Sr. No.	DAA Unit 04
1.	What is backtracking? What is state space tree? Explain both terms with one example. And generate state space tree for the same example.

Ans

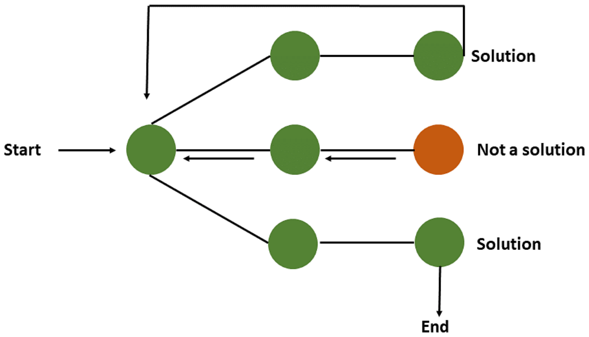
Backtracking: Backtracking is an algorithmic technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point in time.

Backtracking is a general algorithm for finding all (or some) solutions to some computational problems, that incrementally builds candidates to the solutions, and abandons each partial candidate ("backtracks") as soon as it determines that the candidate cannot possibly be completed to a valid solution.

It uses recursive calling to find the solution by building a solution step by step increasing values with time. It removes the solutions that doesn't give rise to the solution of the problem based on the constraints given to solve the problem.

State-Space Tree: A space state tree is a tree that represents all of the possible states of the problem, from the root as an initial state to the leaf as a terminal state.

Example: Crossword



2.	Give the difference between backtracking and branch & bound.

Ans

Parameter	Backtracking	Branch and bound
Approach	Backtracking is used to find all possible solutions available to a problem. When it realizes that it has made a bad choice, it undoes the last choice by backing it up. It searches the state space tree until it has found a solution for the problem.	Branch-and-Bound is used to solve optimization problems. When it realizes that it already has a better optimal solution that the pre-solution leads to, it abandons that pre-solution. It completely searches the state space tree to get optimal solution.
Traversal	Backtracking traverses the state space tree by DFS(Depth First Search) manner.	Branch-and-Bound traverse the tree in any manner, DFS or BFS.
Function	Backtracking involves feasibility function.	Branch-and-Bound involves a bounding function.
Problems	Backtracking is used for solving Decision Problem.	Branch-and-Bound is used for solving Optimization Problem.
Searching	In backtracking, the state space tree is searched until the solution is obtained.	In Branch-and-Bound as the optimum solution may be present anywhere in the state space tree, so the tree need to be searched completely.

Efficiency	Backtracking is more efficient.	Branch-and-Bound is less efficient.
Applications	Useful in solving N-Queen Problem, Sum of subset.	Useful in solving Knapsack Problem, Travelling Salesman Problem.
Solve	Backtracking can solve almost any problem. (Chess, sudoku, etc.).	Branch-and-Bound cannot solve almost any problem.

3. Explain the terms: a) Explicit constraints b) Implicit constraints

Ans Explicit constraint: The rules that restrict each element to be chosen from the given set.
Implicit constraint: It is a rule in which how each element in a tuple is related.

4. Give the recursive control abstraction of backtracking and mention its time complexity.

Ans **Recursive backtracking solution:**

```
void findSolutions(n, other params) :
    if (found a solution) :
        solutionsFound = solutionsFound + 1;
        displaySolution();
        if (solutionsFound >= solutionTarget) :
            System.exit(0);
        return

    for (val = first to last) :
        if (isValid(val, n)) :
            applyValue(val, n);
            findSolutions(n+1, other params);
            removeValue(val, n);
```

Time Complexity:

The time complexity of backtracking depends on the number of times the function calls itself. For example, if the function calls itself two times, then its time complexity is O(2 ^ N), and if it calls three times, then O(3 ^ N) and so on.
Hence the time complexity of backtracking can be defined as O(K ^ N), where ‘K’ is the number of times the function calls itself.

5. Explain 4-queen problem using backtracking.

Ans https://medium.com/@viveksonani22/4-queens-problem-using-backtracking-632b9f3d6620#:~:text=The%204%2DQueens%20Problem%5B1,column%20or%20the%20same%20diagonal.

