence of decisions maximizes the object function.

6. Define 0/1 knapsack problem.

The solution to the knapsack problem can be viewed as a result of sequence of decisions. We have to decide the value of xi. xi is restricted to have the value 0 or 1 and by using the function knap(l, j, y) we can represent the problem as maximum pi xi subject to wixi < y where l - iteration, j - number of objects, y – capacity.

5. Write about traveling salesperson problem.

Let g = (V, E) be a directed. The tour of G is a directed simple cycle that includes every vertex in V. The cost of a tour is the sum of the cost of the edges on the tour. The traveling salesperson problem to find a tour of minimum cost.

6. Write some applications of traveling salesperson problem.

Routing a postal van to pick up mail from boxes located at n different sites. Using a robot arm to tighten the nuts on some piece of machinery on an assembly line. Production environment in which several commodities are manufactured on the same set of machines.

7. Give the time complexity and space complexity of traveling salesperson problem.

Time complexity is O (n2 2n).
Space complexity is O (n 2n).

8. **What is biconnected component? *(UQ Nov'12)***

Graph with no articulation point then it is called as biconnected component

9. What is articulation point?

When a node in a graph is deleted, the graph divided into two or more graph then the node is called articulation point.

10. Write about multistage graph with example UQ APRIL'13 & Nov'12,APRIL/MAY'14)

A multistage graph G = (V,E) is a directed graph in which the vertices are portioned intoK > = 2 disjoint sets Vi, 1 <= i<= k. In addition, if < u,v > is an edge in E, then u < = Viand V ∑ Vi+1 for some i, 1<= i < k.

11. Write about optimal merge pattern problem.

Two files x1and x2containing m & n records could be merged together to obtain one merged file. When more than 2 files are to be merged together. The merge can be accomplished by repeatedly merging the files in pairs. An optimal merge pattern tells which pair of files should be merged at each step. An optimal sequence is a least cost sequence.

12. Define warshall's algorithm?

Warshall's algorithm is an application of dynamic programming technique, which is used to find the transitive closure of a directed graph.

13. Define Floyd's algorithm?

Floyd's algorithm is an application, which is used to find all the pairs shortest paths problem. Floyd's algorithm is applicable to both directed and undire ted weighted graph, but they do not contain a cycle of a negative length

14. Define prim's algorithm.

Prim's algorithm is greedy and efficient algorithm m, which is used to find the minimum spanning tree of weighted connected graph

15. How efficient is prim's algorithm?

The efficiency of the prim's algorithm depends on data structure chosen for the graph.

16. List the advantage of Huffman's encoding?

Huffman's encoding is one of the most important file compression methods.1.It is simple

2. It is versatility

3. It provides optimal and minimum length encoding

17. Define Huffman trees?

A Huffman tree is binary tree that minimizes the weighted path length from the root tothe leaves containing a set of predefined weights.

PART B QUESTIONS

1. Discuss in detail about Floyd and warshalls algorithm.

2. Explain about optimal binary search tree with suitable examples.

3. Plan the knapsack arrangement with capacity (W=5) for the following situation. Use Dynamic programming algorithm.

Item	Weight	Value
1	2	$12
2	1	$10
3	3	$20

4. Let A= {l/119,m/96,c/247,g/283,h/72,f/77,k/92,j/19} be the letters and its frequency of distribution in a text file. Analyze a suitable Huffman coding to compress the data effectively.

5. Apply function OBST to compute w(i, j), r(i, j), and c(i, j), 0<=i<j<=4, for the identifier set ( ) 4 3 2 1 , , , a a a a = (cout, float, if, while) with p(1) = 1/20, p(2) =1/5, p(3) = 1/10, p(4) = 1/20, q(0) = 1/5, q(1) = 1/10, q(2) = 1/5, q(3) = 1/20,and q(4) 1/20. Using the r(i, j)'s, construct the optimal binary search tree.

