Proving that a problem X is NP-hard requires several steps:

- Choose a problem Y that you already know is NP-hard.
- Describe an algorithm to solve Y, using an algorithm for X as a subroutine. Typically this algorithm has the following form: Given an instance of Y, transform it into an instance of X, and then call the magic black-box algorithm for X.
- Prove that your algorithm is correct. This almost always requires two separate steps:
  - Prove that your algorithm transforms "yes" instances of Y into "yes" instances of X.
  - Prove that your algorithm transforms "no" instances of Y into "no" instances of X. Equivalently: Prove that if your transformation produces a "yes" instance of X, then it was given a "yes" instance of Y.
- Argue that your algorithm for Y runs in polynomial time.

Proving that X is NP-Complete requires you to additionally prove that $X \in NP$ by describing a non-deterministic polynomial-time algorithm for X. Typically this is not hard for the problems we consider but it is not always obvious.

Recall that a *Hamiltonian cycle* in a graph G is a cycle that goes through every vertex of G exactly once. The problems of determining whether or not there is a directed Hamiltonian cycle in a directed graph, or an undirected Hamiltonian cycle in an undirected graph, are both hard. We investigate reductions involving Hamiltonian cycle problems.

1. Recall that in lecture we saw a reduction for the *Hamiltonian cycle* problem in directed graphs to the same problem in undirected graphs. In particular, if G is a directed graph, then the undirected graph G' is formed as follows: G' contains all vertices of G, but in addition, for each vertex v in G, two new vertices are added to G': v_{in} and v_{out}. Edges of G' include:

   - For each vertex v, undirected edges (v_{in}, v) and (v, v_{out}), are included in G'.
   - For each directed edge (u, v) of G, the undirected edge (u_{out}, v_{in}) is included in G'.

   First draw a simple directed graph G with vertices u, v, w and directed edges $(u, v), (u, w), (v, w)$. Now create G' and check it against a different group's answer to make sure you understand the reduction.

   Now, prove the correctness of the reduction.

2. A ***tonian cycle*** in an undirected graph G is a cycle that goes through at least *half* of the vertices of G, and a ***Hamilhamiltonian circuit*** in an undirected graph G is a closed walk that goes through every vertex in G exactly *twice*.

   (a) Prove that it is NP-hard to determine whether a given graph contains a tonian cycle. (This reduction should be easy.)

   (b) (harder) Prove that it is NP-hard to determine whether a given graph contains a Hamilhamiltonian circuit. *Hint: hang a small "gadget" off of each vertex*