6. What is bounding function? Explain bounding function on 4-queen problem.

7. Find the 2 possible solution vectors for 4-queen problem.

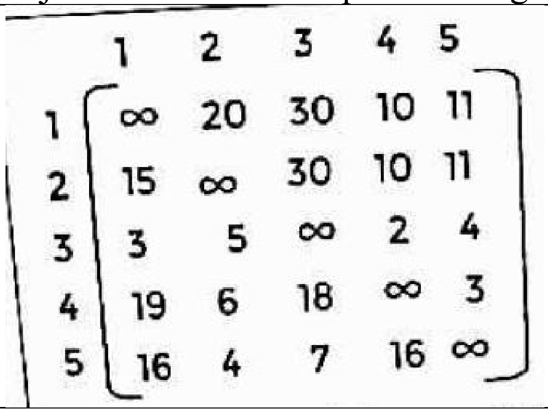
8. Explain 8-queen problem and find out its solution vector.

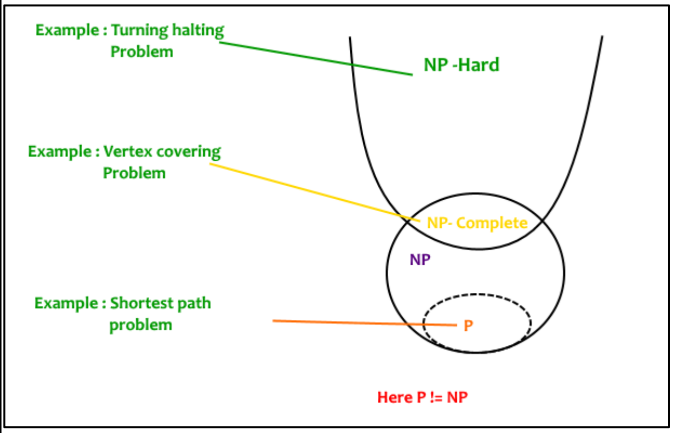
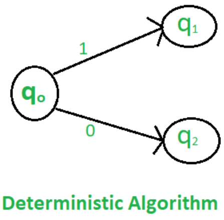
Ans https://www.geeksforgeeks.org/8-queen-problem/

9. Give the algorithm of 8-queen problem

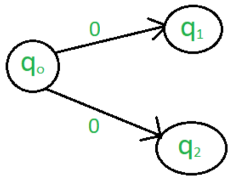
10. Give the algorithm of n-queen problem and analyze the same.

Ans	https://www.geeksforgeeks.org/n-queen-problem-backtracking-3/
11.	Explain the concept of sum of subsets problem in detail.
Ans	https://www.geeksforgeeks.org/subset-sum-backtracking-4/#:~:text=Subset%20sum%20problem%20is%20to,(no%20duplicates%20are%20presented).
12.	Consider the sum-of-subset problem, n = 4, Sum = 13, and w1 = 3, w2 = 4, w3 = 5 and w4= 6. Find a solution to the problem using backtracking. Show the state-space tree leading to the solution. Also, number the nodes in the tree in the order of recursion calls.
13.	Analyze sum of subsets algorithm on data: M = 35 and w= {5, 7, 10, 12, 15, 18, 20}
14.	Analyze sum of subsets algorithm on data: M = 35 and w= {20, 18, 15, 12, 10, 7, 5}
15.	Analyze sum of subsets algorithm on data: M=35 and w= {15, 7, 20, 5, 18, 10, 12}
16.	Solve the sum of subset problems using backtracking algorithmic strategy for the following data: n = 4, W =(w1, w2, w3, w4) = (11, 13, 24, 7) and M = 31.
17.	Solve the sum of subset problem using backtracking method: Input: set[ ] = (4, 16, 5, 23, 12), sum = 9
18.	What is bounding function?
19.	What are the bounding functions imposed on sum of subsets problem?
20.	Give the algorithm of sum of subsets problem using backtracking and analyze the same. (Give its time complexity).
Ans	Algorithm of sum of subsets problem using backtracking: if(subset is satisfying the constraint) print the subset exclude the current element and consider next element else generate the nodes of present level along breadth of tree and recur for next levels Time Complexity: Worst case time complexity: Θ(2^n)
21.	Solve the sum of subsets problem on data: M = 50 and w= {10,20,30,40}
22.	Explain the working principle of branch and bound method, how is it different from backtracking?
Ans	https://www.geeksforgeeks.org/branch-and-bound-algorithm/
23.	Explain the terms: a) State space tree b) Problem state c) Solution state d) Answer node
Ans	a) State space tree: A space state tree is a tree that represents all of the possible states of the problem, from the root as an initial state to the leaf as a terminal state. b) Problem state: Problem States change infrequently and are not dependent on workflow actions (eg. update from a client), which is why the decision was made to use the term state (current condition of an object) versus status (an outcome of an action).
24.	Explain the terms: a) Live node b) E-node c) Dead node
25.	What are the different ways of solving branch and bound problems? Explain all with suitable example. And amongst these different ways, which method de we prefer and why?
26.	Give the control abstraction of Least Cost Branch & Bound method.

