

Disc Week 9

[1] Intro to P vs. NP

[2] Reductions

Note: Next Week We'll Do Sample Final Problems

K-Color

Input: Graph $G=(V,E)$, undirected, planar

Output: Yes/No, can G be colored w/ k -colors?

(i.e. Vertices assigned 1 of k colors,
no edge (u,v) has same color on u and v)

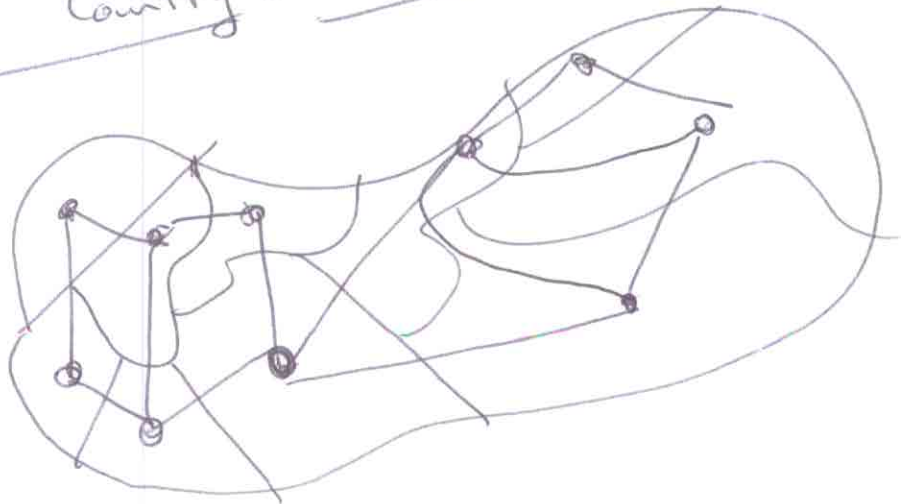
- We've seen how to solve 2-color efficiently with BFS. 1 color is also solvable easily.
- Can you suggest an ^{efficient} algorithm for 3-Color?

IF SO TELL ME

- Note: 4-color for planar graphs just recently solved: 4 Color Theorem, Ans: Always Yes for any planar G .

Even's Country: Map Color: 4

k	Efficient Alg?
1	✓
2	✓
3	?
4+	✓



• 3-color is a member of a group of very hard problems called NP-complete. What is that?

• Defn: A search problem is any problem that has a checking algorithm C , that takes input an instance I and solution S where

$$C(I, S) = \begin{cases} \text{true} & \text{if } S \text{ is a solution to } I \\ \text{false} & \text{if } S \text{ is not a solution to } I \end{cases}$$

and C runs in time polynomial in $|I|$.

Ex: (3-Color)

I = a graph G

S = a potential 3-coloring of G

C = checks each edge to see if colors are the same in S . Runs in $O(|E|)$.

• Defn: NP is the class of all search problems -
e.g. 3-Color, 2-Color, etc...

• Defn: P is the class of all search problems that can be solved in polynomial time,
 $O(n^c)$ for some $c > 0$ fixed,
e.g. 2-Color.

