NP vs P

P = class of problems for which efficiently find a solution

↓ polynomial time

MST
ED
Shortest Paths

NP = class of problems for which efficiently verify a given solution.

$NP \stackrel{?}{=} P$

HAMILTONIAN

RUDRATA CYCLE

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

SOLUTION: Find a cycle that visits every vertex exactly once

RUDRATA CYCLE $\in$ NP

Proof: VERIFY (Graph $G = (V, E)$, a cycle C)

Check if C is a Rudrata cycle in G.

FACTORIZATION $\in$ NP

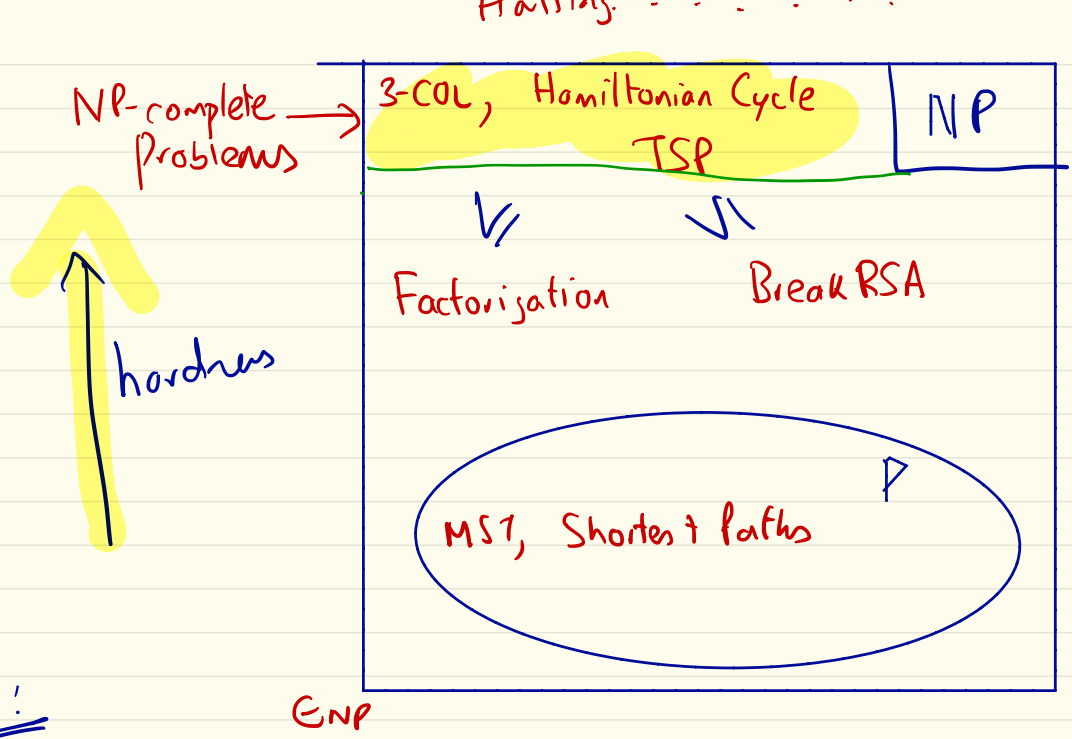
INPUT: An n-bit integer N

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

Verify ($\overbrace{p, q}^{\text{Solution}}$, $\overbrace{N}^{\text{INPUT}}$)

$p \cdot q = N$

FACTORIZATION $\notin$ P (general belief)



Def:

A problem A is NP complete if every problem in NP reduces to A.

REDUCTIONS:

Def: problem A $\leq_P$ problem B → reduction can take polytime

problem A reduces in poly time to problem B

if

you can use an algorithm for B

to solve A

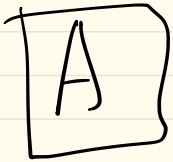
problem A "Matching" $\leq_P$ problem B MaxFlow

HAMILTONIAN

RUDRATA CYCLE

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

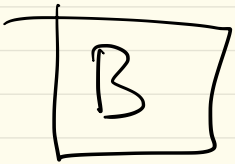
SOLUTION: Find a cycle that visits every vertex exactly once



