This is the last graded homework before the final exam.

1. (a) A subset $S$ of vertices in an undirected graph $G$ is **half-independent** if each vertex in $S$ is adjacent to at most one other vertex in $S$. Prove that finding the size of the largest half-independent set of vertices in a given undirected graph is NP-hard.

(b) A subset $S$ of vertices in an undirected graph $G$ is **sort-of-independent** if if each vertex in $S$ is adjacent to at most 374 other vertices in $S$. Prove that finding the size of the largest sort-of-independent set of vertices in a given undirected graph is NP-hard.

2. Fix an alphabet $\Sigma = \{0, 1\}$. Prove that the following problems are NP-hard.\(^1\)

   (a) Given a regular expression $R$ over the alphabet $\Sigma$, is $L(R) \neq \Sigma^*$?

   (b) Given an NFA $M$ over the alphabet $\Sigma$, is $L(M) \neq \Sigma^*$?

   [Hint: Encode all the **bad** choices for some problem into a regular expression $R$, so that if all choices are bad, then $L(R) = \Sigma^*$.]

3. Let $\langle M \rangle$ denote the encoding of a Turing machine $M$ (or if you prefer, the Python source code for the executable code $M$). Recall that $x \cdot y$ denotes the concatenation of strings $x$ and $y$. Prove that the following language is undecidable.

   $SELF\overline{SELF ACCEPT} := \{ \langle M \rangle \mid M \text{ accepts the string } \langle M \rangle \cdot \langle M \rangle \}$

   Note that Rice’s theorem does **not** apply to this language.

---

\(^1\)In fact, both of these problems are NP-hard even when $|\Sigma| = 1$, but proving that is much more difficult.
Solved Problem

4. A double-Hamiltonian tour in an undirected graph $G$ is a closed walk that visits every vertex in $G$ exactly twice. Prove that it is NP-hard to decide whether a given graph $G$ has a double-Hamiltonian tour.

This graph contains the double-Hamiltonian tour $a \rightarrow b \rightarrow d \rightarrow g \rightarrow e \rightarrow b \rightarrow d \rightarrow c \rightarrow f \rightarrow a \rightarrow c \rightarrow f \rightarrow g \rightarrow e \rightarrow a$.

**Solution:** We prove the problem is NP-hard with a reduction from the standard Hamiltonian cycle problem. Let $G$ be an arbitrary undirected graph. We construct a new graph $H$ by attaching a small gadget to every vertex of $G$. Specifically, for each vertex $v$, we add two vertices $v^\uparrow$ and $v^\downarrow$, along with three edges $vv^\downarrow$, $vv^\uparrow$, and $v^\downarrow v^\uparrow$.

![A vertex in $G$, and the corresponding vertex gadget in $H$.](image)

I claim that $G$ has a Hamiltonian cycle if and only if $H$ has a double-Hamiltonian tour.

$\Rightarrow$ Suppose $G$ has a Hamiltonian cycle $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$. We can construct a double-Hamiltonian tour of $H$ by replacing each vertex $v_i$ with the following walk:

$$\cdots \rightarrow v_i \rightarrow v_i^\downarrow \rightarrow v_i^\uparrow \rightarrow v_i^\uparrow \rightarrow v_i^\downarrow \rightarrow v_i \rightarrow \cdots$$

$\Leftarrow$ Conversely, suppose $H$ has a double-Hamiltonian tour $D$. Consider any vertex $v$ in the original graph $G$; the tour $D$ must visit $v$ exactly twice. Those two visits split $D$ into two closed walks, each of which visits $v$ exactly once. Any walk from $v^\downarrow$ or $v^\uparrow$ to any other vertex in $H$ must pass through $v$. Thus, one of the two closed walks visits only the vertices $v$, $v^\downarrow$, and $v^\uparrow$. Thus, if we simply remove the vertices in $H \setminus G$ from $D$, we obtain a closed walk in $G$ that visits every vertex in $G$ once.

Given any graph $G$, we can clearly construct the corresponding graph $H$ in polynomial time.

With more effort, we can construct a graph $H$ that contains a double-Hamiltonian tour that traverses each edge of $H$ at most once if and only if $G$ contains a Hamiltonian cycle. For each vertex $v$ in $G$ we attach a more complex gadget containing five vertices and eleven edges, as shown on the next page.
Non-solution (self-loops): We attempt to prove the problem is NP-hard with a reduction from the Hamiltonian cycle problem. Let $G$ be an arbitrary undirected graph. We construct a new graph $H$ by attaching a self-loop every vertex of $G$. Given any graph $G$, we can clearly construct the corresponding graph $H$ in polynomial time.

Suppose $G$ has a Hamiltonian cycle $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_n \rightarrow v_1$. We can construct a double-Hamiltonian tour of $H$ by alternating between edges of the Hamiltonian cycle and self-loops:

$v_1 \rightarrow v_1 \rightarrow v_2 \rightarrow v_2 \rightarrow v_3 \rightarrow \cdots \rightarrow v_n \rightarrow v_n \rightarrow v_1$.

On the other hand, if $H$ has a double-Hamiltonian tour, we cannot conclude that $G$ has a Hamiltonian cycle, because we cannot guarantee that a double-Hamiltonian tour in $H$ uses any self-loops. The graph $G$ shown below is a counterexample; it has a double-Hamiltonian tour (even before adding self-loops!) but no Hamiltonian cycle.

Rubric: 10 points, standard polynomial-time reduction rubric