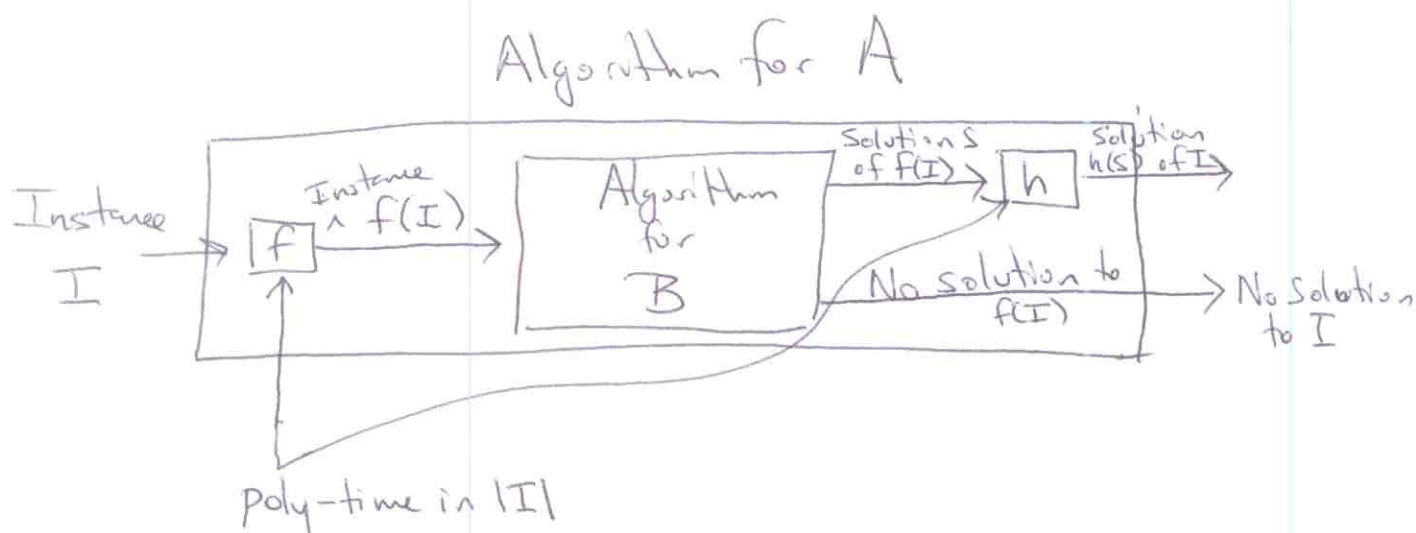
• Note: By definition $P \subseteq NP$

Big Question - Does $P = NP$? ~~log~~ is 3-Color in P?
Most people think not, but nobody knows for sure...

Defn: A search problem is NP-complete if all other search problems reduce to it.

- What does reduce mean?

Defn: Problem A reduces to problem B if the following picture holds:



- Idea is A reduces to B if we can solve A efficiently using an efficient solution to B.

2 Reductions

3SAT

Input: 3CNF formula ϕ

Output: Is there a satisfying assignment for ϕ ? (Yes/No)

$$\text{e.g. } \phi = (x \vee y \vee z) \wedge (x \vee \bar{y}) \wedge (y \vee \bar{z}) \wedge (z \vee \bar{x}) \\ \wedge (\bar{x} \vee \bar{y} \vee \bar{z})$$

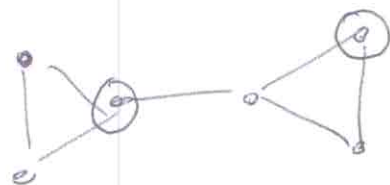
Can we set x, y, z to make ϕ true? No, here.

Independent Set

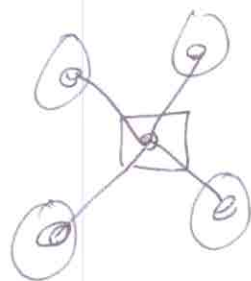
Input: Undirected $G=(V,E)$; K

Output: Does there exist $V' \subseteq V$ such that
 $\forall x,y \in V', (x,y) \notin E$ and $|V'| \geq K$.

e.g.



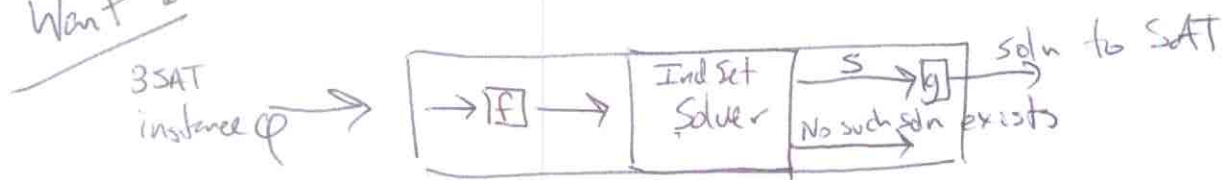
These two nodes
are an independent set of
size 2.



