NP vs P

$P = \text{class of problems for which efficiently find a solution in polynomial time}$

$NP = \text{class of problems for which efficiently verify a given solution}$.

$NP \not= P$
Rudrata Cycle

**Input:** Graph $G = (V, E)$

**Solution:** Find a cycle that visits every vertex exactly once

Rudrata cycle $\in$ NP

**Proof:** Verify $(\text{Graph } G = (V, E), \text{ a cycle } C)$

Check if $C$ is a Rudrata cycle in $G$. 
Factorization $\in$ NP

Input: An $n$-bit integer $N$

Sol: Some $p, q > 1$ integers such that $p \cdot q = N$.

Solution Input

Verify $(p, q, N)$

$p \cdot q = N$

Factorization $\notin P$ (general belief)
A problem $A$ is NP-complete if every problem in NP reduces to $A$. 

**Def:**

$NP$-complete problems $\rightarrow$ 3-COL, Hamiltonian Cycle, TSP

$\Rightarrow$

- $\lor$
  - Factorisation
  - Break RSA

$\Rightarrow$

- $\lor$

$\Rightarrow$

- $\lor$

- MST, Shortest Paths

Halting $\Rightarrow$
**Reductions:**

**Def:** problem \( A \) \( \leq_p \) problem \( B \)  
\( \Rightarrow \) reduction can take polytime

Problem \( A \) reduces in polytime to problem \( B \)

If you can use an algorithm for \( B \) to solve \( A \)

\( \text{Problem } A \) \( \leq_p \) \( \text{Problem } B \)

\( \text{Matching} \) \( \leq_p \) \( \text{MaxFlow} \)
HAMILTONIAN
RUDRATA CYCLE

Input: GRAPH $G = (V, E)$

Solution: Find a cycle that visits every vertex exactly once

HALF - CYCLE

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

Solution: Find a cycle that visits half the nodes $|V|/2$
\[ A \leq B = \text{using an algorithm for } B \text{ to solve } A \]

\[ \downarrow \text{Half-Cycle} \]

**Algorithm for B = Half Cycle**

**Input to A**
\[ G = (V,E) \]

**Input to B**
\[ G' = (V',E') \]

**Reduction**
\[ G' = G \cup \{ \text{n disjoint vertices} \} \]

**Solution to B**
\[ \Rightarrow \text{Solution to A} \]

**No solution to B**
\[ \Rightarrow \text{No solution to A} \]

1) Describe reduction algo
2) Solution to B \(\Rightarrow\) Solution to A efficiently
3) No soln to B \(\Rightarrow\) No soln to A.
Half cycle in $G'$  \[\Rightarrow\]  Cycle in $G$

Half cycle in $G'$  \[\Rightarrow\]  Cycle in $G$

Hydra cycle in $G$
NP-complete problems

1) All of them are reducible to one another

\[ 3\text{COL} \leq_p \text{Rudrata Cycle} \]

2) "Hardest problems within NP"

If a polytime algo for one of them gives a polytime algo for all of NP

\[ \text{NP} = \text{P} \]
**Theorem:**

Circuit SAT is NP-complete

Mother of all NP-completeness

**Circuit SAT**

**Input:** A boolean circuit $C$ with $n$ boolean inputs $(x_1, x_n)$ & one output.

**Sol:** An input assignment $x$ s.t. output of $C = 1$. 

\[
\text{A} \quad \text{And} \\
\text{Or} \quad \text{And} \\
\text{A} \quad \text{A} \quad \text{Or} \\
x_1, x_n, x_n
\]