6. Explain Multi stage graph in detail

7. Summarize Knapsack and memory functions problem in detail.

8. Discuss in detail about Dijkstra's algorithm.

9. Analyze the algorithm by applying the following keys and probabilities to obtain the **optimal binary tree.**

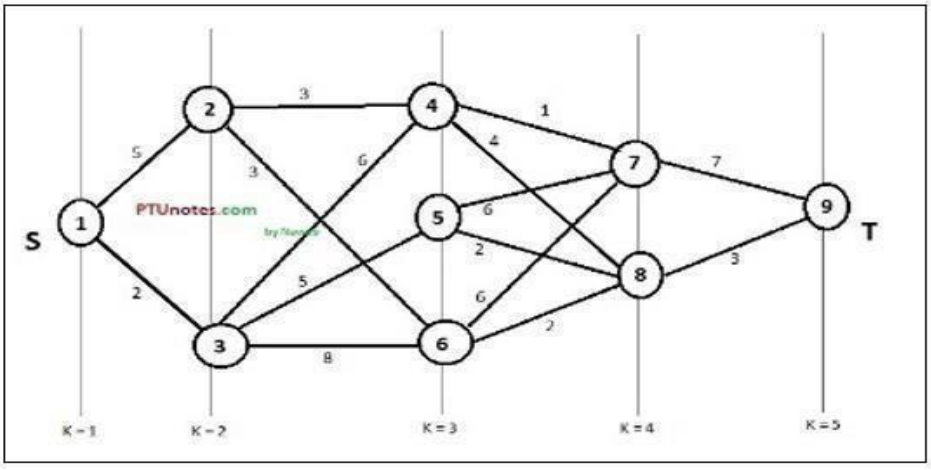
Key	A	B	C	D
Probability	0.1	0.2	0.4	0.3

10. Let us consider that the capacity of the knapsack is W = 25 and the items are as shown in the following table.

Item	A	B	C	D
Profit	24	18	18	10
Weight	24	10	10	7

Solve the problem using greedy technique.

11. Find the minimum cost of the following multi stage graph.



UNIT IV
PART A QUESTIONS

1. Define the iterative improvement technique. [Nov/Dec 2016]

An iterative improvement is a mathematical procedure that generates a sequence of improving approximate solution for a problem. Each subsequent solution involves a small, localized change in the previous feasible solution. When no such change improves the value of the objective function, the algorithm returns the last feasible solution as optimal and stops.

2. What are the disadvantages of iterative improvement?

- No small change makes an improvement, but some sequence of changes can bring the improvement
- Finding an initial feasible solution is difficult
- It is not always clear what changes should be allowed in a feasible solution

3. What is simplex method?

- It is a technique to solve linear programming problems
- The general problem of optimizing a linear function of several variables subject to a set of linear constrains is linear programming

4. Define slack variable.

Variables transforming inequality constraints into equality constrains are called slack variables

5. What is maximum flow problem?

Given a flow network G = (V, E), the maximum flow problem is to find a flow with maximum possible value.

6. Define flow network

A flow network is a directed graph G = (V, E) with the following features:

- Associated with each edge e ∈ E, there is a capacity (ce), which is a nonnegative number
- There is a single source node s ∈ V
- There is a single sink node t ∈ V

7. What is matching in bipartite graph?

A maximum matching or maximum cardinality of matching is a matching with the largest possible number of edges; it is globally optimal.

8. What do you mean by ‘perfect matching’ in bipartite graphs? [April/May 2015]

In a bipartite graph, a perfect matching is a matching in which each node has exactly one edge incident on it. A perfect matching matches all vertices of the graph.

9. What is maximum cardinality matching? [Nov/Dec 2016]

A maximum cardinality matching is the largest subset of edges in a graph such that no two edges share the same vertex. For a bipartite graph, it can be found by a sequence of augmentations of previously obtained matchings.

