UNIT- ELEVEN

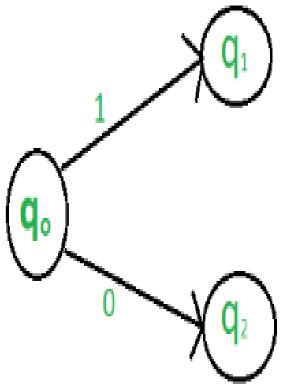
ALGORITHMS

Topics to be Covered

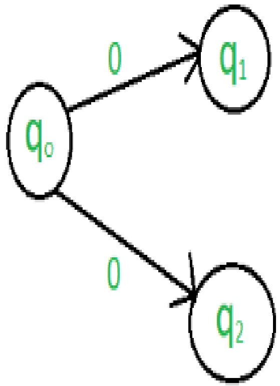
- Deterministic and non deterministic Algorithms
- Divide and conquer Algorithm
- Series and parallel Algorithm
- Heuristic and Approximate Algorithms

Deterministic and non-deterministic Algorithm

- In a **deterministic algorithm**, for a given particular input, the computer will always produce the same output going through the same states.
- 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.



Deterministic Algorithm



Non-Deterministic Algorithm

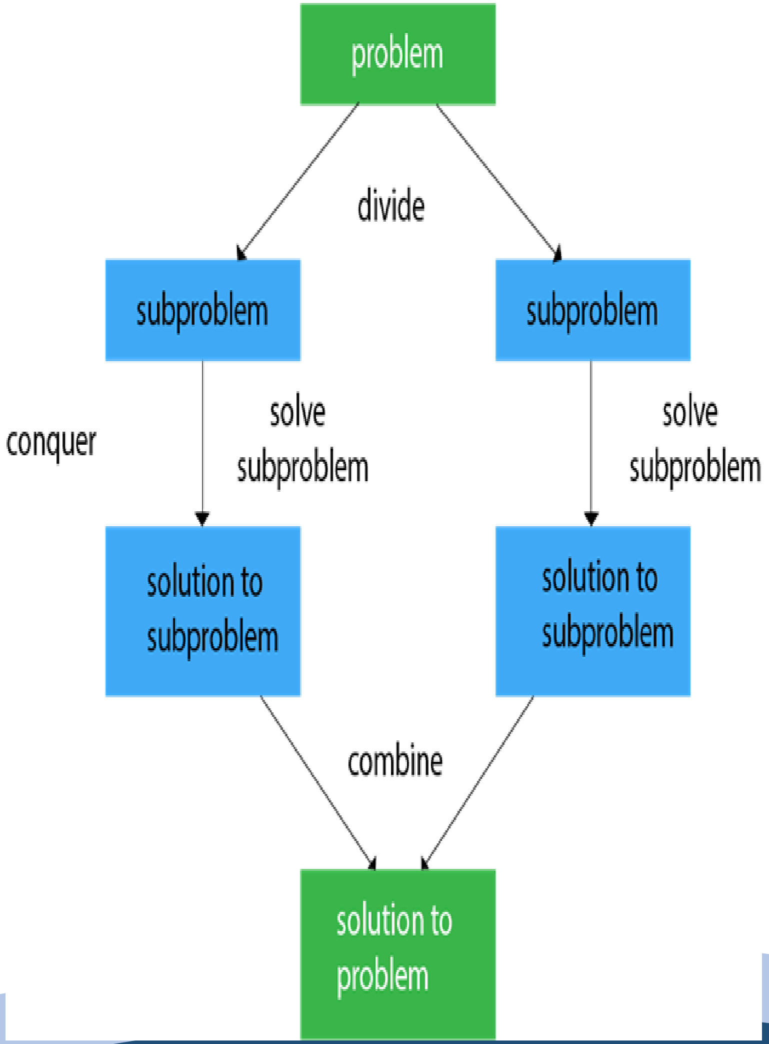
Deterministic Algorithm	Non-deterministic Algorithm
For a particular input, the computer will give always the same output.	For a particular input the computer will give different outputs on different execution.
Can solve the problem in polynomial time.	Can't solve the problem in polynomial time.
Can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

Divide and Conquer Algorithm

- **Divide and Conquer** is an algorithmic pattern. In algorithmic methods, the design is to take a dispute on a huge input, break the input into minor pieces, decide the problem on each of the small pieces, and then merge the piecewise solutions into a global solution. This mechanism of solving the problem is called the Divide & Conquer Strategy.