27.	Give the formulas of calculating upper bound and cost of nodes in 0/1 Knapsack problem using LC B&B method.
28.	Give the algorithm of 0/1 Knapsack problem using LC B&B with its time complexity.
Ans	https://www.geeksforgeeks.org/0-1-knapsack-using-least-count-branch-and-bound/
29.	Solve the given 0/1 knapsack problem using LC B&B method. N=4, C= 15, V[ ]= {10, 10, 12, 18}, W[ ]= {2, 4, 6, 9} and update the value of upper bound as per solution.
30.	Solve the following Travelling Salesman Problem using LC B&B method. The adjacent matrix of the problem is given below:
	 $\begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} \infty & 20 & 30 & 10 & 11 \\ 15 & \infty & 30 & 10 & 11 \\ 3 & 5 & \infty & 2 & 4 \\ 19 & 6 & 18 & \infty & 3 \\ 16 & 4 & 7 & 16 & \infty \end{bmatrix} \end{array}$
	Give the complete minimum cost tour path.
Ans	https://www.geeksforgeeks.org/traveling-salesman-problem-using-branch-and-bound-2/
31.	Give the algorithm of TSP using LC B&B method with its time its complexity.
Ans	https://www.geeksforgeeks.org/traveling-salesman-problem-using-branch-and-bound-2/

Sr. No.	DAA Unit 05
1	Define the term P-Class problem.
Ans	The P in the P-class stands for Polynomial Time. It is the collection of decision problems(problems with a “yes” or “no” answer) that can be solved by a deterministic machine in polynomial time. Features: i.The solution to P problems is easy to find. ii.P is often a class of computational problems that are solvable and tractable. This class contains many natural problems like: 1.Calculating the greatest common divisor. 2.Finding a maximum matching. 3.Decision versions of linear programming.
2	Define the term NP-Class problem.
Ans	The NP in NP-class stands for Non-deterministic Polynomial Time. It is the collection of decision problems that can be solved by a non-deterministic machine in polynomial time. Features: i.The solutions of the NP class are hard to find since they are being solved by a non-deterministic machine but the solutions are easy to verify. ii.Problems of NP can be verified by a Turing machine in polynomial time. This class contains many problems that one would like to be able to solve effectively: 1.Boolean Satisfiability Problem (SAT). 2.Hamiltonian Path Problem. 3.Graph coloring.
3.	Define the terms with diagram: a) P-class problem b) NP-class problem
Ans	 Example : Turning halting Problem NP -Hard Example : Vertex covering Problem NP- Complete NP Example : Shortest path problem P Here P != NP
4.	Describe the term Deterministic Algorithms
Ans	In a deterministic algorithm, for a given particular input, the computer will always produce the same output going through the same states. 
5.	Describe the term Non-deterministic Algorithms

Ans in the case of the non-deterministic algorithm, for the same input, the compiler may produce different output in different runs. In fact, non-deterministic algorithms can't solve the problem in polynomial time and can't determine what is the next step. The non-deterministic algorithms can show different behaviors for the same input on different execution and there is a degree of randomness to it.



Non-Deterministic Algorithm

6. Give the difference between Deterministic algorithms and non-deterministic algorithms with some examples.

Ans

Deterministic Algorithm	Non-deterministic Algorithm
For a particular input, the computer will give always the same output.	For a particular input the computer will give different outputs on different execution.
Can solve the problem in polynomial time.	Can't solve the problem in polynomial time.
Can determine the next step of execution.	Cannot determine the next step of execution due to more than one path the algorithm can take.

