Can you ...

- show sorting is $\Omega(n \log n)$?
- define the class P?
- define the class NP?
- show 3SAT is NP-complete? (at least at a high level)
- show that TAUTOLOGY is in Co-NP?
- define a one-way function?
- understand every relationship in

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXP?$$
Can you do these reductions?

- $3\text{SAT} \leq_P \text{IND}$
- $\text{IND} \leq_P \text{CLIQUE}$
- $3\text{SAT} \leq_P 3$-Colourability
- $3\text{SAT} \leq_P \text{HAM}$
- $\text{HAM} \leq_P \text{TSP}$
- $3\text{SAT} \leq_P 3\text{DM}$
- $3\text{DM} \leq_P \text{XSC} \ (\text{Exact} \ \text{Set} \ \text{Cover})$
- $\text{XSC} \leq_P \text{SC} \ (\text{Set} \ \text{Cover})$
- $\text{XSC} \leq_P \text{KNAPSACK}$

(Undirected) Hamiltonian Path problem (HAM-PATH)

**HAM-PATH**

Given a graph $G = (V, E)$, does it contain a path that visits every node exactly once?

HAM-PATH is NP-complete.

Proof (Papadimitriou, pages 193 to 198): The problem is in NP since we can guess a path and check it in polynomial time. To show that it is NP-complete we do a reduction from 3SAT. First, we need a gadget to represent each variable $x$:

We construct a chain of these gadgets, one for each variable.
The exclusive or (XOR) gadget

Representing clauses

For each clause of three literals \((l_1 \lor l_2 \lor l_3)\), we construct a (virtual) triangle where each “edge” is associated with a literal, and connected with an XOR gadget to the virtual link that of the literal’s variable that would make the literal true.
Example

\[(x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3)\]

The initial end-points are connected to a new node with label 1, the terminal end-points are connected to a dotted node (All dotted nodes will be connected in one large clique.) Finally, we add one arc with a dotted node and a node labeled 3. Thus, any Hamiltonian path must link nodes 1 and 3.

Eulerian Path Problem is in P

**Eulerian Path Problem**

Given a graph \( G = (V, E) \), does it contain a path that visits every edge exactly once?

1. Pick any vertex to start.
2. From the current vertex, pick any edge, but never cross a bridge in the reduced graph (the graph with marked edges deleted), unless there is no other choice. A bridge is an edge whose deletion would increase the number of connected components.
3. Mark the edge (so will not use it again).
4. Traverse the edge, picking the node at the other end.
5. Repeat steps 2 through 4, until back at the starting point (or failure).

**Exercise**

Prove that this algorithm is correct.
Directed Hamiltonian Path problem (DHAM-PATH)

**HAM-PATH**
Given a directed graph $G = (V, E)$, does it contain a path that visits every node exactly once?

**DHAM-PATH is NP-complete.**
The problem is in NP since we can guess a path and check it in polynomial time.
To show that it is NP-complete we do a reduction from 3SAT.

We construct a graph that looks like this ...

![Graph Diagram]

[Diagram showing a directed graph with nodes labeled $s$, $x_1$, $x_2$, ..., $t$, and additional nodes $c_1$, $c_2$, ..., $c_k$.]
Each variable diamond has a chain ...

... chains are connected to clause nodes

xi appears on Cj

\neg xi appears on cj