18-5-2023
Thursday

PAGE NO.:
DATE / /

(Resarch Topic)

- P, NP, NP-Hard and N-P complete.

	NP-hard ~~stat~~ satisfiability / CNF
Polynomial Algorithm	Exponetial Algorithm.
1) Linear Search	0/1 Knap Sack problem
2) Binary Search	TSP
3) Huffman coding	Sum of subsets
4) Minimum cost spanning Tree	Graph coloring
5) Optimal merge pattern	Hamiltonian
6) Merge Sort	~~θ(n^n), θ~~
7) $o(n), o(n\log n), o(\log n), o(n^2)$	$o(n^n), o(2^n)$

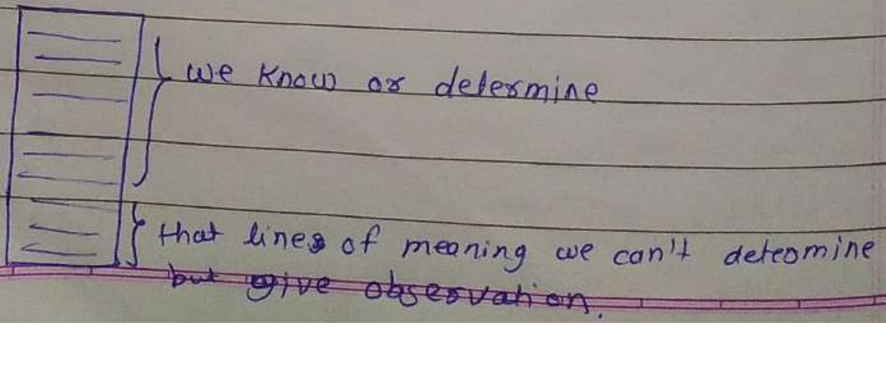
Exponential Algorithm take more time compare to polynomial Algorithm.

Objective:-

Convert exponetial time taking algorithm to polynomial time taking algorithm.

- Similarity
- Write non-deterministic Algorithm.

When we not determine the meaning of algorithm the it is ND Algo.


we know or determine
that lines of meaning we can't determine but give observation.

Example :- of non

```
Algorithm NSearch (A, n, key)
{
    j = choice ();
    if (key == A[j])
    {
        write(1);
        success ();
    }
    else
    {
        write(0);
        failure();
    }
}
```

choice () :-

It arbitarly choose index of the key element from a given array but how it choose select the index of key element that is not known to us.

failure () :-

It signals unsuccessful completion but ~~body~~ structure of the failure function that is not known to us.

Success () :-

It signals successful completion but structure of the success () function that is not known to us

Writing an algorithm meaning of few statments is known to us but rest of few statments meaning not know to us.

- Determinsistic Algorithm.

If you have the algorithm and meaning of the all statements of algorithm is know to you or derived to you then it is deterministic algorithm.

- ~~Simal~~ Similarity:-
- CNF = propositional calculas problem.

CNF = satisfiability.

CNF = $(x_1, \sqrt{\overline{x_2}} \cdot \vee (\overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee \overline{x_3})$

Possible solution

x_1	x_2	x_3
0	0	0
1	0	0
0	1	0
0	0	1
1	0	1
0	1	1
	⋮	

$2^3 \simeq 2^n$

base ~~time~~ Problem Algorithm ~~is~~ time taken is same of an expontial algorithm. So Justify that expantial algorithm is similar to base problem.

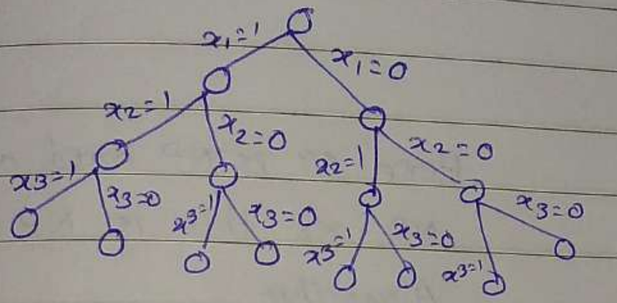
0/1 Knapsack problem

$p = \{10, 5, 8\}$

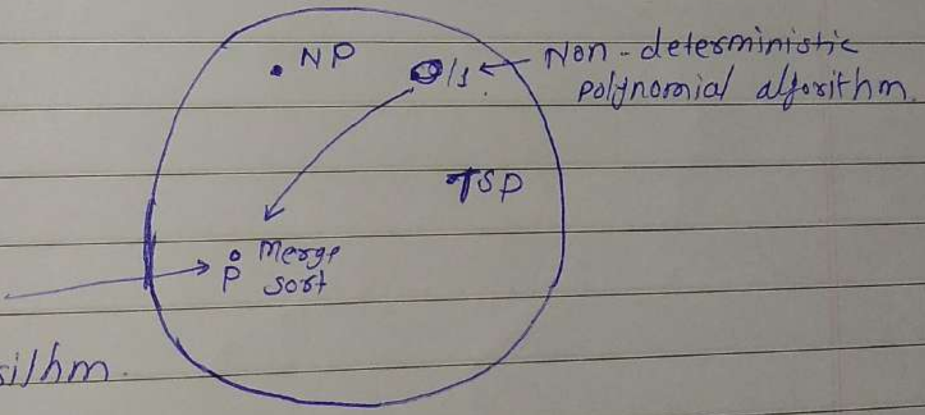
$w = \{2, 5, 3\}$.

0/1 0/1 0/1.

Backtracking - state space tree.



Base problem ~~time~~ is ~~take~~ ~~time~~ similar to Exponetial algorithm because of state space tree.



Determinstic Polynomial algorithm.

Today which is non determinstic algorithm, tommorow or in future it may becomes deterministic Algorithm.

When non deterministic algorithm converted into deterministic algorith then size of the deterministic algorithm goen on increases.

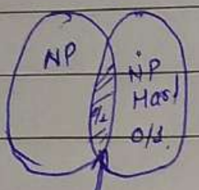
NP-Hard   NP-Hard

$\text{Sat} \propto \text{0/1 knap.}$ — NP — NP complete

↑

Reduction.

NPcomplete = NP Hard + NP +similarity.



Here 0/1 is NP hard and also NP so this is NP complete Algorithm.