NP Completeness:
a Machine model
Recap: 3COLOR is equivalent to SAT.

Defn: let A & B be decision problems. We say A reduces to B (written \( A \leq_p^m B \)) if there exists a polynomial-time computable function \( f \) s.t.

\[
x \in A \iff f(x) \in B
\]

"polynomial-time computable" = algorithm for \( f \) runs in \( O(n^k) \) time for some \( k \in \mathbb{N} \).

\( A \leq_m B + \text{fast algorithm for } B \) \implies "fast" algorithm for A
Suppose \( A \leq_m^p B \).

\[ x \in A \Rightarrow f(x) \in B \]

\[ y \in A \Rightarrow f(y) \in B \]
Problems equivalent to SAT & 3COLOR are NP-complete. 

Equivalence is transitive: 
\[ A \leq_m B \land B \leq_m C \Rightarrow A \leq_m C \]

Thousands of problems are NP-complete.

Capture many important optimization problems, eg:

- Clique
- Vertex Cover
- Traveling Salesman Problem
- Partition
- 3D Matching.

[Will give you the flavor of a range of NP-complete problems.]

[Will define later]
Clique:

Input: undirected graph $G = (V, E)$
- number $k$

Question: Does $G$ have a $k$-clique?

$k$-clique = $k$ vertices in $V$ s.t. any two vertices are connected by an edge.

3-Clique

4-Clique

5-Clique
Vertex Cover

Input: an undirected graph $G = (V, E)$
a number $k$

Question: does there exist a subset $V' \subseteq V$ s.t.
for all edges $(u, v) \in E$, either $u \in V'$ or $v \in V'$?

$|V'| \leq k$, and
Traveling Salesman Problem

Input: an undirected graph $G = (V, E)$

- a weight function $w : E \rightarrow \mathbb{R}^+$
- a bound $B$

Question: Does there exist a simple cycle in $G$ that visits every vertex exactly once s.t. the sum of the edge weights of the edges in the cycle is $\leq B$.
Partition

Input: \( n \) number \( a_1, \ldots, a_n \in \mathbb{Z}^+ \)

Question: Does there exist a subset \( S \) of the numbers such that:
\[
\sum_{i \in S} a_i = \sum_{i \notin S} a_i
\]

I.e., pick a subset of the numbers such that the sum of the numbers is exactly half of the total sum.
3-Dimensional Matching

Input: disjoint sets \( W, X, Y \) s.t. \( n = |W| = |X| = |Y| \)

\[ M \subseteq W \times X \times Y \]

\[ M = \{ (w, x, y) \mid w, x \text{ and } y \text{ are "compatible"} \} \]

Question: does there exist \( M' \subseteq M \) s.t. \( |M'| = n \) and no two elements of \( M' \) agree in any coordinate.

\[ W = \{ w \mid \exists x \in X \text{ and } \exists y \in Y \text{ such that } (w, x, y) \in M' \} \]

\[ X = \{ x \mid \exists w \in W \text{ and } \exists y \in Y \text{ such that } (w, x, y) \in M' \} \]

\[ Y = \{ y \mid \exists w \in W \text{ and } \exists x \in X \text{ such that } (w, x, y) \in M' \} \]

Each \( w, x, y \) appears exactly once
$P =$ decision problems that can be solved by some algorithm that runs in time $O(n^k)$ for some constant $k$.

$NP =$ decision problems that can be verified in $P$.

= "non-deterministic" polynomial time.

$P=NP$ means there is a free lunch.

Check this is true for Clique, VC, 3DM, partition, etc.
Working definition of NP

A decision problem $A \in \text{NP}$ if

1. $\exists B \in \text{P}$
2. $\exists k \in \mathbb{N}$

$x \in A \iff \exists y, |y| \leq |x|^k, \text{ s.t. } (x, y) \in B.$

Note: some properties are difficult to verify.

$\{(G, k) \mid \text{the largest clique in } G \text{ has } \leq k \text{ vertices}\}$

= $G$ does not have cliques $> k$
**Defn:** A decision problem $X$ is NP-complete, if

1. $X \in \text{NP}$
2. for all $Y \in \text{NP}$, $Y \leq^p X$

**Cook's Theorem [1971]:** SAT is NP-complete.

How to show that a new problem $\Omega$ is NP-complete.

1. Show $\Omega \in \text{NP}$
2. Show $\text{SAT} \leq^p \Omega$

↑ or some other known NP-complete problem.
Example: Vertex Cover (VC)

\[ VC = \{ (G, k) \mid \exists V' \subseteq V, |V'| \leq k \text{ and for all } (u, v) \in E \text{ either } u \in V' \text{ or } v \in V' \} \]

Show \( VC \in \text{NP} \). Guess \( V' \subseteq V \), check each edge in \( E \).