7. Define the terms:
a) Tractable problems
b) Intractable problems.

Ans a)Tractable means that the problems can be solved in theory as well as in practice. b)The problems that can be solved in theory but not in practice are known as intractable.

8. What are the two basic points to be focused on while reducing the exponential time algorithms into polynomial time algorithms?

Ans https://www.superwits.com/library/design-analysis-of-algorithm/course-content-daa/polynomialvsexponentialrunningtime

9. What is Satisfiability problem? Why do we use it?

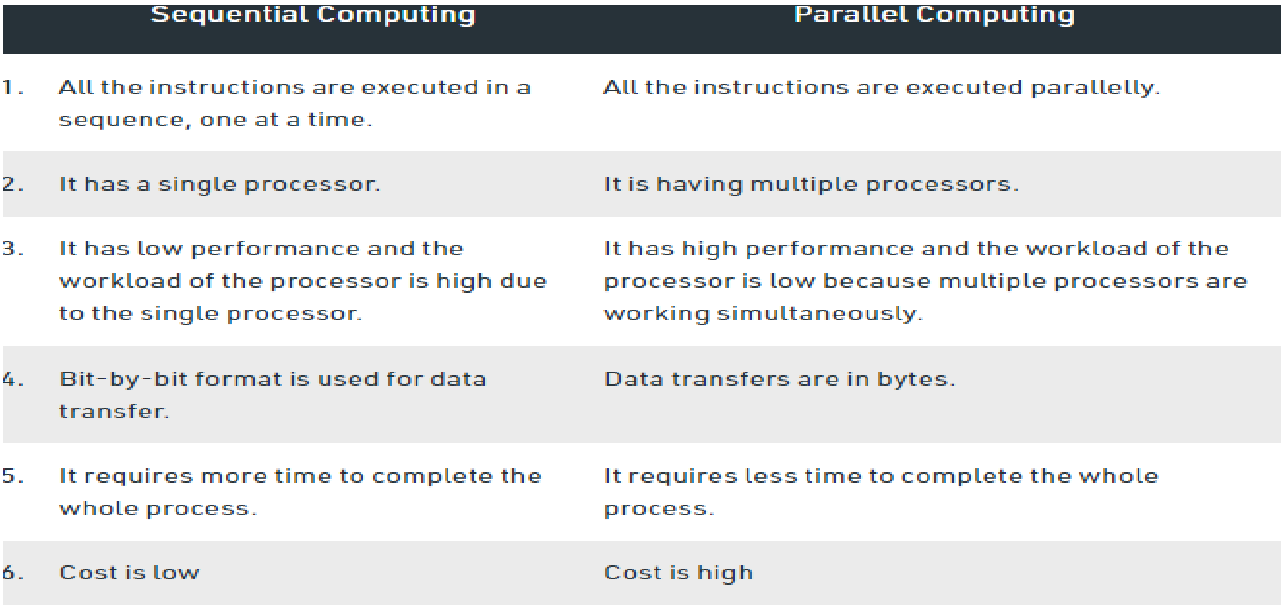
Ans Boolean Satisfiability or simply SAT is the problem of determining if a Boolean formula is satisfiable or unsatisfiable.
Satisfiable : If the Boolean variables can be assigned values such that the formula turns out to be TRUE, then we say that the formula is satisfiable.
Unsatisfiable : If it is not possible to assign such values, then we say that the formula is unsatisfiable.

10. What is reduction? How do we use it?

Ans If the solution of NPC problem does not exist then the conversion from one NPC problem to another NPC problem within the polynomial time. For this, you need the concept of reduction. If a solution of the one NPC problem exists within the polynomial time, then the rest of the problem can also give the solution in polynomial time
Example: Suppose there are two problems, A and B. You know that it is impossible to solve problem A in polynomial time. You want to prove that B cannot be solved in polynomial time. So you can convert the problem A into problem B in polynomial time.
{Refer Class Notes}

