MODULE 5
ALGORITHMIC COMPLEXITY

Complexity of an algorithm are of two types.

1.Structural complexity

2.Computational complexity

Computational complexity is based on two measure –time and space.
ie Computational complexity is the amount of time ,space and other resources needed to recognize a language using a universal computing device like Turing Machine

Tractable and intractable problems

Tractable problems are problems that can be solved effectively ie we are able to construct algorithms to solve the problem effectively.

A problem is solvable if there exists some algorithm to solve this problem.But in practice algorithm may require a lot of space and time.When the time and space required for implementing the steps of a particular algorithm are reasonable, we can say that the problem is tractable.

Eg:Sorting algorithms,Linear search etc

Certain decidable problems required exponential time to solve it.Such hard and time –consuming problems are called intractable problems.

Eg: Integer bin packing problem.

Complexity Classes:

<u>1.P Class</u>

P-problems are solved using algorithm whose execution time is either given by a polynomial on the size of the input or can bounded by such a polynomial.It produces correct anser for any string of length 'n' in atmost n^k steps where k is a constant independent of 'n'.

P is the set of all decision problems solvable by deterministic algorithms in polynomial time.

Eg: Mergesort, quicksort etc

Eg:Equivalence of DFA's –Let M1 and M2 are two DFA's then the problem is L(M1) =L(M2)

Another eg is :Given a set of 'n' numbers x1,x2…xn and a key number 'x'.Determine the occurance of 'x' in the given list.ie Linear Search and the time complexity is O(n).

<u>2.NP Class or NP Problems</u>

NP is the set of all decision problems solvable by non-deterministic algorithms in polynomial time.

Examples are

1.Equivalence of NFA's and Regular expressions

We can solve this problem by transforming the two given NFA to deterministic ones and then check the resultant DFA for equivalence.The problem is that the transformation from Regular expression or NFA to DFA may increase exponentially the number of states of the automaton.Hence time complexity will be exponential.

2.Integer-Bin Packing

Given a set of 'n' positive integers.Our task is to arrange these numbers into 2 bins so that the sum of integers in one bin is equal to the sum of integers in other pile.
For eg:Given the integers (19,23,32,42,50,62,77,88,89,105,114,123,176) These numbers sum to 1000.Can they be divided into 2 bins A and B such that the sum of integers in each bin is 500.There is a non-deterministic algorithm,for each number,put it in the correct bin.This requires linear time. There is a deterministic solution to this problem.ie try to make 2^{13} arrangements.But it takes $O(2^n)$ time.

3.Boolean Satisfiability problem

A Boolean variable is the one that can takes exactly two values(true or false).Boolean operators are then used to combine Boolean variables to form Boolean expressions.Consider

Boolean expression in Conjuctive Normal Form(CNF).We can create expressions from variables x_1, x_2 …..x_n starting with e=t1 ^t2^…tk

The terms t1,t2…tk are created by OR ing together variables and their negations.

ie $t_i = S_1 V S_2 V S_p$ where $S_1, S_2, \ldots S_p$ are some variables or negations.

Satisfiability problem states that given an expression 'e' in CNF, is there an assignment of variables to the variables x_1, x_2 …..x_n that will make the value of 'e' true?

For x_1, x_2 …..x_n deterministic algorithm will take all the possible values and evaluate the expression.Then the time complexity is $O(2^n)$.But the non-deterministic approach is to guess the value of each x_i and evaluate 'e'.Then it is O(n) algorithm.

<u>3.NP-Hard Class</u>

An algorithm for solving the problem can be translated into one for solving any other NP-Problem.

<u>4.NP-Complete Class</u>

- **A problem which is both NP and NP-Hard**
- **Hardest of all NP-Problems**
- **Eg: problems are 3CNF,Formula satisfiability(SAT) problem**

COOK'S THEORM:

In computational complexity theory,the cook's –levin theorm also known as Cook's theorm states that Boolean satisfiability problem is NP-Complete. That is any problem in NP can be reduced in polynomial time by a deterministic TM to a problem of determining whether a Boolean formula is satisfiable.