Page No.
Date

MOD-5

5.1 NP and NP complete
5.2 NP reducibility

1st objective → find relation betn exponential time algo.

Non-deterministic algo: we dont know how this is working.
- this helps preserving our algorithm / research. of converting exponential time algo to polynomial.

- basically we can just depict our idea in non-deterministic algorithms.

eg: nondeterministic Search (A, n, key)
{
    j = choice (); // non deterministic ——— 1 unit time
    if (k = A[j])
    { write (j);
      Success (); // non deterministic ——— 1 time
    }
    write (0);
    failure (); // non deterministic ——— 1 time.
}

O(1)

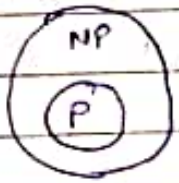
suppose I have an array

1	2	3	4	5
4	5	8	6	7

and we search for 8,
choice () will give me index directly in constant time.

* P = set of algo that take polynomial time deterministic.
* NP = non deterministic polynomial time taking algo.

* P is a subset of NP



* If we are unable to solve in poly. time, we then just need to show that they are related and can be solved in poly time., if one is solved other can be solved.

for reaching them, we need some problem as base for problem.

⇒ base problem → satisfiability problem.

2^n CNF – satisfiability {using boolean var x_1, x_2, x_3}

$$CNF = \underbrace{(x_1 \vee \overline{x_2} \vee x_3)}_{c_1} \wedge \underbrace{(\overline{x_1} \vee x_2 \vee \overline{x_3})}_{c_2}$$

$x_i = \{x_1, x_2, x_3\}$, ∴ for what value of x_i, the CNF formula is true

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

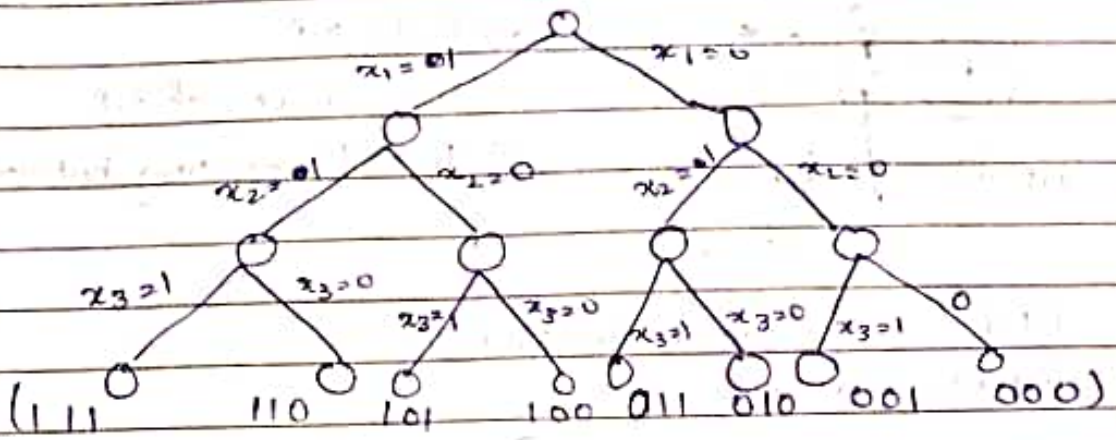
$= 2^3 = 2^n$ problems.

the values to be substituted in CNF formula.

$= 2^n$ time

satisfiability ⇒ NP hard.

State Space :



∴ 0/1 knapsack. $p = \{10, 8, 12\}$ $n=3$

$w = \{5, 4, 3\}$ $m=8$

$x_i = \{\ ,\ ,\ \}$

now, the above tree is also same for 0/1 knapsack point

* Reduction,

→ converting satisfiability into 0/1 knapsack with polynomial time.

if satisfying is reducing in 0/1 knapsack ~~NP hard te. satisfiability~~ the it is also NP hard.

* for satisfiability ~~toes~~ we have a NP algorithm, then this satisfiability is called NP complete.

NP hard + NP algorithm = NP complete

* If any algo is reduced by NP hard, then the algo is also NP hard

and if we have NP algo for the algo(s), then it is NP complete.

Calculating time Complexity of algo.

① loc

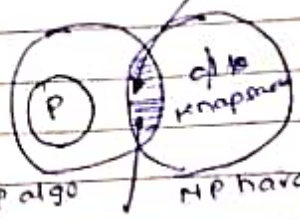
2 7 6
9 5 1
4 3 8

Satisfiability = NP hard.

Page No.
Date

③ neste

A = and.



→ P ⊆ NP

P is subset of NP

→ P = NP, existing, but you dont kno

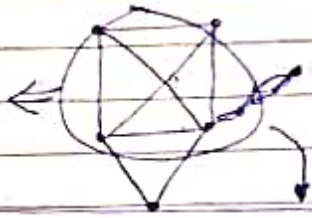
⑥ If

CDP → clique decision.

complete graph.

$|V| = n$

$|E| = \frac{n(n-1)}{2}$



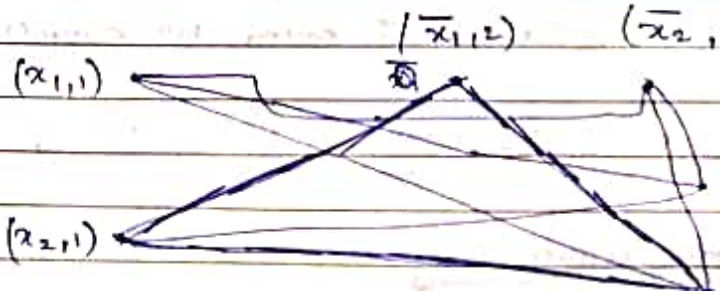
Sub graph of a graph & sub-graph is complete graph = clique.

satisfiability reduces to CDP

x_1, x_2, x_3 — 0 OR 1 = 1 — 1 OR 0 = 1 — 0 OR 1 = 1 = 1+1+1 = 3

$f = \underbrace{(x_1 \vee x_2)}_{c_1} \wedge \underbrace{(\bar{x}_1 \vee \bar{x}_2)}_{c_2} \wedge \underbrace{(\bar{x}_1 \vee x_3)}_{c_3}$

$f = \wedge C_i \quad i = 1 \text{ to } k$, $k = 3$. ⇒ k size clique



$x_1 = 0$
$x_2 = 1$
$x_3 = 1$

clique vertex