Algorithms

More reductions

Recap

- Slight change on HW:
  You may do *either* #3 or #4.
  (The other is extra credit.)

- HW is due next Monday.
- Final HW - due after break.

P, NP, + co-NP

$P \subseteq NP$

Consider only decision problems:
so Yes/No output

P: Set of decision problems that can be solved in polynomial time.

Ex: - Is x in the ~~lst~~ list? $O(n)$ or $O(\log n)$

- Is there a cut in G of size 100? <-> F-F: $O(VE)$

NP: Set of problems such that, if the answer is yes & you hand me proof, I can verify/check in polynomial time.

Non-determinate poly time

Ex: CircuitSAT: hand me inputs, I can check in $O(n+m)$ time

Co-NP: If answer is no, I can check that in poly time.

Dfn: NP-Hard

X is NP-Hard

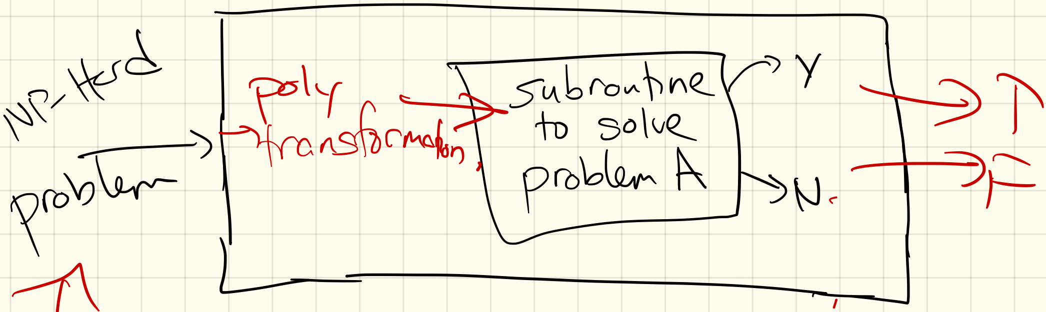
$\Longleftrightarrow$

If X could be solved in polynomial time, then $P = NP$.

So if <u>any</u> NP-Hard problem could be solved in polynomial time, then all of NP could be.

To prove NP-Hardness of A:

Reduce a known NP-Hard problem to A.



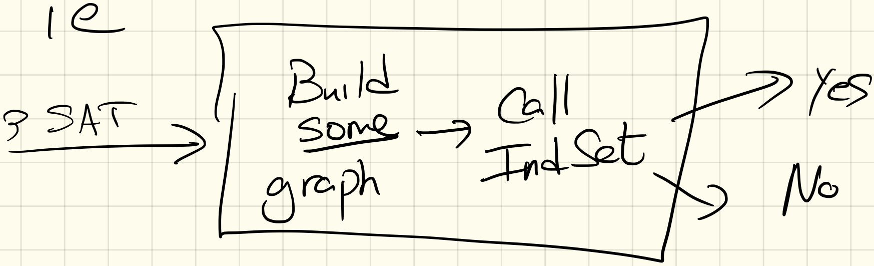
circuitSAT

If tranformation + subroutine for A is poly time, then could solve circuitSAT in that time.

The Pattern:

1) Find an NP-Hard problem, & solve it using unknown problem as a subroutine

ie



Proof:

Need if & only if!

(ie might be some wierd indep. set that doesn't make a SAT)

So far:

1. CircuitSAT - Cook's thm
   (1971)

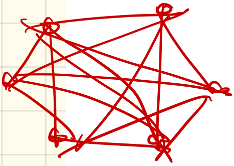
2. SAT
   (Cook 1971)

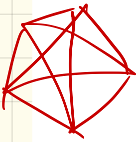
3. 3SAT
   ??