10. Define stable marriage problem

SMP (Stable Marriage Problem) is the problem of finding a stable matching between two sets of elements with a set of preferences for each element

11. How can you tell that a marriage is unstable?

A marriage is unstable if

a. Some given element A of the matched set prefers some given element B of the second matched Set over the element to which A is already matched
b. B also prefers A over the element to which B is already matched

12. State the extreme points

Any LP Problem with a nonempty bounded feasible region has an optimal solution; moreover an optimal solution can always be founded at an extreme point of the problem’s feasible region.

13. What is linear programming?

The general problem of optimizing a linear function of several variables subject to a set of linear constraints is linear programming

14. Define Ford – Fulkerson Method.

Ford-Fulkerson algorithm is used to solve maximum flow problem. It works as follows

- Start with the zero flow (xij = 0 for every edge)

- On each iteration, try to find a augmenting path from source to sink, it is a path along which some additional flow can be sent
- If a augmenting path is found, adjust the flow along the edges of this path by constructing residual graph and try again
  - If no flow-augmenting path is found, the current flow is maximum

15. Define Cut and its capacity.

Let any flow that goes from source s to sink t in a Graph G must cross the edge A to B at some point. It has some edge capacity from A to B. Each such division is called cut. Minimum capacity of any such division, is called as minimum cut.

PART B QUESTIONS

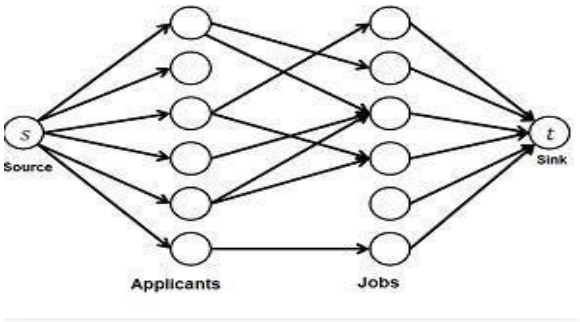
1. Explain in detail about simplex method. Every LP problem can be represented in such form

   Maximize $3x+ 5y$

   Subject to $x+ y\leq 4$

   $x+ 3y\leq 6$ $x\geq 0$, $y\geq 0$

2. Explain the maximum flow problem.
3. Explain stable marriage problem.
4. Explain maximum matching in bipartite graphs.
5. Find maximum matching in the following bipartite graph.



UNIT V PART A QUESTIONS

1. What is a Biconnected Graph? [April /May 2010]

A **biconnected** undirected graph is a connected graph that is not broken into disconnected pieces by deleting any single vertex (and its incident edges). A **biconnected** directed graph is one such that for any two vertices *v* and *w* there are two directed paths from *v* to *w* which have no vertices in common other than *v* and *w*.

2. Differentiate deterministic and nondeterministic algorithm

Algorithm with the property that the result of every operation is uniquely defined are termed deterministic algorithm.

Algorithms contains operations whose outcomes are not uniquely defined but are limited to specified sets of possibilities are termed nondeterministic algorithm.

3. Define Branch-and-Bound method. What are the searching techniques that are commonly used in Branch-and-Bound?

Branch and bound (BB) algorithm generally used for solving combinatorial optimization problems.

The algorithm explores branches of state space tree, which represent subsets of the solution set. Before exploring, the branch is checked against upper and lower bounds, and discarded if it cannot produce a better solution. The searching techniques that are commonly used in Branch-and-Bound method are:

- FIFO (Breadth First Search)
- LIFO (Depth Search)
- LC (Least Cost Search)

4. What is satisfiability problem?

The satisfiability problem is to determine whether a formula is true for some assignment of truth values to the variables.

5. Define reducibility

Let L1 and L2 be problems. Problem L1 reduces to L2 (It is written as L1 $\propto$ L2) if and only if there is

a way to solve L1 by a deterministic polynomial time algorithm using a deterministic algorithm that solves L2 in polynomial time

6. What is a decision problem?

Any problem for which the answer is either zero or one is called decision problem.

