Any alg for TSP would reject (6, 4)
would accept (6, 20)
Poly-time non-deterministic decision procedure for $TSP(G,k)$

1) randomly permute vertices to make a tour 
2) compute cost of tour 
3) output YES iff cost $\leq k$

A non-deterministic decision procedure is said to accept an input if some choices lead to YES, if answer should be NO.

repeat = no choices lead to YES.

Problem is in NP if there is a non-deterministic poly-time algorithm that solves it. $TSP \in NP$
**Composite:** given \( n \), determine if \( n \) is composite

1) guess two factors \( a, b \) in \( (2, \sqrt{n}) \) certificate
2) compute product \( a \cdot b \)
3) output \( 1 \) iff \( a \cdot b = n \)

\( \text{Composite} \in \text{NP} \)

**Clique:** given \( G, k \) determine if \( G \) has a \( k \)-clique

\( G \)

\( k \) vertices all connected to each other

has a \( (\text{some}) \) \( 3 \)-clique but no \( 4 \)-clique
CLIQUE $\in$ NP

1) randomly choose $k$ verts $\rightarrow$ certificate
2) check if they form a $k$-clique
3) output $1$ if they do

NP can also be defined as poly-time verifiable

for any input that should be accepted, there is evidence the input should be accepted (poly-length) certificate and can be verified in poly-time

$P \leq NP$ for $P=NP$, need $NP \leq P$

no one has proved that $NP \subseteq P$ or that $NP \not\in P$
Problem B is NP-complete if it is as hard as any other NP problem and is in NP.

⇒ any other problem in NP can be reduced to it

\[ A \leq_p B \text{ if we can write alg for } A; \]

\[ \text{poly-time} \rightarrow 1) \text{ take input, convert input to format for } B \\
2) \text{ run alg for } B \text{ on result of 1) } \\
3) \text{ output result of 2) } \]

if B is easy then A is easy

if A is hard then B is hard
HAM-CYCLE: given $G$, does it have a cycle visiting each vertex exactly once (Hamiltonian cycle)

HAM-PATH: given $G$, does it have a path visiting each vertex exactly once

HAM-CYCLE is NP-complete

To show that HAM-PATH is NP-complete: show $H-C \leq_p H-P$

(since $\leq_p$ is transitive and $A \leq_p H-C$ for all $A \in NP$

then $A \leq_p H-P$ for all $A \in NP$

We want to construct $G'$ from $G$ s.t. $G'$ has $H-P$

Alg for $H-C$

1) Construct $G'$
2) Run $H-P$ on $G'$
3) Output result of $2$
1) add vertex $s$, edge $(s, u)$
   (choose $u$ arbitrarily)

2) add vertex $t$, edge $(t, x)$
   for all $x$ adj to $u$

   now messed up $\Rightarrow$

To make $G'$, pick any $u \in G$

add new vertex $u'$ s.t., $u \in G \Rightarrow (u', v) \in G'$

add $(s, u)$ $(u', t)$
$G$ has $H-C \iff G'$ has $H-P$

$\Rightarrow$: Suppose $G$ has $H-C$. Call it $u, v_1, v_2, v_3, \ldots, v_k, u.$

$G'$ has $H-P$: $s, u, v, \ldots, v_k, u', \ell.$

$\Leftarrow$: Suppose $G'$ has $H-P$. It must start/end at $s, \ell$.

$H-P$ is $s, u, v_1, \ldots, v_k, u', \ell$

So $G$ has $H-C: u, v_0, \ldots, v_k, u$.

3-CNF-SAT: Given $\phi$ in 3-CNF form, determine if satisfiable

$$\varphi \iff (\bar{x} \lor y \lor z) \land (x \lor \bar{y} \lor \bar{w})$$

3-CNF-SAT $\in$ NP (evidence is the assignment - plug in values to evaluate $\phi$ in poly time)

3-CNF-SAT is NP-complete (reduce from SAT)
Given $G, k$, does $G$ have $k$ vertices connected to each other

CLIQUE is NP-complete
1) CLIQUE $\in$ NP (above)

2) $3$-CNF-SAT $\leq_p$ CLIQUE

- Need Alg for $3$-CNF-SAT($\varphi$)

  1) Construct $G$, pick $k$ from $\varphi$

  2) run CLIQUE on input $(G, k)$

  3) output result of 2)

$\varphi$ is satisfiable

$(\overline{x}y \lor \overline{z}) \land (x \lor y \lor \overline{w}) \land (w \lor x \lor \overline{z})$
3 verts for each clause representing literals in clause
Edges between verts in diff clauses if not contradictory
\( k = \# \text{ of clauses} \)
Suppose \( \Phi \) is satisfiable. Then 2 one literal in each clause of \( \Phi \).

Pick one \( T \) literal per clause — that's a 3-clique.

Suppose \( G \) has a 3-clique. Must be 1 vert from each group of 3. These are the satisfying assignment.