Much easier than showing \( VC \leq^P \text{m} \in E \).
$3SAT \leq^P m VC$

\[ \emptyset = (u_1 \lor \overline{u}_3 \lor \overline{u}_4) \land (\overline{u}_1 \lor u_2 \lor \overline{u}_4) \land (u_2 \lor \overline{u}_3 \lor u_4) \]

\[ \emptyset \text{ has } n \text{ variables } \& \text{ } m \text{ clauses} \]

\[ G \text{ has } 2n + 3m \text{ nodes } \& \text{ } n + 6m \text{ edges} \]

\[ k = n + 2m \]

**Claim:**

\[ \emptyset \in 3SAT \iff G \text{ has a vertex cover } \forall \text{ } k \text{ vertices} \]
Cock's Theorem in 20 minutes

Hand Wavy Part

Turing Machines

- In each step, a TM M can read one symbol of the tape under the tape head, enter a new state, replace the symbol underneath the tape head and move the tape head left or right.
Turing machines

- Transition function
  \[ S: \Sigma \times \Gamma \rightarrow \Sigma \times \Gamma \times \{L, R\} \]

- An input string \( x \) is accepted by a TM \( M \) if \( M \) starting in the start state \( q_0 \) and the tape head on the leftmost tape cell \( x \) on the tape, enters a unique accepting state \( q_f \) after a finite number of transitions.

- \( x \in L(M) \) if \( x \) is accepted by TM \( M \).
Church-Turing Thesis:
If $A$ is a "computable" set, then $A = L(M)$ for some Turing machine $M$

Robustness of TM's
extra heads, tapes, ... do not add computational power to TM's.

TM's & running time:
If $A \in P$ via the RAM model
then $A = L(M)$ for some TM that makes a polynomial number of transitions.
Representing TM configurations

"instantaneous description" = ID

$x|y$ means

Tape holds $xy$. Tape head reading first symbol $x$. Machine $M$ in state $q$. 
NOT SO HAND WAVEY PART

Tableau: visual aid for thinking about a sequence of IDs

Working defn of NP:

$A \in NP \text{ if } \exists B \in P \text{ and a polynomial } q(\cdot) \text{ s.t. }$

$x \in A \iff \exists y \in \Sigma^*, |y| \leq q(|x|) \land (x, y) \in B.$

Thus, $x \in A \iff$

$\exists \text{ a legal tableau}

\text{Starting with ID } \#0x\#y

\text{s.t. } M \text{ enters the accepting state } \#acc
FIGURE 7.8
A tableau is an $n^k \times n^k$ table of configurations
Use Boolean formulas to check a tableau is legal.

Each cell can hold a state, a tape symbol, or a #. Let $C = \mathcal{Q} \cup \Gamma \cup \{\#\}$.

Each cell indexed by $i, j$, $1 \leq i \leq n$ and $1 \leq j \leq n$.

For each cell $i, j$ and each symbol $s \in C$

$X_{i,j,s}$ is true "means" cell $i, j$ holds symbol $s$. 

NOT HAND WAVY PART
Enforce that each cell has a symbol

\[ \varphi_{\text{cell1}} = \bigwedge_{i,j} \bigvee_{s \in \mathcal{C}} X_{i,j,s} \]

Enforce that each cell has no more than one symbol

\[ \varphi_{\text{cell2}} = \bigwedge_{i,j} \bigwedge_{s,t \in \mathcal{C}} \left( \overline{X_{i,j,s}} \lor \overline{X_{i,j,t}} \right) \]

Enforce that initial configuration is \( \varphi_{\text{start}} \neq y \)

\[ \varphi_{\text{start}} = X_{1,1,1} \land X_{1,2,\#} \land X_{1,3,} \land X_{1,4,} \land X_{1,n+2,} \land X_{1,m+2,} \land \ldots \land X_{1,n+2,} \land X_{1,n,} \land \# \]

where \( n = 1 \times 1 \), \( m = 1 \times \#y \)

This ensures the first line of the tableau is

\[ \# \# \# \# \# \# \# \# \# \ldots \# \# \]

for some \( y \).
Enforce that M entered the accepting state

$$\phi_{acc} = \bigvee_{i,j} X_{i,j,acc}$$

Enforce that line i+1 of the tableau follows from line i.

Observation: Only need to check that all 2x3 "windows" are legal.

LEGAL = set of all legal 2x3 windows \( \subseteq C \times C \times C \times C \times C \times C \times C \times C \)

Note: \(|\text{LEGAL}|\) is finite & constant.

$$\phi_{move} = \bigwedge_{i,j} \text{2x3 window at index } i,j \text{ isLEGAL}$$

$$= \bigwedge \bigwedge_{(a_1, \ldots, a_6) \in \text{LEGAL}} \left( X_{i,j,a_1} \lor X_{i,j,a_2} \lor X_{i,j,a_3} \lor X_{i-1,i+1,a_4} \lor X_{i,i,j+1,a_5} \lor X_{i+1,i+1,a_6} \right)$$

i.e., a legal window a \( i,j \) has one entry different from every illegal window.
FIGURE 7.9
Examples of legal windows

FIGURE 7.10
Examples of illegal windows
Claim: \( x \in A \)

\( \iff \exists \) a legal tableau starting with \( \phi_0 \times \#y \)

\( \iff \phi_{cell_1} \land \phi_{cell_2} \land \phi_{start} \land \phi_{acc} \land \phi_{move} = \phi \)

is satisfiable.

Note: \( \phi \) is in conjunctive normal form.