7. What is optimization problem?

Any problem that involves the identification of an optimal (either minimum or maximum) value of a given function is known as an optimization problem.

8. Define tractable and intractable problems.

Problems that can be solved in polynomial time are called tractable problems, problems that cannot be solved in polynomial time are called intractable problems.

9. Define the theory of computational complexity.

A problem's intractability remains the same for all principal models of computations and all reasonable input encoding schemes for the problem under consideration.

10. How NP-Hard problems are different form NP-Complete? [April/May 2015]

- The NP complete has a property that it can be solved in polynomial time if and only if all other NPcomplete problems can also be solved in polynomial time.
- If an NP hard problem can be solved in polynomial time then all the NP complete problems can besolved in polynomial time.
- All the NP complete problems are NP hard but there are some NP hard problems that are not known tobe NP complete.

11. Define Hamiltonian circuit problem. [April/May 2015]

Hamiltonian circuit problem is a problem of finding a Hamiltonian circuit in a graph. Hamiltonian circuit is a circuit that visits every vertex exactly once and return to the starting vertex.

12. Given an application for knapsack problem.

The knapsack problem is problem in combinatorial optimization. It derives its name from the maximum problem of choosing possible essential that can fit into one bag to be carried on a trip. A similar problem very often appears in business, combinatory, complexity theory, cryptography and applied mathematics.

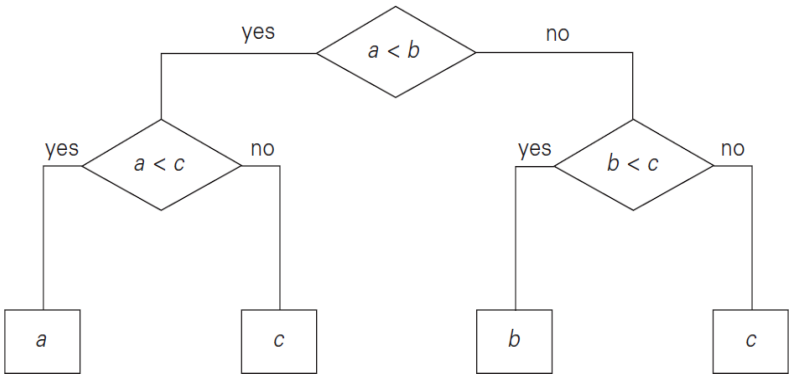
13. Give the purpose of lower bound. [May/June 2016]

- The elements are compared using operator < to make selection.
- Branch and bound is an algorithm design technique that uses lower bound comparisons.
- Main purpose is to select the best lower bound.
- Example: Assignment problem and transportation problem.

14. What is The Euclidean minimum spanning tree problem? [May/June 2016]

The Euclidean minimum spanning tree or EMST is a minimum spanning tree of a set of n points in the plane where the weight of the edge between each pair of points is the Euclidean distance between those two points. In simpler terms, an EMST connects a set of dots using lines such that the total length of all thelines is minimized and any dot can be reached from any other by following the lines.

15. Draw the decision tree for comparison of three values. [Nov/Dec 2015]



$$C_{worst}^{bs}(n) = \lfloor \log_2 n \rfloor + 1 = \lceil \log_2 (n+1) \rceil$$

16. State the reason for terminating search path at the current node in branch and bound algorithm.[Nov/Dec 2016]

- The value of the nodes bound is not better than the value of the best solution seen so far.
- The node represents no feasible solutions, because the constraints of the problem are already violated.
- The subset of feasible solutions represented by the node consists of a single point – here we compare the value of the objective function for this feasible solution with that of the best solution

seen so far and update the latter with the former, if the new solution is better.

17. What is articulation point?

A vertex v in a connected graph G is an articulation point if and only if the deletion of vertex v together with all edged incident to v disconnects the graph in to two or more nonempty components.