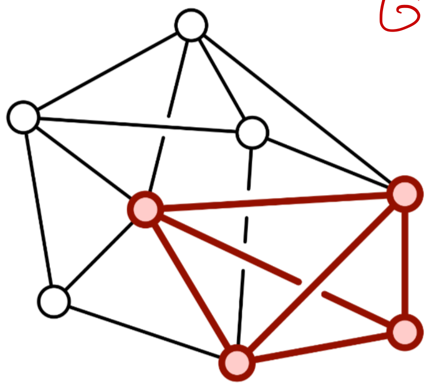
4. Ind. Set in a graph:
   Karp 1972-3
   How? Took 3SAT, & changed it to a graph

Next one: Clique

A clique in a graph is a subgraph which is complete – all possible edges are present.







Given G & k!

A graph with maximum clique size 4.

Try $\binom{n}{k}$ possible cliques

How could we check if G has a clique of size k?

Decision version: Does G have a clique of size k?

Input: G, k

Output: Yes/no

This is NP-Complete:

① In NP. Why?

If you give me k vertices:

Each have vertex list.

For i=1 to k

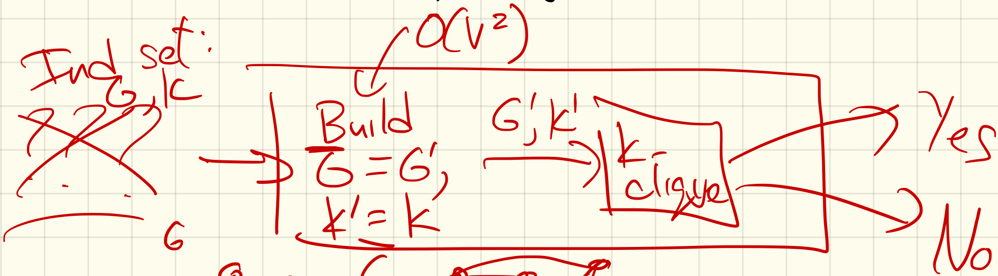
For j = 1 to size of list k

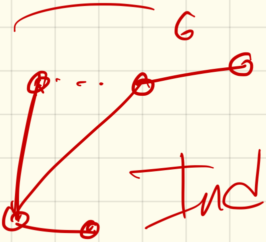
check that (k-1) other vertices are in list

$O(kV^2)$

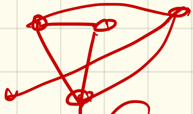
② NP-Hard:

What should we reduce to k-Clique?





$\iff$ $\bar{G}$



Ind Set: Subset with <u>no</u> edges

Input: G, k

Take the <u>complement</u> of a graph, $\bar{G}$:

- If $uv \in E(G)$, <u>not</u> in $E(\bar{G})$
- If $uv \notin E(G)$, $uv \in E(\bar{G})$

<u>Obs</u>: ~~If~~ G has indep set of some k vertices; ~~then~~ $\iff$ G' has a clique on some set.

So:

$$\begin{tikzcd} \text{graph } G \arrow[r, "\overset{O(V^2)}{\cancel{O(n)}}"] & \text{complement graph } \overline{G} \arrow[d, Rightarrow, "\textsc{MaxClique}"] \\ \text{largest independent set} & \text{largest clique} \arrow[l, "\text{trivial}"'] \end{tikzcd}$$

k-clique

Next: Vertex Cover:

A set of vertices which touches every edge in G.

k-Vertex cover (decision version):

Given G & k, does G contain a set of k vertices covering every edge?

In NP:

If you give me k vertices:

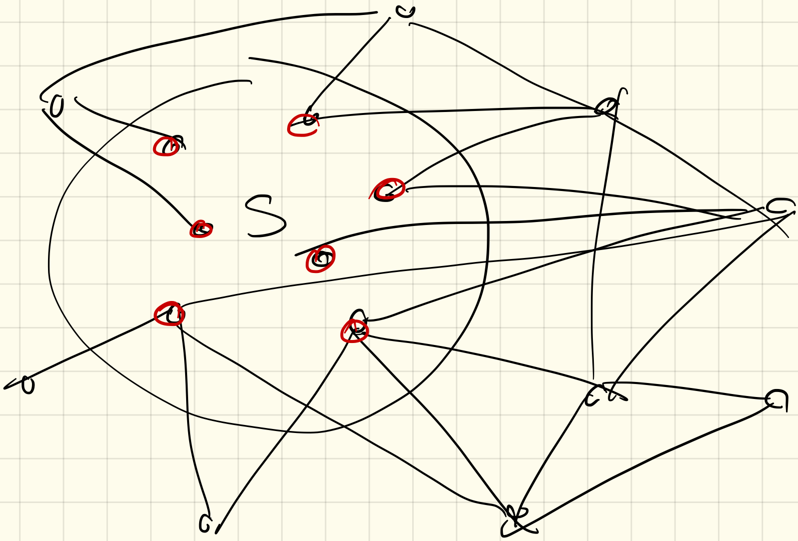
For each edge in G
check if one endpt is in k set

$O(kE)$

NP-Hardness: reduce what?

(propably clique or ind set!)

Key: If S is independent set, what is $V-S$?



All edges can't have 2 endpoints in S

$\Leftrightarrow$ Every edge has ≥ 1 endpt in $V-S$

So simple reduction!

Given G & k to indep. set, ask if $\exists$ vertex cover of size $n-k$.

Set $G' = G$

$k' = n-k$

independent set of k, S $\iff$ VC of size $n-k$, $V-S$

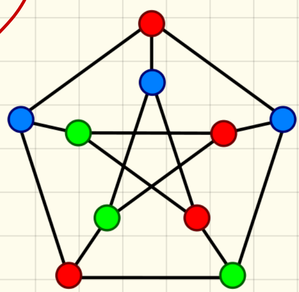
$$
\begin{tikzcd}
 & \boxed{\text{graph } G=(V,E)} \arrow[r, "\text{trivial}"] & \boxed{\text{graph } G=(V,E)} \arrow[d, Rightarrow, "\textsc{MinVertexCover}"] \\
 & \boxed{\text{largest independent set } V \setminus S} & \boxed{\text{smallest vertex cover } S} \arrow[l, "O(n)"']
\end{tikzcd}
$$

Next: Graph Coloring

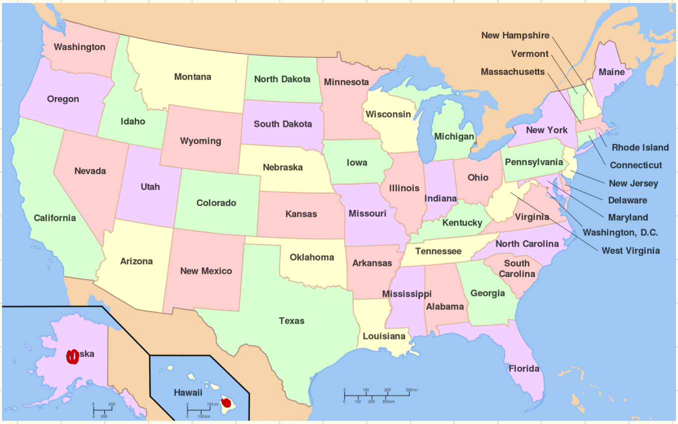
A **k-coloring** of a graph G is a map: $c: V \to \{1, \ldots, k\}$ that assigns one of k "colors" to each vertex so that every edge has 2 different colors at its endpoints.

Goal: Use few colors

$|V|$-coloring is easy 😊



Aside: this is famous!
Ever heard of map coloring?



Famous theorem:

Any planar graph only needs 4 colors.

Thm: 3-colorability is NP-Complete.

(Decision version: Given G, output yes/no)

In NP:

If you give me coloring: $c: V \to \{1,2,3\}$

Loop through edges uv check $c(u) \neq c(v)$

$O(E)$

NP-Hard:

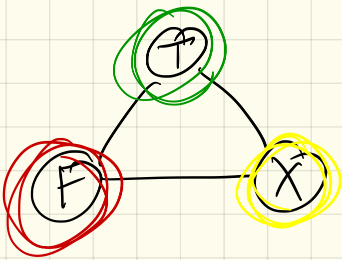
Reduction from 3SAT.

