NP vs P Problem

classes of computational problems

P = class of problems for which one can find the solution in polynomial time

= efficiency

NP = class of problems for which verifying the solution is easy

3COL: (#3COL)

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

SOLUTION: A coloring $c: V \to \{R,B,G\}$ such that each edge (u,v), $c(u) \neq c(v)$.

(Number of different colorings)

Claim: $3COL \in NP$

Proof: Verify (Input $G=(V,E)$, Solution $c: V \to R,B,G$)

- for each edge $e=(u,v)$ Check if $c(u) \neq c(v)$.

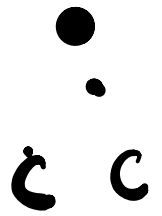
MINIMUM SPANNING TREE $\in P$

VERIFY (INPUT G , TREE T) {

Run MST on G to get a tree T^*

Check cost(T^*) = cost(T)

}



Compute the solution efficiently $\Rightarrow$ Verify the solution efficiently

$P \subseteq NP$.

Problems not in NP:

1) #3COL

2) Games : Does white/black have a winning strategy ?

Hamiltonian Cycle / Rudrata Cycle: $\in NP$

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

SOL: A cycle containing all vertices exacty once

(OPTIMIZATION)

Min TSP (Travelling Salesman Problem): $\notin$ NP Not technically true.

INPUT: Graph $G=(V,E)$ weights $\{w_e\}_{e \in E}$

SOL: A tour of minimum cost

↓

a cycle going through every vertex exactly once.

(SEARCH)

BUDGET TSP: $\in$ NP (search) Budget B.

INPUT: Graph $G=(V,E)$ with $\{w_e\}_{e \in E}$ &

SOL: A tour with cost $\leq B$.

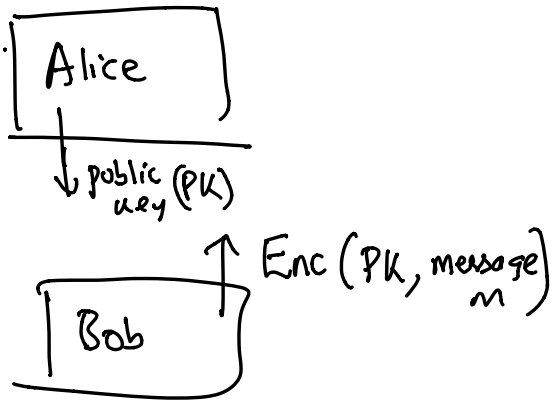
MIN TSP can be solved efficiently $\Rightarrow$ Budget TSP solved efficiently

DECISION-BUDGET TSP $\in$ NP (decision-)

INPUT: Graph G weights, Budget B.

Sol: Is there a tour of cost $\leq B$?

BREAKING RSA: ∈ NP



INPUT: Publickey PK, Ciphertext Enc(PK, m)

SOL: Message m.

Verify ((PK, Enc(PK, m)) , Message m)
INPUT

—— ≠ ——

P ≠ NP ??