Lecture -07 14.09.24

The graph coloring problem:

— originated from map coloring problem

— coloring regions of a map so that regions can be visualized; so no two neighbors will have the same color.

— difficult when # of colors < # of regions

Map coloring → ~~edge~~ face(?) coloring

→ Dual graph (or Vertex coloring.

Graph coloring: coloring vertices

constraint: If $(u,v) \in E$, then u and v must be colored in diff. colors.

K- coloring prob:

coloring the vertices with at most K colors. So that coloring constraint is satisfied.

Decision Version: Given a graph G and an integer K. Does G have a K-coloring?

K=1: Only possible for null graph.

যে graph এর edge set empty.

K=2: Bipartite

There will be no odd cycles.

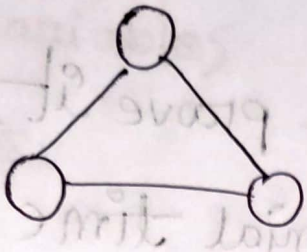
checking: DFS/BFS

কোথাও conflict হইলেই gone.

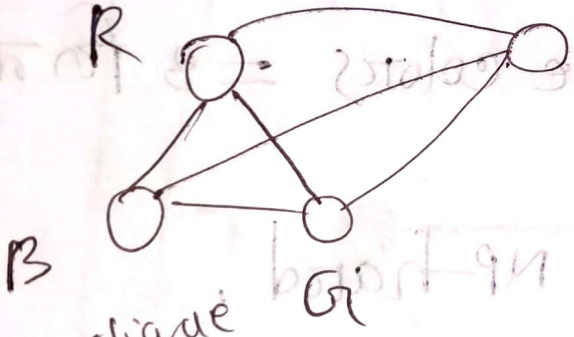
$\in$ class P.

$K = 3$:

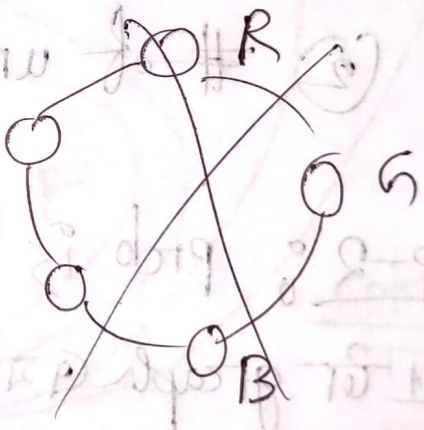
Triangle असेल तर, then 3 ला color लागतोच.

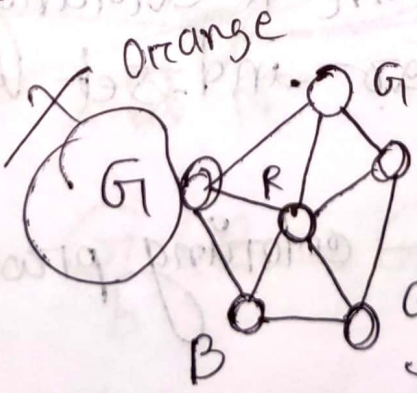


// Max clique चा size जर K असेल, तर कमीत color at least लागतोच. → but opposite not true.



clique चा size 2 असेल.





→ wheel graph

→ 4 size चा कोणता clique नाही, still 4 ला color लागतोच.

Theorem: 3-coloring problem is NP-complete.

step: prob is in NP — prove it.

checking in polynomial time.

(1) If $(u,v) \in E$, then whether u and v have diff. colors

(2) # of unique colors $= 3$ कि ना।

step-2: prob is in NP-hard.