Divide and Conquer algorithm consists of a dispute using the following three steps.

- **Divide** the original problem into a set of subproblems.
- **Conquer:** Solve every subproblem individually, recursively.
- **Combine:** Put together the solutions of the subproblems to get the solution to the whole problem.



Divide and Conquer Algorithm

- The specific computer algorithms that are based on the Divide & Conquer approach:

1. Maximum and Minimum Problem
2. Binary Search
3. Sorting (merge sort, quick sort)
4. Tower of Hanoi.

Fundamental of Divide & Conquer Strategy:

- Relational Formula
- Stopping Condition

❑ **Relational Formula:** It is the formula that we generate from the given technique. After generation of Formula we apply D&C Strategy, i.e. we break the problem recursively & solve the broken sub-problems.

❑ **Stopping Condition:** When we break the problem using Divide & Conquer Strategy, then we need to know that for how much time, we need to apply divide & Conquer. So the condition where the need to stop our recursion steps of D&C is called as Stopping Condition.

Applications of Divide and Conquer Approach

- **Binary Search:** The binary search algorithm is a searching algorithm, which is also called a half-interval search or logarithmic search. It works by comparing the target value with the middle element existing in a sorted array.
- **Quicksort:** It is the most efficient sorting algorithm, which is also known as partition-exchange sort. It starts by selecting a pivot value from an array followed by dividing the rest of the array elements into two sub-arrays. The partition is made by comparing each of the elements with the pivot value. It compares whether the element holds a greater value or lesser value than the pivot and then sort the arrays recursively.
- **Merge Sort:** It is a sorting algorithm that sorts an array by making comparisons. It starts by dividing an array into sub-array and then recursively sorts each of them. After the sorting is done, it merges them back.
- **Closest Pair of Points:** It is a problem of computational geometry. This algorithm emphasizes finding out the closest pair of points in a metric space, given n points, such that the distance between the pair of points should be minimal.

Advantages of Divide and Conquer

- This algorithm is much faster than other algorithms.
- It efficiently uses cache memory without occupying much space
- Since these algorithms inhibit parallelism, it does not involve any modification and is handled by systems incorporating parallel processing.

Disadvantages of Divide and Conquer

- Since most of its algorithms are designed by incorporating recursion, so it necessitates high memory management.
- An explicit stack may overuse the space.
- It may even crash the system if the recursion is performed rigorously greater than the stack present in the CPU.

Series and Parallel Algorithm

- **Sequential Algorithm** − An algorithm in which some consecutive steps of instructions are executed in a chronological order to solve a problem.
- **Parallel Algorithm** − The problem is divided into sub-problems and are executed in parallel to get individual outputs. Later on, these individual outputs are combined together to get the final desired output.

Heuristic and Approximate Algorithm

- **A heuristic algorithm** is one that is designed to solve a problem in a faster and more efficient fashion than traditional methods by sacrificing optimality, accuracy, precision, or completeness for speed.
- Heuristic algorithms often times used to solve NP-complete problems, a class of decision problems.
- In these problems, there is no known efficient way to find a solution quickly and accurately although solutions can be verified when given.
- Heuristics can produce a solution individually or be used to provide a good baseline and are supplemented with optimization algorithms.
- Heuristic algorithms are most often employed when approximate solutions are sufficient and exact solutions are necessarily computationally expensive.

Approximation Algorithm

- **An approximation algorithm** is a way of dealing with NP-completeness for an optimization problem.
- This technique does not guarantee the best solution. The goal of the approximation algorithm is to come as close as possible to the optimal solution in polynomial time. Such algorithms are called approximation algorithms.

Features of Approximation Algorithm :

- An approximation algorithm guarantees to run in polynomial time though it does not guarantee the most effective solution.
- An approximation algorithm guarantees to seek out high accuracy and top quality solution(say within 1% of optimum)
- Approximation algorithms are used to get an answer near the (optimal) solution of an optimization problem in polynomial time

Any Query?