$\square$ is an ind. set of size 1
 O's are " " " " 4

Claim: 3SAT reduces to Independent Set.

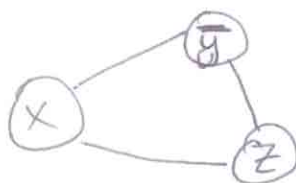
Want:



Idea: Construct a G whose independent set
is the truth assignment for φ

φ = as before in example

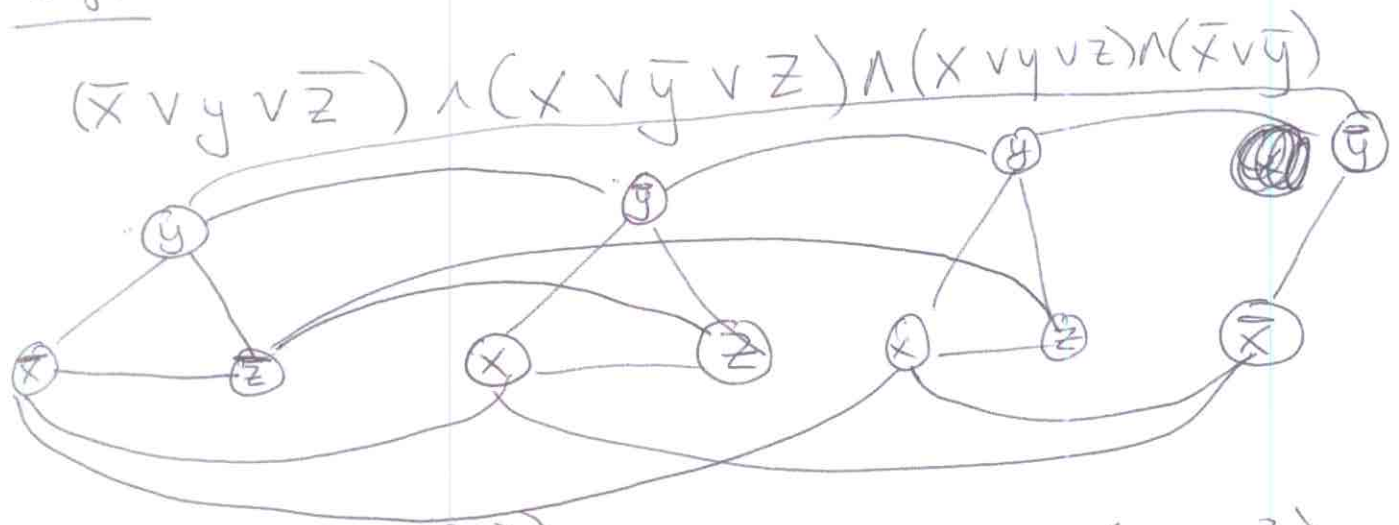
① for each clause make Δ e.g. $(X \vee \bar{y} \vee Z)$



② connect opposite literals etc.



e.g.



Note: Construction is poly-time since it takes $O(\#clauses^2)$

Let K be the # of clauses, then $O(K^2)$.

2 things to show we've solved 3SAT

1) If we find an ind. set of size at least K in G , then ϕ has an assignment that makes ϕ TRUE.

• Since there are only k clauses in ϕ and one variable must be selected from each clause and $x, \bar{x}$ cannot be selected simultaneously, this is our truth assignment.

e.g. if $\bar{x}$ is in the ind. set set $x = F$.

2) If there is no ind. set of size at least k in G , then ϕ is unsatisfiable.

$\equiv$ prove $\Leftarrow$ contrapositive: If ϕ satiable, then G has an ind. set of size $\geq k$.

• Choose the one literal in each clause that could satisfy the clause, ~~be set to~~ ^{and hence} to satisfy ϕ , and add to indep. set.

There must be k of them since ϕ is satisfiable. $\square$

We've shown, if we could solve ind. set efficiently, then we could also solve 3SAT efficiently.