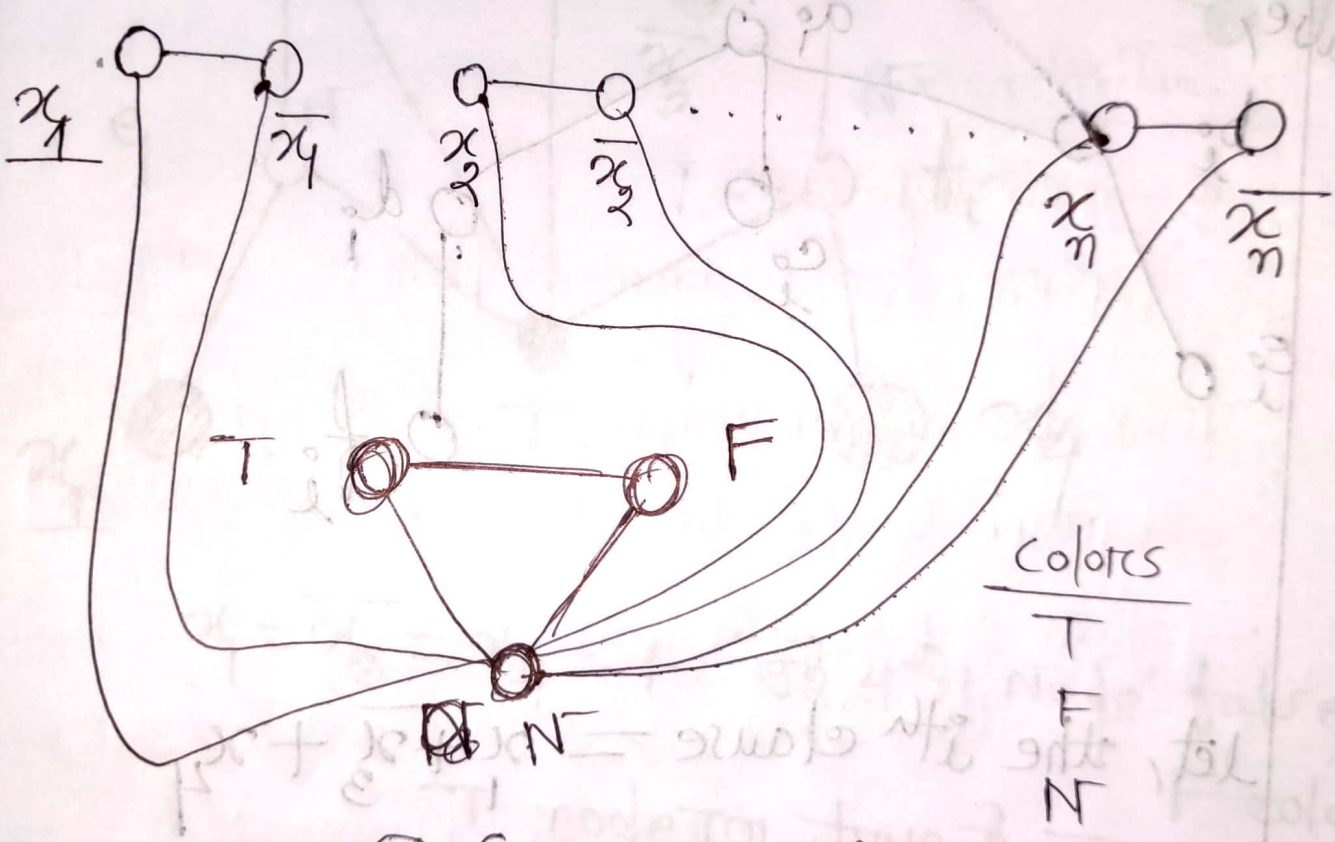
// 1 दि graph G यदि K-coloring भाए, then K size G० ind. set भाकाव।

3-SAT $\leq_P$ 3-coloring problem

$Y = (\quad)(\quad)(\quad)$

let, n variables

why edge? coloring এমন same color না পায়।



G_b (Base Graph)

This base graph is 3-colorable.

