CPSC 421/501 Nov 25, 2021

P vs NP Ch 7 [Sip]

→ real action Ch 9

- how you might show that $P \neq NP$
- what won't work to " " "
- if you can show $P = NP$
  - find an algorithm
  - (somehow prove the existence of an algorithm)

of a certain type - -

The idea:

2 COLOUR

3 COLOUR

4 COLOUR

⋮

Decision problems "over graphs"

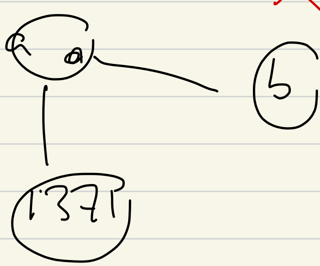
?

(directed graph)

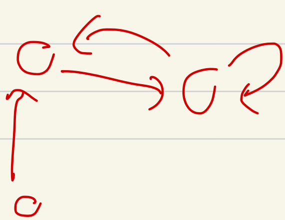
A graph is a collection (V, E) s.t.

V is a finite set, and

E some subset of pairs of vertices



ordered pairs



$2COLOUR = \{ \langle G \rangle \mid$ G is a graph that can be 2-coloured $\}$

$3COLOUR =$ – – – – –

– – 3-coloured$\}$

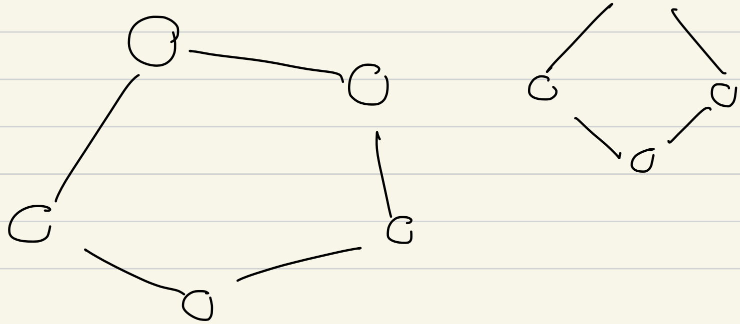
$4COLOUR =$ – – – –

4-coloured$\}$

=

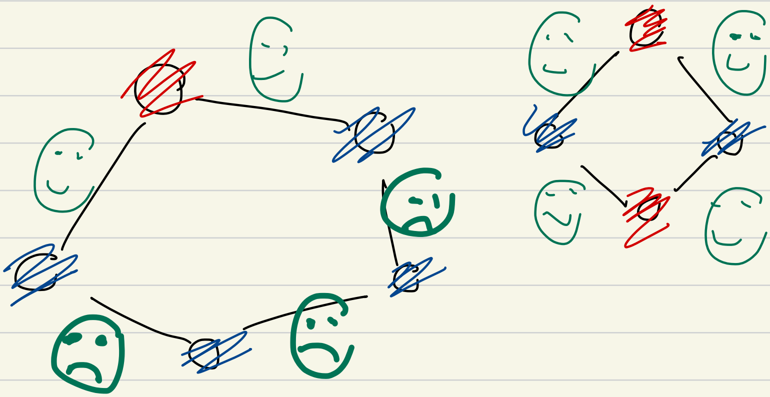
2COLOUR: input G

x   y



Does this graph have a legal 2-colouring

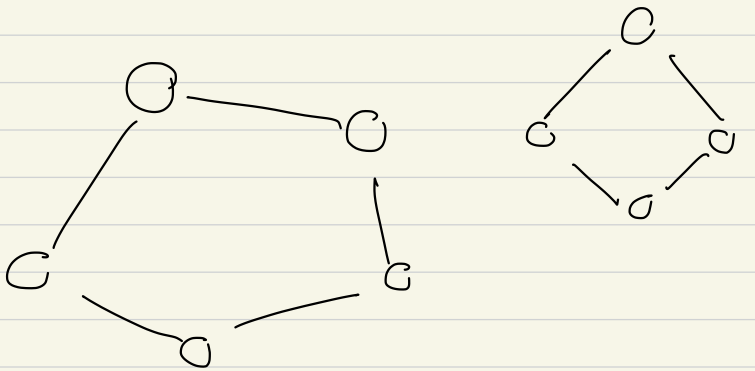
2-colouring:



edge edge has different colours on its endpoints

(2-colouring: $V \to \{\ \text{red}, \text{blue}\ \}$

"legal" means) each edge has distinct colours



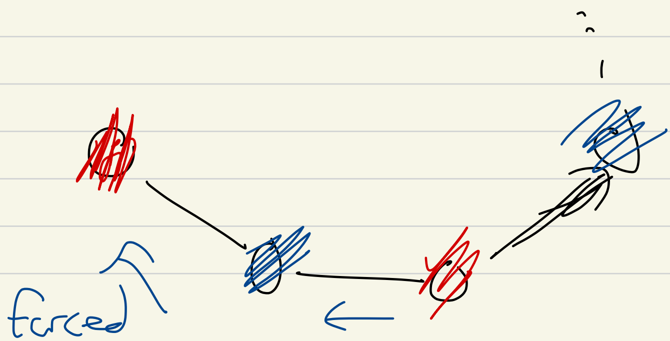
Does this graph have a 2-colouring

Here: it suffices to 2colour

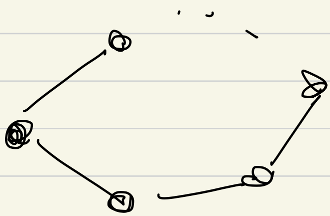
1. A 5-cycle
2. A 4-cycle

=

An n-cycle!



Fact: An n-cycle



can be 2-coloured iff n is even

<u>Obs</u>: If a graph contains an odd length cycle, then it can't be 2-colored

<u>Converse</u>: If G contains only even length cycles

$\Leftrightarrow$ "G is bipartite" $\Leftrightarrow$ G can be 2-colored

This gives a "quick" algorithm to determine if a

graph $\in$ 2COLOUR

a language over some alphabet

language over Σ_{Graphs}

Def: Graph = (V, E)

Def! Standardized graphs

$$V = \{1, 2, \ldots, n\}$$

Or

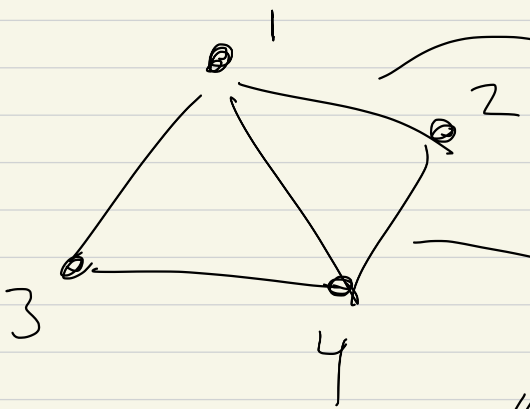
$$V \subset \{a, b, c, \ldots, z, aa, ab, \ldots$$