11.	Describe NP-Hard problem in detail.
Ans	A Problem X is NP-Hard if there is an NP-Complete problem Y, such that Y is reducible to X in polynomial time. NP-Hard problems are as hard as NP-Complete problems. NP-Hard Problem need not be in NP class. To solve this problem, it do not have to be in NP . Do not have to be a Decision problem Example: Halting problem, Vertex cover problem, etc
12.	Describe NP-Complete problem in detail.
Ans	A problem X is NP-Complete if there is an NP problem Y, such that Y is reducible to X in polynomial time. NP-Complete problems are as hard as NP problems. A problem is NP-Complete if it is a part of both NP and NP-Hard Problem. A non-deterministic Turing machine can solve NP-Complete problem in polynomial time. NP-Complete problems can be solved by a non-deterministic Algorithm/Turing Machine in polynomial time. To solve this problem, it must be both NP and NP-hard problems. It is exclusively a Decision problem. Example: Determine whether a graph has a Hamiltonian cycle, Determine whether a Boolean formula is satisfiable or not, Circuit-satisfiability problem, etc.
13.	Describe the terms NP-Hard and NP-Complete with its diagram.
Ans	Q.3
14.	Show that Clique Decision Problem (CDP) is NP-Hard Problem.
Ans	https://www.geeksforgeeks.org/proof-that-clique-decision-problem-is-np-complete/#:~:text=The%20Boolean%20Satisfiability%20Problem%20(S,C%20to%20be%20NP%2DHard.
15.	What is a clique?
Ans	In an undirected graph, a clique is a complete sub-graph of the given graph. Complete sub-graph means, all the vertices of this sub-graph is connected to all other vertices of this sub-graph.
16.	What is a vertex cover problem?
Ans	The vertex Cover of a graph is defined as a subset of its vertices, such for every edge in the graph, from vertex u to v, at least one of them must be a part of the vertex cover set.
17.	Show that Vertex Cover Problem is NP-Complete Problem.
Ans	https://www.geeksforgeeks.org/proof-that-vertex-cover-is-np-complete/
18.	Show that 3-Sat Problem is NP-Complete Problem.
Ans	https://www.geeksforgeeks.org/proof-that-sat-is-np-complete/
19.	Which problems are termed as NP-Complete problems?
Ans	NP-complete problem, any of a class of computational problems for which no efficient solution algorithm has been found. Many significant computer-science problems belong to this class. e.g., the traveling salesman problem, satisfiability problems, and graph-covering problems.
20.	Draw and explain the basic diagram of complexity classes.
Ans	https://www.geeksforgeeks.org/types-of-complexity-classes-p-np-conp-np-hard-and-np-complete/ Q.1,2,3,11,12,13

Sr. No.	DAA Unit 06
1	Define the term Parallelism.
Ans	The state of being parallel. Data Parallelism means concurrent execution of the same task on each multiple computing core. It is the use of multiple processing elements simultaneously for solving any problem. Problems are broken down into instructions and are solved concurrently as each resource that has been applied to work is working at the same time.
2	Write the principles of Parallel Algorithm Design.
Ans	
3.	What are the different approaches of Parallel algorithm?
Ans	1.Bit-level parallelism – It is the form of parallel computing which is based on the increasing processor's size. It reduces the number of instructions that the system must execute in order to perform a task on large-sized data. Example: Consider a scenario where an 8-bit processor must compute the sum of two 16-bit integers. It must first sum up the 8 lower-order bits, then add the 8 higher-order bits, thus requiring two instructions to perform the operation. A 16-bit processor can perform the operation with just one instruction. 2.Instruction-level parallelism – A processor can only address less than one instruction for each clock cycle phase. These instructions can be re-ordered and grouped which are later on executed concurrently without affecting the result of the program. This is called instruction-level parallelism. 3.Task Parallelism – Task parallelism employs the decomposition of a task into subtasks and then allocating each of the subtasks for execution. The processors perform the execution of sub-tasks concurrently. 4. Data-level parallelism (DLP) – Instructions from a single stream operate concurrently on several data – Limited by non-regular data manipulation patterns and by memory bandwidth
4.	Discuss about Sequential Computing.
Ans	Sequential Computing is the type of computing where one instruction is given at a particular time and the next instruction has to wait for the first instruction to execute. It is also known as a traditional computing method because all the instructions are executed in a sequence. It is having a single processor with low performance and high work-load of the processor. The main disadvantage of using this computing is that it takes more time as a single instruction is getting executed at a given point of time. To remove this disadvantage of sequential computing, parallel computing was introduced.
5.	Describe Parallel Computing.
Ans	Parallel Computing is a type of computing where many calculations or the execution of the processes are carried out parallelly or simultaneously.