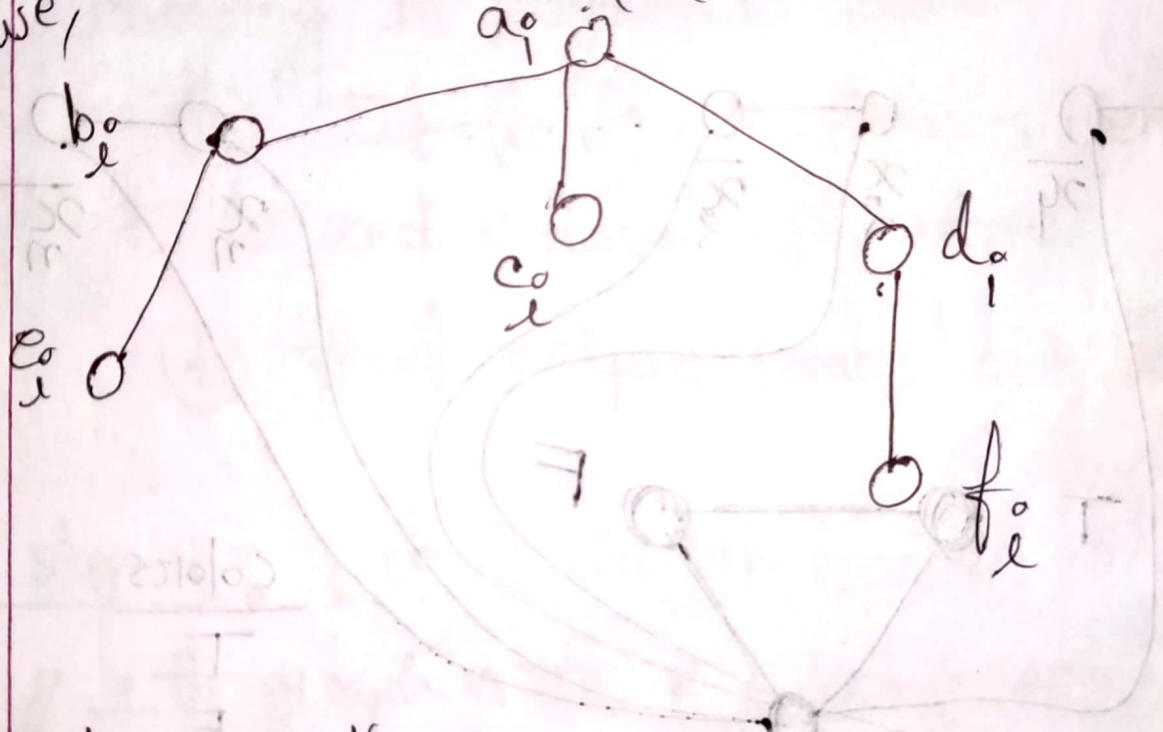
$(x_i, \bar{x}_i)$ এই edge এমন $x_i = T$ হলে $\bar{x}_i = F$

$x_i = F$ " $\bar{x}_i = T$

Base graph টা 3-colorable
Y satisfiable তার প্রমাণ কী ?

