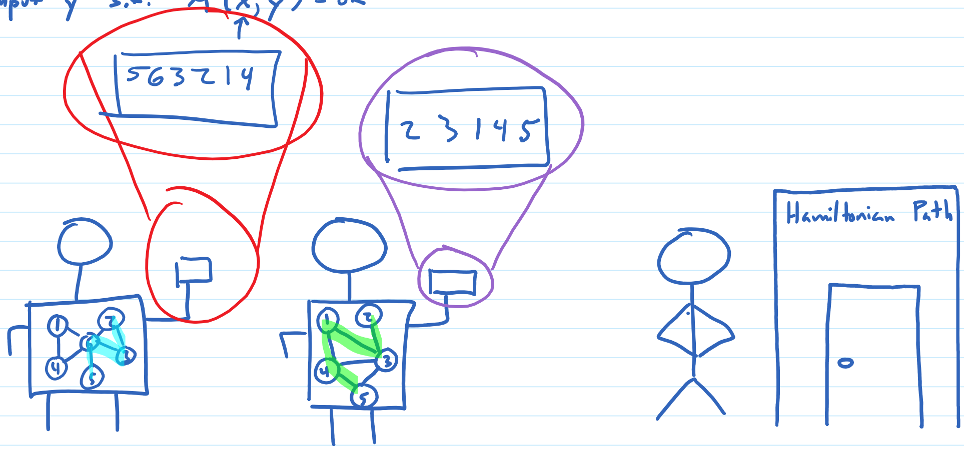
P vs NP

P: set of decision problems w/ poly-time solutions

NP: set of decision problems w/ poly-time verification algorithms

Problem A is in NP means there is a poly-time verification alg A' s.t. the set of inputs x to A for which $A(x) = \text{YES}$ is exactly the set of 1st inputs to A' for which there is some poly-size 2nd input y s.t. $A'(x, y) = \text{ok}$



HP-VERIFY(G, p)

1) check if each edge (u,v) in P is in $G.E$
2) check if $|p| = |G.V| - 1$
3) check if no repeats in p
4) if YES/YES/YES then return OK
   else return NOT OK

HP $\in$ NP

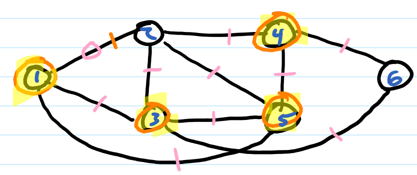
input to VC

VC-VERIFY(G, k, C)

evidence certificate

1) check that each edge in $G.E$ has one endpoint in C — poly
2) check $|C| \le k$ — poly
3) $C \subseteq G.V$ — poly

poly



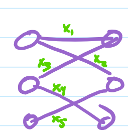
VC $\in$ NP

v = (HVN PVG JAC NUQ FRA GRU BOG HVN) $k = 28000$ $k = 10000$

tour = HVN FRA PVG JAC NUQ BOG GRU HVN TSP $\in$ NP

Given φ, $x_1, \ldots, x_n$

3-SAT-VERIFY: evaluate formula using values $x_1, \ldots, x_n$ 3-SAT $\in$ NP



$((x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)) \wedge (\ldots)$

↳ given Boolean formula φ in 3-CNF, determine if $\exists$ assignment of T/F to all vars that makes φ true

$x_1 = T$ $x_2 = F$ $x_3 = T$

$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3)$

T ∨ T ∨ T   T ∨ F ∨ F

T ∧ T

$\varphi = C_1 \wedge C_2 \wedge C_3 \wedge \cdots \wedge C_n$

where each $C_i = (t_1 \vee t_2 \vee t_3)$

↳ x_i or $\neg x_i$

NP-complete: hardest problems in NP

A is NP-complete means

1) $A \in NP$
2) for all $B \in NP$, $B \le_p A$

THM: If $X \in NP$ and Y is NP-complete and $Y \le_p X$ then X is NP-complete

Proof: [for all $B \in NP$, $B \le_p X$]