Given formula for 3SAT Φ, we'll make a graph G_Φ.

Φ will be satisfiable $\Longleftrightarrow$ G_Φ can be 3-colored.

Key notion: Build gadgets!

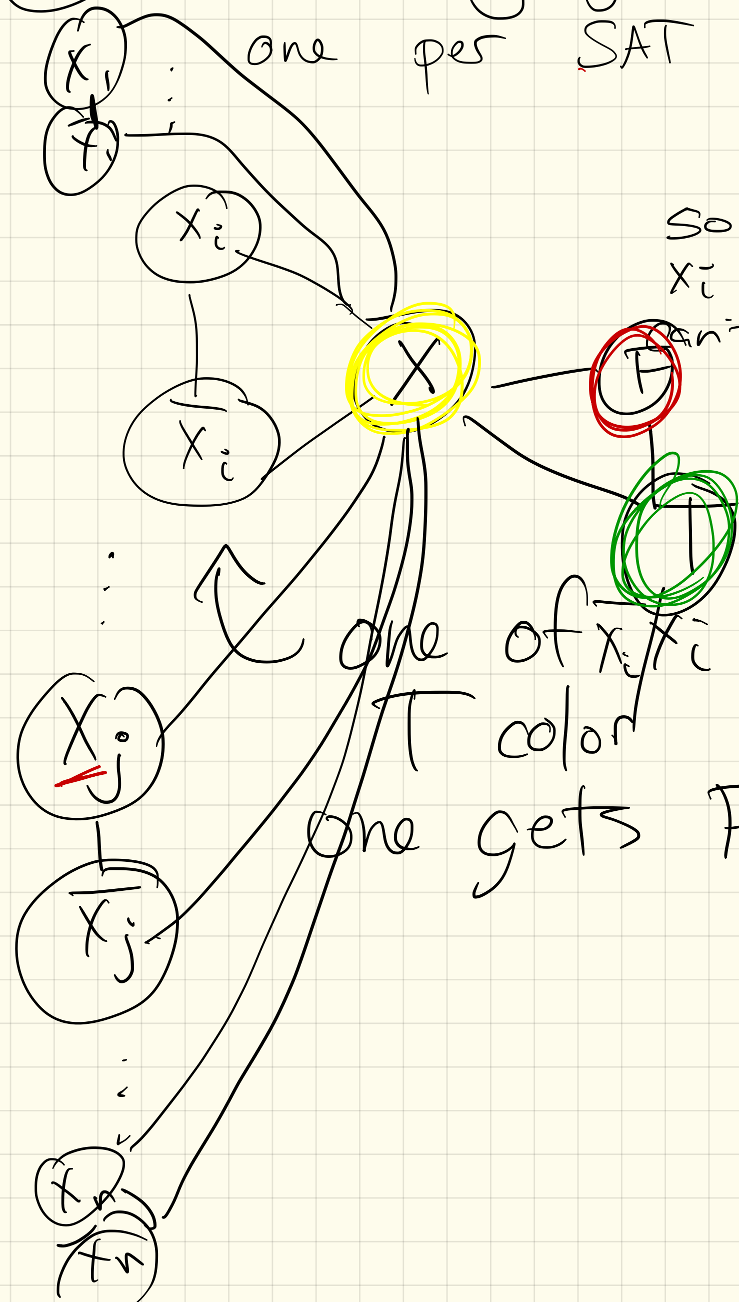
① Truth gadget - one



Must use 3 colors - establishes a "true" color.