Now, প্রতিটা clause এর জন্য 1 টা subgraph তৈরী করবো।

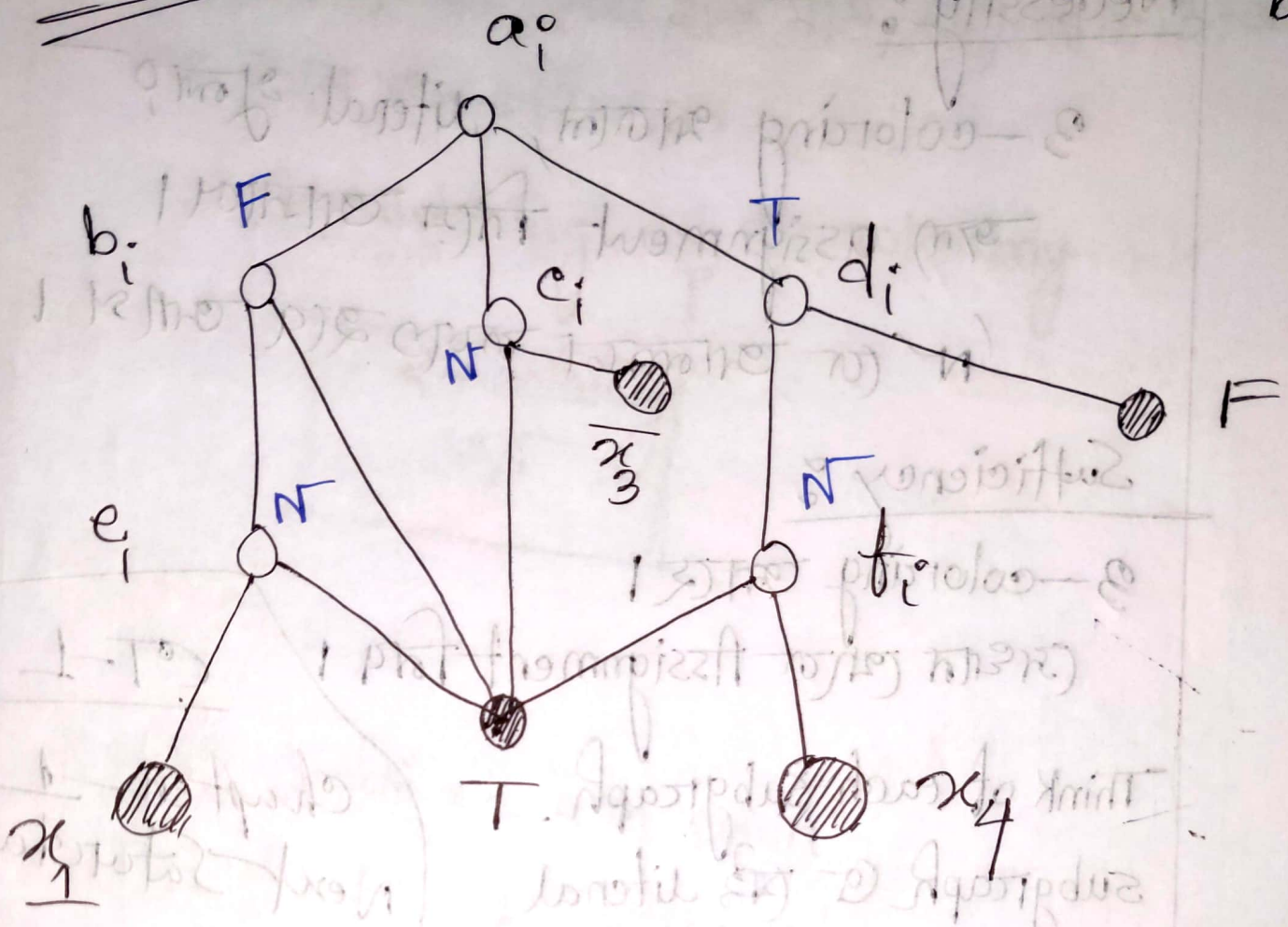
for the ith clause,



let, the ith clause $= x_1 + \bar{x}_3 + x_4$

G_1

= Nodes from G_{1b}



$x_1 = \overline{x_3} = x_4 = F$ এই 4টা Node false, F color,

T node তে true ই, T বাকিটা propagate করতে হবে।

→ clause এর প্রতিটা যদি false হয়, then 3-coloring possible না।

at least 1টাও যদি True হয়, ∈ literals, then 3-coloring possible.

Y is satisfiable iff G has a coloring.

Necessity :

3-coloring থাকলে, literal গুলো
কোন assignment নিয়ে আসবে।
'N' কে আলাদা করতে হবে আগে।

Sufficiency :

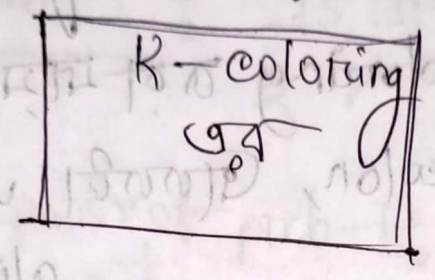
3-coloring আছে।
সেখান থেকে Assignment নিব।

Think of each subgraph.
subgraph এ যেই literal
গুলো আছে, কোন গুলো
possibly True → এগুলো listing
করে করে process
করলেই বের হবে কোন গুলা
must T করতেই হবে।

CT-1
Chapter 1
Next Saturday.