18. Define Heap sort.

Heap sort is a sorting algorithm. Heap sort is a 2 stage algorithm. The two stages areStage 1- Heap Construction- Construct a heap for a given array of elements
Stage 2- Maximum Deletion- Apply the root deletion operation n-1 times to the remaining heap.

19. What is backtracking?

Backtracking is one of the algorithm design technique applied to some specific types of problems as follows

- Problems which deal with searching for a set of solutions
- Problems which are used to find the set of all feasible solutions
- Problems which ask for an optimal solution satisfying some constraints

20. What is a state space tree?

A space state tree is a tree representing all the possible states (solution or non solution) of the problem from the root as an initial state to the leaf as a terminal state.

21. What is a promising node in the state-space tree?

A node in a state-space tree is said to be promising if it corresponds to a partially constructedsolution that may still lead to a complete solution.

22. What is n-queens problem?

N-Queen's Problem is classic combinatorial problem. N queens are to be placed on an n x n chessboard so that no two attack; that means no two queens are on the same row, column, or diagonal.

23. What is the subset-sum problem? Give example

Given n distinct positive numbers (usually called weights) w_i, $1 \leq i \leq n$, and m. The problem of finding all subsets of the w_i whose sums are m.
For example, let n = 4, (w1, w2, w3, w4) = (11, 13, 24, 7), and m = 31
The desired subsets are (11, 13, 7) and (24, 7)

24. What is a feasible solution and what is an optimal solution?

In optimization problems, a feasible solution is a point in the problem's search space that satisfies all the problem's constraints, while an optimal solution is a feasible solution with the best value of the objective function.

25. Compare backtracking and branch-and-bound.

Backtracking	Branch and Bound
• State-space tree is constructed using depth-first search • Finds solutions for combinatorial non-optimization problems • No bounds are associated with the nodes in the state-space tree	• State-spacetree is constructed using breadth-first search • Finds solutions for combinatorial optimization problems • Upper and Lower bounds are associated with the each and every node in the state-space tree

26. What are the requirements that are needed for performing Backtracking?

To solve any problem using backtracking, it requires that all the solutions satisfy a complex set of constraints. They are:

i. Explicit constraints - rules that restrict each xi to take on values only from a give set

ii. Implicit constraints - rules that determine which of the tuples in the solution space of I satisfy the criteriafunction

27. What are the factors that influence the efficiency of the backtracking algorithm?

The efficiency of the backtracking algorithm depends on the following four factors. They are:

i. The time needed to generate the next xk

ii. The number of xk satisfying the explicit constraints.

iii. The time for the bounding functions Bk

iv. The number of xk satisfying the Bk.

PART B QUESTIONS

1. Elaborate how backtracking techniques can be used solve the n-queens problem. Explain with an example. Nov/Dec 2019
2. Give solution to subset sum problem using Backtracking techniques. APR/MAY 2019
3. Explain the approximation algorithm for the travelling salesman problem. APR/MAY 2019
4. What is Hamiltonian problem? Explain with an example using backtracking? NOV/DEC 2017
5. Discuss the approximation algorithm for NP hard problems. NOV/DEC 2018
6. What is class NP? Discuss about any five problems for which no polynomial time algorithm has been found. APR/MAY 2018
7. Apply approximation algorithm (nearest neighbour algorithm,multifrgment-heuristic algorithm)for travelling salesperson problem. Assume that the cost function satisfies the triangle inequality. APR/MAY 2018
8. State the subset-sum problem and complete state space tree of the backtracking algorithm applied to the instance A={3,5,6,7} and d=15 of the subset-sum problem. MAY/JUNE 2016
9. Explain how the branch and bound technique is used to solve Knapsack problem (OR) Implement an algorithm for Knapsack problem using NP-Hard Approach. APR/MAY 2015
10. Apply branch and bound algorithm to solve the following travelling salesman problem. APR/MAY 2017