② Varicble gadget —

one per SAT variable



so both x_i + $\bar{x}_i$ can't be true

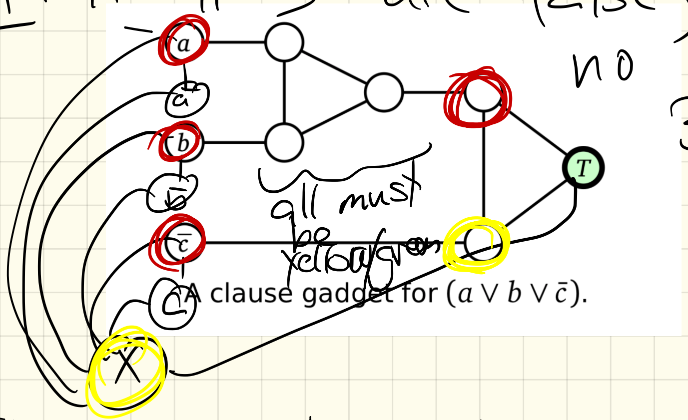
one of x_i/$\bar{x}_i$ gets T color

one gets F color

③ Clause gadget:

For each clause, join 3 of the variable vertices to the "true" vertex from the truth gadget.

Goal: If all 3 are false, no valid 3-coloring



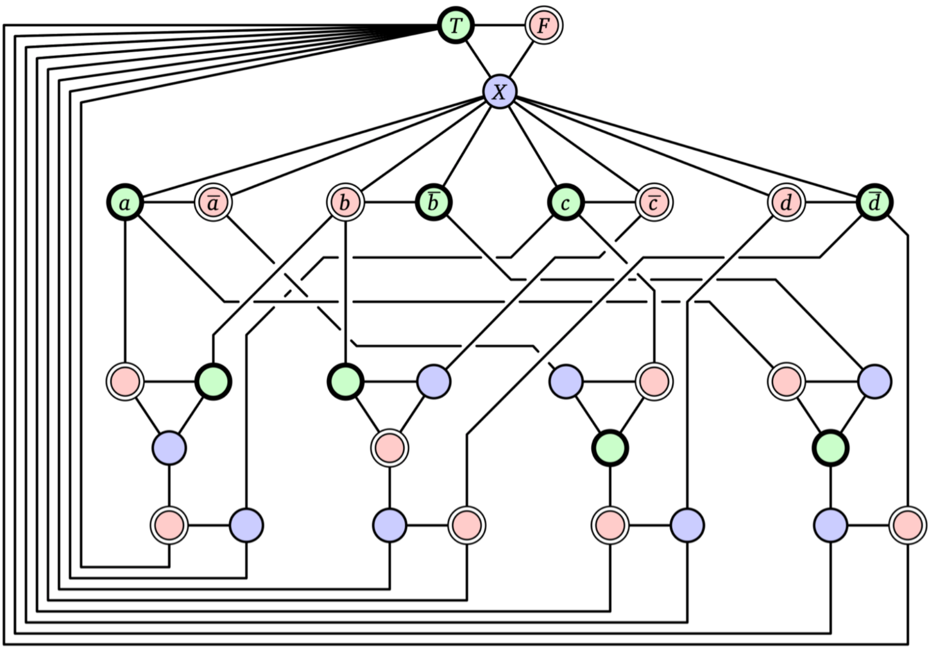
A clause gadget for $(a \vee b \vee \bar{c})$.

Idea: If all inputs are colored False, can't 3-color:

3 colorings of G_{Φ} $\Longleftrightarrow$ Φ is satisfiable

Pf:

Final reduction image:



A 3-colorable graph derived from the satisfiable 3CNF formula
$(a \vee b \vee c) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{d})$

Time to build G_Φ:

n variables

m clause gadgets

↳ $O(1)$ vertices

So:

$$
\begin{tikzcd}
\boxed{\text{3CNF formula}} \arrow[r, "\cancel{O(n)}"] & \boxed{\text{graph}} \arrow[d, Rightarrow, "\textsc{3Colorable}"] \\
\boxed{\textsc{True or False}} & \boxed{\textsc{True or False}} \arrow[l, "\text{trivial}"']
\end{tikzcd}
$$

Next time:

- More reductions
- Plus some non-graph problems!