=

$V \subset$ fixed countably infinite set

$\Sigma_{\text{Graphs}} = \{0, \ldots, 9, \#\}$

G



4 # ### 1 # 2 ##

2 # 4 ##

4 # 3 ##

3 # 1 ##

1 # 4

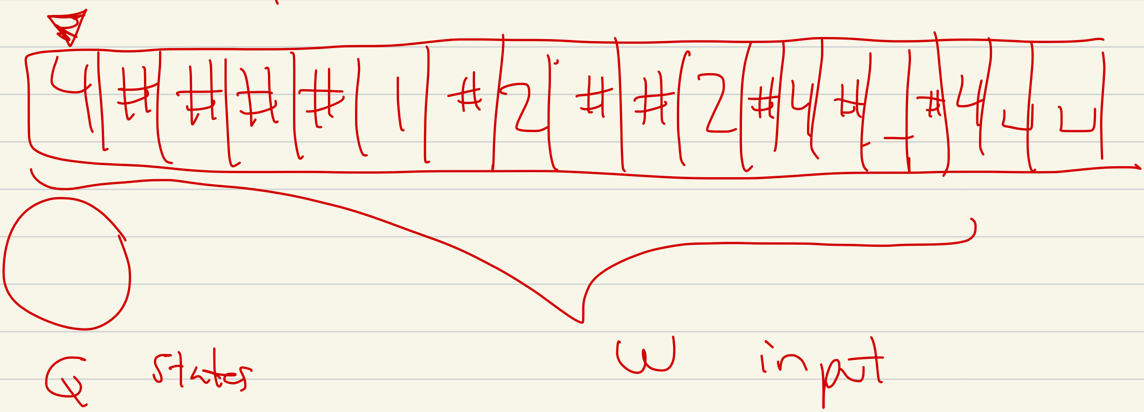
\# of vertices

$\langle G \rangle$ = description of G

Algorithm: You give input:

4####1#2##2#4#_#4

or TM



Algorithm: Input w

Claim: $\exists$ TM, M, s.t. on input w, M runs for at most $\underline{\text{poly}(n)}$ steps, n = length of input; M decides if input $\in$ 2COLOUR poly(n) = less than

$C n^k$ for some

fixed C, k,

but arbitrary n.

=

Break 10:13, we'll give

3 COLOUR at shot ...

=

10:13 – 10:18

CPCS 320

CPCS 420

Poly time decidable languages

2COLOUR

dynamic prog alg

2S~~AT~~

IS A GRAPH CONNECTED

3COLOUR

others

SAT, 3SAT

$$SAT = \{ \langle f \rangle \mid f \text{ is a boolean formula s.t. } f \text{ is satisfiable} \}$$

Boolean formula:

$$f(x_1, \ldots, x_n) =$$

$$\text{not}\Big((X_1 \text{ and } x_2) \text{ or } (X_3) \Big) \text{ or } X_1$$

Boolean formula:

$$\Sigma_{\text{Bool formula}} = \{ (,), \text{not}, \text{or}, \text{and}, X, 0, 1, \ldots, 9 \}$$

$$\langle \text{not}\Big((X_1 \text{ and } x_2) \text{ or } (X_3) \Big) \text{ or } X_1 \rangle$$

$= \text{not} ((X 1 \text{ and } X 2 \ldots$

A Boolean formula

$$f : \{\text{true}, \text{false}\}^n \to \{\text{true}, \text{false}\}$$

written down as a word over $\Sigma_{\text{Bool formula}}$

f is <u>satisfiable</u> if $\exists$ true/false assignment to $x_1, \ldots, x_n$ (where $f = f(x_1, \ldots, x_n)$) s.t. f is true under this assignment

"Naive algorithms" for

3 COLOUR

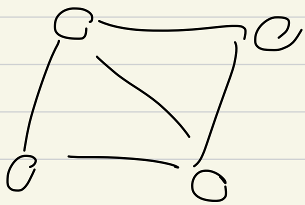
SAT

⋮

---

3 COLOUR :

Problem



← input

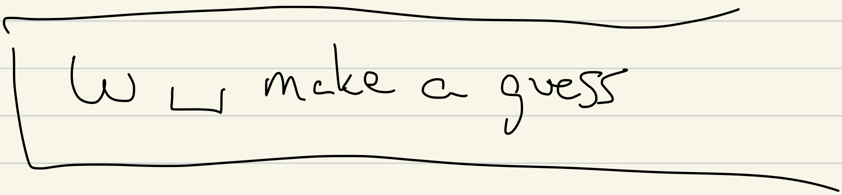
input $\in \Sigma^*_{Graph}$

input size is $|w|$

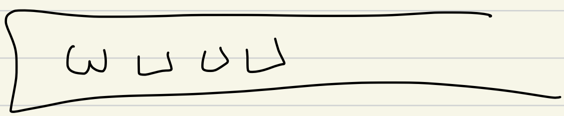
say $4\#\#\ldots\#4 = w$,

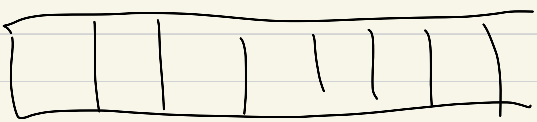
Naively:

1-tape



2-tape





← nice guess

Def: If $L \subset \Sigma^*$ language over Σ, we say that L is *verifiable in poly time* if there is

$\Sigma_{guessing} \supset \Sigma$ and a

TM, M, with input alphabet

Σ_{guess}

s.t.