HALF-CYCLE

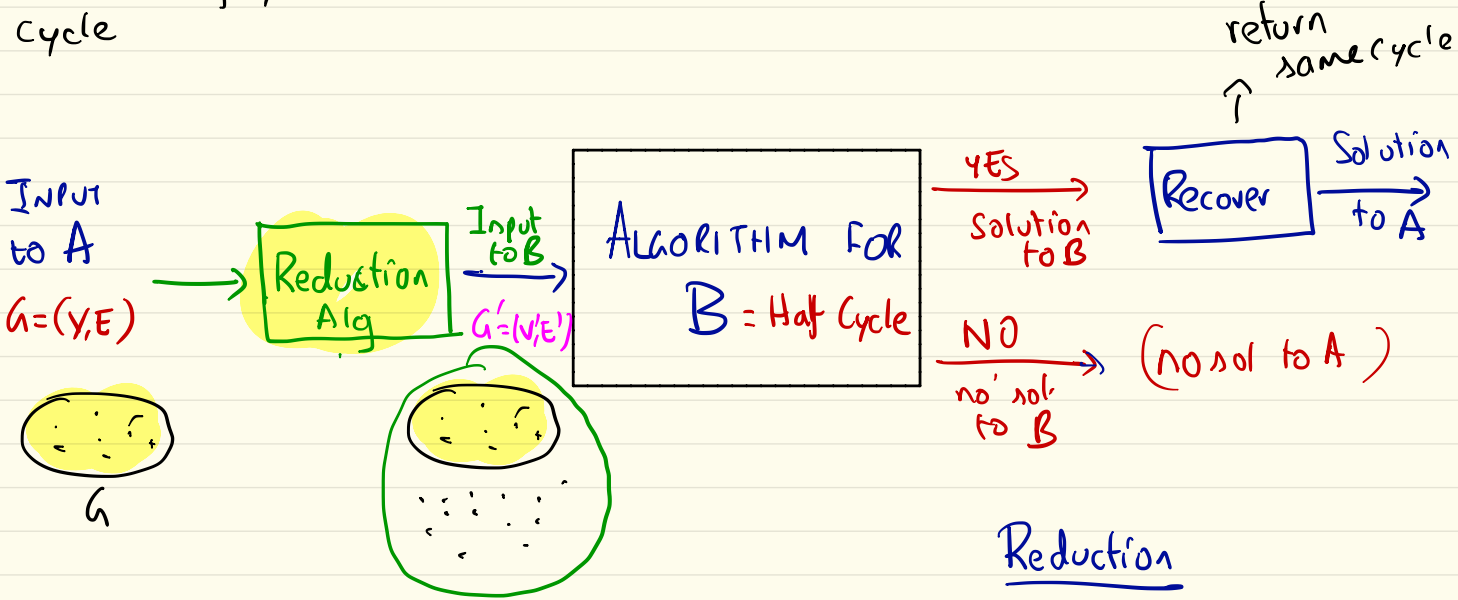
INPUT: Graph $G=(V,E)$

Solution: Find a cycle that visits half the nodes $|V|/2$



$A \leq_P B$ = using an algorithm for B to solve A

↓ Rudrata cycle ↓ Half-Cycle



G' = G ∪ {n disjoint vertices}

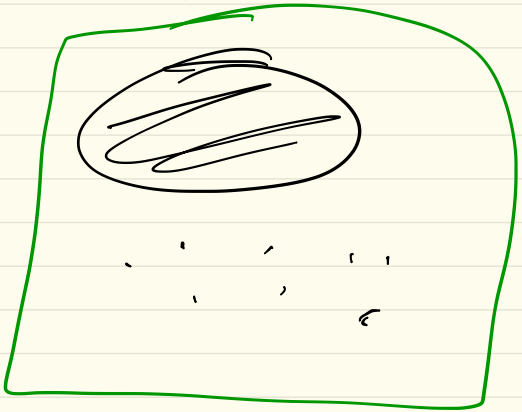
Reduction

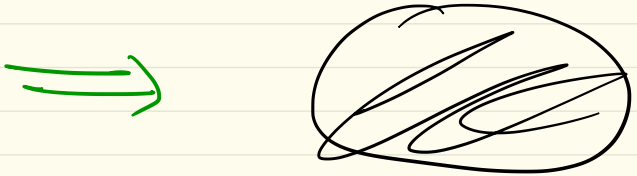
1) Describe reduction algo

2) Solution to B ⟹ Soln to A efficiently

3) No soln to B ⟹ No soln to A.

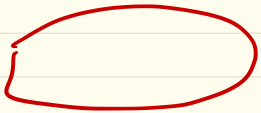
Half cycle in G'





Cycle in G

Rudratacycle in G



Half cycle in G'

NP- complete Problems

1) All of them are reducible to one another

$$3COL \leq_P \text{Rudrata Cycle}$$

$$\leq_P$$

2) "Hardest problems within NP"

⇓

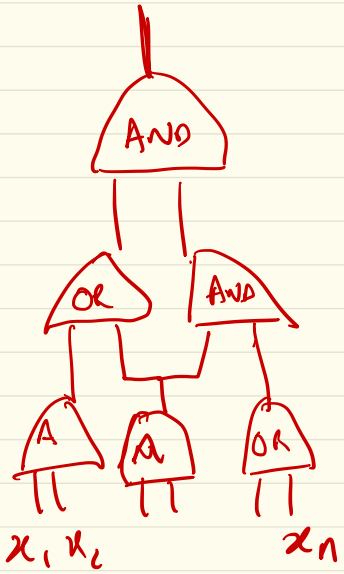
a polytime algo for one of them gives a polytime algo for all of NP

$NP = P$

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 $C = 1$.



Thm:

CircuitSAT is NP-complete

↑

Mother of all NP-completeness