<table>
<tr><td></td><td>The type of computing where processes can execute simultaneously at a time is known as parallel computing. It saves time as the processes are executed simultaneously.
It solves larger problems. There are multiple processors with high performance and low work-load per processor.</td></tr>
<tr><td>6.</td><td>Differentiate between Sequential Processing and Parallel Processing.</td></tr>
<tr><td>Ans</td><td>
<table>
<tr><th></th><th>Sequential Computing</th><th>Parallel Computing</th></tr>
<tr><td>1.</td><td>All the instructions are executed in a sequence, one at a time.</td><td>All the instructions are executed parallelly.</td></tr>
<tr><td>2.</td><td>It has a single processor.</td><td>It is having multiple processors.</td></tr>
<tr><td>3.</td><td>It has low performance and the workload of the processor is high due to the single processor.</td><td>It has high performance and the workload of the processor is low because multiple processors are working simultaneously.</td></tr>
<tr><td>4.</td><td>Bit-by-bit format is used for data transfer.</td><td>Data transfers are in bytes.</td></tr>
<tr><td>5.</td><td>It requires more time to complete the whole process.</td><td>It requires less time to complete the whole process.</td></tr>
<tr><td>6.</td><td>Cost is low</td><td>Cost is high</td></tr>
</table></td></tr>
<tr><td>7.</td><td>Explain RAM and PRAM models in detail with diagrams.</td></tr>
<tr><td>Ans</td><td>Q.8,9</td></tr>
<tr><td>8.</td><td>Explain the RAM model with diagram.</td></tr>
<tr><td>Ans</td><td>https://www.geeksforgeeks.org/what-is-random-access-machine/#:~:text=The%20Random%20Access%20Machine%20(RAM,memory%20and%20cannot%20be%20modified.</td></tr>
<tr><td>9.</td><td>Discuss PRAM model with its diagram.</td></tr>
<tr><td>Ans</td><td>https://www.geeksforgeeks.org/pram-or-parallel-random-access-machines/</td></tr>
<tr><td>10.</td><td>Write the constraints been enforced on PRAM model in detail.</td></tr>
<tr><td>Ans</td><td></td></tr>
<tr><td>11.</td><td>Describe the terms:
a) EREW c) CREW
b) ERCW d) CRCW</td></tr>
<tr><td>Ans</td><td>a) EREW: also called Exclusive Read Exclusive Write is a constraint that doesn’t allow two processors to read or write from the same memory location at the same instance.
b) CREW: also called Concurrent Read Exclusive Write is a constraint that allows all the processors to read from the same memory location but are not allowed to write into the same memory location at the same time.
c) ERCW: also called Exclusive Read Concurrent Write is a constraint that allows all the processors to write to the same memory location but are now allowed to read the same memory location at the same time.
d) CRCW: also called Concurrent Read Concurrent Write is a constraint that allows all the processors to read from and write to the same memory location parallelly.</td></tr>
<tr><td>12.</td><td>How to analyze parallel algorithms?</td></tr>
<tr><td>Ans</td><td>https://www.tutorialspoint.com/parallel_algorithm/parallel_algorithm_analysis.htm</td></tr>
</table>

13.	What is the basic concept behind Amdahl's Law?
Ans	ppt
14.	Define the term Speedup.
Ans	
15.	Define the term Speedup under Amdahl's Law.
Ans	The speedup of a parallel algorithm over a corresponding sequential algorithm is the ratio of the compute time for the sequential algorithm to the time for the parallel algorithm. If the speedup factor is n, then we say we have n-fold speedup.
16.	Explain Amdahl's Law in detail.
Ans	ppt
17.	State the factors on which Amdahl's Law is dependent.
Ans	
18.	What does Amdahl's law say?
Ans	
19.	Consider n=10, P=70% and S=30%. Calculate Amdahl's Speedup.
Ans	
20.	Consider n=10, P=90% and S=10%. Calculate Amdahl's Speedup.
Ans	
21.	Describe Optimal Parallel Algorithms.
Ans	
22.	Write a short note on Optimal Parallel Algorithms
Ans	
23.	State the 8 points to design an embedded algorithm.
Ans	
24.	How to design an embedded algorithm? Give examples of embedded systems.
Ans	