Homework 1: P vs. NP; definitions

Instructions: Five classes of problems are listed below. For each / for as many as you want, attempt to do the following:

- Find an algorithm that solves the problem. Check the runtime of your algorithm. (It will likely be huge.)
- Show that your problem is in NP: i.e. find an algorithm that will take in an instance of your problem and a "proof" that claims to show that instance is true, and check in polynomial time whether that solution holds.
- Show that if you can solve this problem "quickly," you can solve 3SAT "quickly." In other words, find a way to transform any algorithm A that solves this problem into an algorithm that solves 3SAT, such that if the runtime of A is $t(n)$, the runtime of this new 3SAT solver is $\text{poly}(t(n))$.
- Then, try to improve your algorithm in such a way that your problem is in P. (This step may be difficult.)

Homework Problems

1. Given an arbitrary $n \times n$ partial latin square P, does it have a completion to an $n \times n$ latin square L ,in which all of its rows and columns are filled?
2. In a graph $G = (V, E)$, a **Hamiltonian cycle** is a sequence of vertices and edges $(v_1, e_{12}, v_2, e_{23}, \ldots v_n, e_{n1}$, such that
   - each vertex in V shows up in our sequence exactly once, and
   - the edges e_{ij} are all edges linking vertex v_i to vertex v_j.

   In other words, a Hamiltonian cycle is a tour that starts and stops at the same vertex, and along the way visits every other vertex exactly once.

   Given an arbitrary graph G on n vertices, does it have a Hamiltonian cycle?
3. A 3-**coloring** of a graph G is a way to assign the colors $\{1, 2, 3\}$ to the vertices of a graph in such a way that no edge has both of its endpoints colored the same color.

   Given a graph G, does it have a 3-coloring?
4. Take a graph G. We can play a solitaire game, called **pebbling**, on this graph. We define this as follows:
   - Setup: a graph G. Also, to every vertex of G, we assign some number of "pebbles," which we imagine are stacked on top of each vertex.

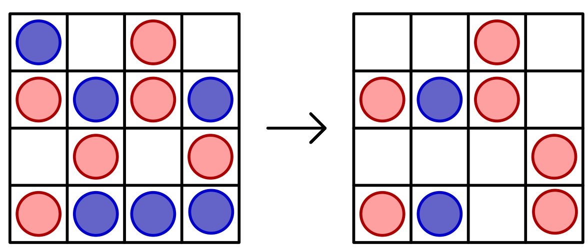
- Moves: Suppose we have an edge e_{12} connecting v_1 to v_2, and another edge e_{23} connecting v_2 and v_3. Suppose further that there is a pebble on v_1 and v_2. We can then "jump" the pebble v_1 over the pebble at v_2 to v_3: i.e. we can remove one pebble from each of v_1 and v_2, and place a pebble on v_3.
- A game is **cleared** if we can reduce it to having only one pebble on the entirety of the board.

Given an arbitrary graph G on n vertices, and some arrangement of n pebbles on G, can this game ever be "cleared"?

5. Consider the following solitare game, which is played on a $n \times n$ board:

   - To start, we place red stones on some of the squares of our board, and blue stones on other squares of our board. We do not have to fill every square of our board; some places may be left blank.
   - The goal of our game is to remove stones one by one until we satisfy both of the following conditions:
     - Every **row** contains at least one stone.
     - Conversely, no **column** contains stones of both colors in it.

   Here is an example board, along with a winning state:



   For some initial configurations, this game is impossible to win (find one!)

   The problem is the folowing: given a starting board state, is is possible to win?