M runs in poly time of

w for some $g \in \Sigma_{guess}$

on input

$$w \cancel{\sqcup} g \sqcup \sqcup \sqcup \sqcup$$

where $|g| \leq \text{poly}(|w|)$

If for some M, TM

over $\Sigma_{guess} \cup \{sep\}$

s.t. ~~iff~~ L' is the language decided

by TM then

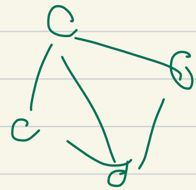
$$w \; sep \; g \in L'$$

$w \in L$ iff for some g (arb long poly size)

and M runs on $w \; sep \; g$

in time $\leq poly(|w|)$

Intution: 3COLOUR



w input

someone gets a "lucky guess" and it works

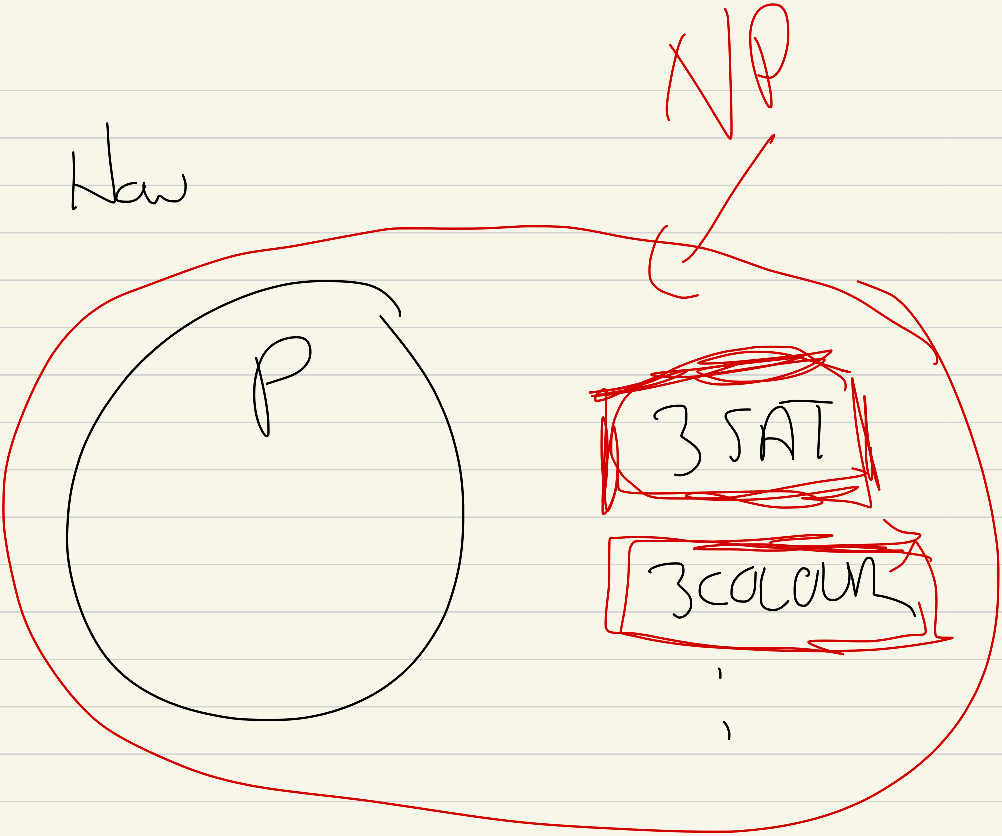
$|\text{guess}| \leq$ poly in $|w|$

time to verify $\leq$ poly in $|w|$

Claim: Poly time verifiable

$\iff$ poly time algorithm in a non-det TM

Now



GRAPH ISOM

NP = poly time verifiable

OR

decidable in poly time by a non-det T.m.