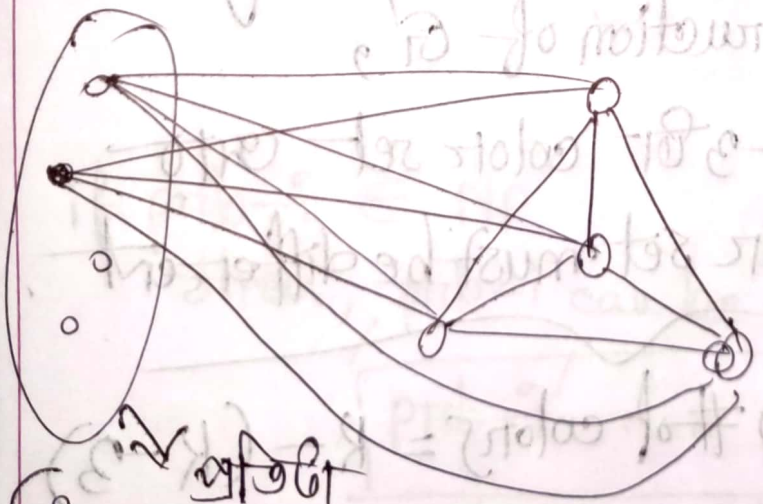
# of colors যদি fixed হয়ে যায়, তাহলেও 3-coloring এর চেয়ে বড় কঠিন না।

$3\text{-coloring} \leq_P K\text{-coloring}$



$K = x$ ∴ extra = 4টা color

2 4টা vertex এর clique নিও।



G → প্রতিটা vertex ∈ G has an edge with each vertex of the K-3 clique.

G' coloring এর graph input instance.

G has a 3-coloring if and only if G' has a K-coloring.

Necessity: G has a 3-coloring

Obvious.

3-coloring করা যায় তাহলে K-3 টা color, প্রত্যেকটা vertex এ clique এ দিব।

QED

Sufficiency:

G' has a K-coloring

prove that G has 3-coloring.

from the construction of G',

clique এর K-3 টা color set আর G এর color set must be different.

∴ # of colors = K - (K-3)

= K - K + 3

= 3

QED.

Complement of a problem :

1 টা prob এর 'yes' ans, আরেকটা prob এর জন্য 'no' হয়ে যায়।

prob-1 : Is it a prime number ?

2 → yes

composite of prob-1 : Is it a composite number ?

complement

2 → No

if prob-i ∈ class P
then prob-i ∈ class P

Does a graph have
2-coloring ?
.... not have
2-coloring ?

if prob-i ∈ NP
then, prob-i can be verified in poly time

prob-i " " " " "

not guaranteed !

instance ⇒ yes ⟶ then obv. গ্রহণ NO 2

This graph does have an Ind set of size k

short proof exist করে।

but

instance

does not have

size k

short proof exist করে না।

যদি NO পাই → পর্যন্ত yes পাওয়া not guaranteed.

prob-X

$X \in P \rightarrow$ class P

$\overline{X} \in P$

↓ complement of X

well, $P \in NP$

$\therefore X \in NP$

$\overline{X} \in NP$ } for those probs who also $\in P$.

Co-NP: যে prob $\in NP$, যার complement ও $\in NP$, সেই prob Co-NP.

$NP = Co\text{-}NP$?

If $NP \neq Co\text{-}NP$, then $P \neq NP$.

contra-positive statement টা prove করতে হবে।

if $P = NP$, then $NP = Co\text{-}NP$.

