1. The complexity class $\text{coNP}$ consists of languages whose complement is in $\text{NP}$. It is currently an open question if $\text{coNP} = \text{NP}$. Show if $\text{NP} \neq \text{coNP}$ then $\text{P} \neq \text{NP}$.

2. Show that if $\text{P} = \text{NP}$ then any language $A \in \text{P}$ where $A \neq \emptyset$ and $A \neq \Sigma^*$ is $\text{NP}$-complete.

3. Give a reduction from $\text{3-COLOR}$ to $\text{3-SAT}$. If we know $\text{3-SAT}$ is $\text{NP}$-complete, what do we know about $\text{3-COLOR}$ based on this reduction?

4. (L&P 6.3.3) Consider a Boolean formula in 2-CNF. Any clause $(x \lor y)$ can be thought of as two implications $\overline{x} \implies y$ and $\overline{y} \implies x$. The clause $(x)$ can be thought of as $\overline{x} \implies x$. If we then consider $x \implies y$ as a directed edge from a vertex $x$ to a vertex $y$, we can construct an implication graph from any 2-CNF formula. Show that a 2-CNF formula is unsatisfiable if and only if there is a variable $x$ such that there is a path in the constructed graph from $x$ to $\overline{x}$ and a path from $\overline{x}$ to $x$. Design an algorithm based on this observation to show $\text{2-SAT} \in \text{P}$.