CS46 practice problems 12

These practice problems are an opportunity for discussion and trying many different solutions. It is **not counted towards your grade**, and **you do not have to submit your solutions.** The purpose of these problems is to get more comfortable with reasoning and writing about P, NP, and polynomial-time reductions.

If you are stumped or looking for guidance, **ask**.

1. Show that if coNP $\neq$ NP then P $\neq$ NP.

2. A **vertex cover** in a graph $G = (V, E)$ is a subset $S \subset V$of vertices where every edge of G has at least one endpoint in the subset.

   $$\textsc{VertexCover} = \{\langle G, k\rangle \mid G \text{ has a } k\text{-node vertex cover}\}$$

   An **independent set** in a graph G is a subset of vertices with no edges between them.

   $$\textsc{IndependentSet} = \{\langle G, k\rangle \mid G \text{ contains an independent set of } k \text{ vertices}\}$$

   Show that $\textsc{IndependentSet} \leq_p \textsc{VertexCover}$.

3. One way to come up with *new* NP-complete problems is to generalize from a problem we already know is NP-complete. Then, if certain parameters of the problem are fixed in a certain way, the problem becomes a known NP-complete problem. One can reduce any problem to its generalization by simply introducing a new parameter, and otherwise leaving the instance as it is.

   Prove that the following language is NP-complete by showing that it is the generalization of an NP-complete problem. Give the appropriate parameter restriction.

   LongestCycle : Given a graph G and integer k, is there a cycle, with no repeated nodes, of length at least k?

4. Show that if P = NP, a polynomial-time algorithm exists that produces a satisfying assignment when given a satisfiable Boolean formula.

   Note: The algorithm you are being asked to write computes a function, but NP contains languages, not functions. The P = NP assumption means that Satisfiability $\in$ P, so there is a deterministic polynomial-time Turing machine M_{SAT} which can test if a formula is satisfiable. You don't know how this test is done, but you may use M_{SAT} in your algorithm.