Assume that, $P = NP$

we have to prove, $NP = Co\text{-}NP$.

set এর proof

$NP \subseteq Co\text{-}NP$

$Co\text{-}NP \subseteq NP$

let, $x \in NP$

$\Rightarrow x \in P$ [As $P = NP$]

$\Rightarrow \overline{x} \in P$

$\Rightarrow \overline{x} \in NP$

$\Rightarrow x \in Co\text{-}NP$ [As $x \in NP$ and $\overline{x} \in NP$]

$\therefore NP \subseteq Co\text{-}NP$

Now, $X \in Co\text{-}NP$

$\Rightarrow \overline{X} \in NP$ [definition]

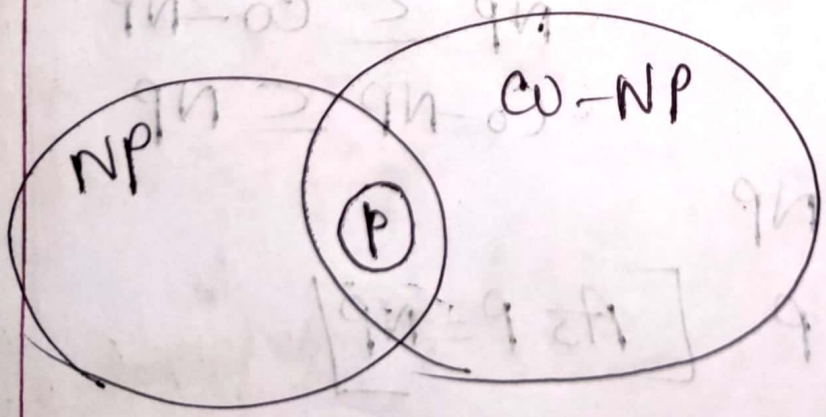
$\Rightarrow \overline{X} \in P$ [P=NP]

$\Rightarrow X \in P$ [def^n of P]

$\Rightarrow X \in NP$ [P=NP]

$\therefore Co\text{-}NP \subseteq NP$

$\therefore$ **NP = Co-NP**



commun relationship bet^n

Lecture-09 15/09/24

Space Complexity: PSPACE

যেসব problem polynomial memory (RAM) দিয়ে solve করা যায়।

Run করার সময় memory:

→ poly. space complexity

P = যাদের time complexity হচ্ছে polynomial, তাদের space ও কখনোই exponential হতে পারে না।

∴ poly time এ expo. space use করা যায় না।

$P \subset PSPACE$

poly space: time যদি expo. হয়, then?

counter: n-bit counter → can count upto 2^n → poly space

2^n → exponential time

Algo যদি expo. time এ নেয়, space complexity can be polynomial.

Th 3-SAT problem can be solved using only a polynomial amount of space.

Proof :

circuit given.

|# of assignments| = exponential

input generation এর জন্য counter

P — NP-complete

PSPACE — PSPACE-complete

এটাও same.

poly time reduction bet^n any two problems.

যে-কোনো prob কে ∈ NP-complete, যে-কোনো prob কে reduce করতে পারি, তাই used the term complete

Exact Algo – DP

Hard problem যেগুলো solve unlikely.
Brute force approach fails.
(আমরা solve করার way জানি না।
B.F দিয়ে করলে ঠিক হবে but ∝ time)

NOT efficient, but 'Faster' version of B.F.
Faster than B.F.

'Exact' → Exact Solution দিবে।
No compromise with solution.
Time complexity = Faster than B.F.

Approach – DP:

Input যদি restricted করে দেই, then poly time এই DP দিয়ে exact sol^n পাওয়া যায়।

Other approaches —

① BKB

② State space Searching

③ Backtracking

: etc

Exact exponential Algo. *

Time complexity তে O use করা হয়।

big-O → constant term গুলো বাদ দেয়া যাবে।

big O* → শুধু expo ই রাখবে।

polynomial term গুলো বাদ দেয়া যাবে।

Parameterized Complexity:

Time complexity শুধু input size এর উপর না, parameter এর উপরও depend করে।

input এর polynomial

parameter এর exponential

$O(n^c 2^k)$

prob size = n

parameter = k

2^k = exponential

crafted input : k যদি ছোট হয়, বা যদি কিছু input এর জন্য দেখাতে পারি 2^k bounded by some constant

then solve হয়ে যাচ্ছে ওরা।

Treewidth — একটা parameter.

co-graph — অনেক graph, co-graph এর জন্য solve করা যায়।