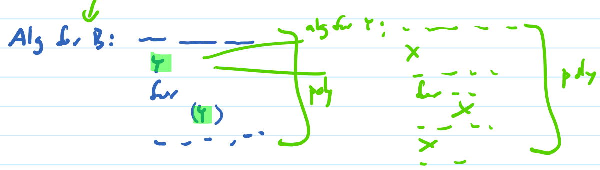
Let $B \in NP$

Then $B \le_p Y$ — Y is NP-complete, def NP-c part 2

Also $Y \le_p X$

so $B \le_p Y \le_p X$ and so $B \le_p X$ — transitivity of $\le_p$

$\therefore$ for all $B \in NP$, $B \le_p X$

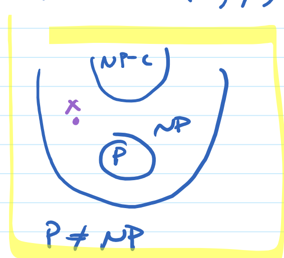


\$1,000,000 question (Millennium prize)

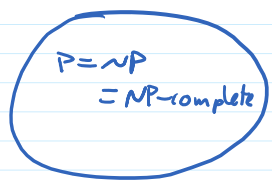
Is $P = NP$???

THM: $P \subseteq NP$

Proof: Let $X \in P$ — X-VERIFY(x, y): if $X(x) = $ YES return OK
else return NOT OK



probably this
(but we're not sure)



$P = NP$

To show $P \ne NP$, it is sufficient to find some super polynomial lower bound for some $X \in NP$

To show $P = NP$, it is sufficient to find some polynomial time algorithm for NP-complete X

let $B \in NP$
then $B \le_p X$ — X is NP-complete
so $B \in P$ — prev thm

SAT : Given Boolean formula φ, does it have a satisfying assignment?

$$((x_1 \vee x_2) \rightarrow (x_1 \wedge x_3)) \wedge \sim x_1 \wedge \sim x_3$$

3-SAT : Given Boolean formula φ in 3-CNF form, is it satisfiable?

$$(\sim x_1 \vee \sim x_2 \vee \sim x_3) \wedge (\sim x_1 \vee \sim x_2 \vee x_3) \wedge (\sim x_1 \vee x_2 \vee \sim x_3) \wedge (\sim x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \sim x_2 \vee \sim x_3) \wedge (x_1 \vee x_2 \vee \sim x_3) \vee (x_1 \vee x_2 \vee x_3)$$

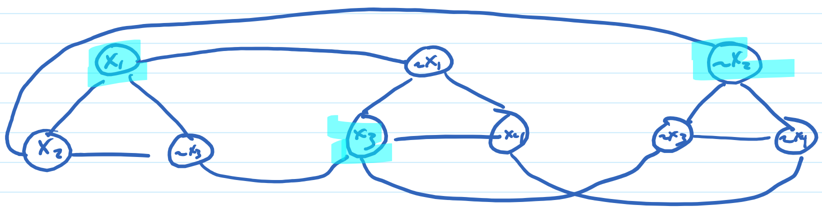
3-SAT $\leq_p$ IND-SET

Goal: given φ in 3-CNF, construct G, pick k s.t.

φ is satisfiable $\Leftrightarrow$ G has IS of size $\geq k$

Alg for 3-SAT:
1) get φ
2) construct G, k] poly
3) output IS(G,k)

$$(\overset{T}{x_1} \vee x_2 \vee \sim x_3) \wedge (\sim x_1 \vee \overset{T}{x_3} \vee x_4) \wedge (\overset{T}{\sim x_2} \vee \sim x_3 \vee \sim x_4)$$



$k = 3$ (# of clauses)

$\Leftarrow$ Suppose G has IS of size $\geq k$.

Then G has IS w/ one vertex per clause/triangle — k triangles, can't have ≥ 2 from any one since there is an edge between them

Setting corresponding terms to T is a valid truth assignment — $x_i, \sim x_i$ not both selected b/c edge between

selected terms make each clause T and so φ T — one T vert/term per clause

$\Rightarrow$ Suppose G has satisfying assign

let S = set of verts whose terms are T

$|S| \geq k$ — ≥ 1 T term per clause (if some clause all F then φ is F)

let $S' = S$ with 1 vert per $\triangle$ chosen arbitrarily

$|S'| = k$ — ≥ 1 T term per clause becomes $= 1$ T term per clause

no edge in any $\triangle$ has both endpoints in S' — 1 term/clause $\rightarrow$ 1 vert per $\triangle$

no edge between $\triangle$ has both endpoints in S' — such edges connect $x_i, \neg x_i$, which cannot both be T so can't both be in S or S'

no edge in G has both endpoints in S' — last 2 stmts cover all the possible edges

no edge in G has both endpoints in S'	last 2 stmts cover all the possible edges
S' is an IS of size $\geq k$	def IS, prev stmt about size