Standard 10-point rubric for all NP-hardness proofs.

- 3 points for a correct reduction. Remember that a reduction is an ***algorithm***; it must be described clearly and precisely (but not necessarily using pseudocode).
- 6 points for a proof of correctness. Every proof of correctness has two parts ("one for each F"); each part is worth 3 points.
- 1 point for time analysis. Writing "The reduction obviously runs in polynomial time." is usually sufficient.

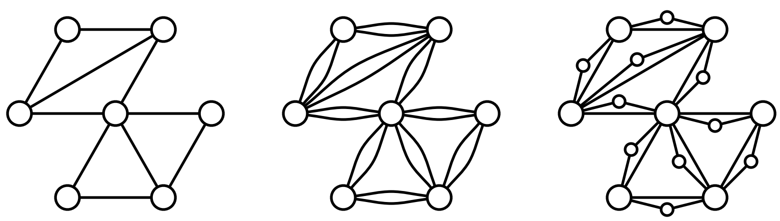
For example, an polynomial-time reduction that sometimes gives false positives but never false negatives is worth at most 4 points.

1. (a) Describe a polynomial-time reduction from EvenGraphIsomorphism to GraphIsomorphism.

Solution: The reduction is trivial: Given two graphs G and H in which every vertex has even degree, we can determine whether they are isomorphic simply by calling GraphIsomorphism(G, H). ■

(b) Describe a polynomial-time reduction from GraphIsomorphism to EvenGraphIsomorphism.

Solution: Suppose we are asked whether two graphs G and H are isomorphic. We can transform the input graphs into *even* graphs G' and H' in polynomial time by replacing each edge with a pair of parallel edges with the same endpoints. (If parallel edges make you uncomfortable, we can replace each edge in the input graphs with a triangle, as shown on the right in the figure below.) The new even graphs G' and H' are isomorphic if and only if the original graphs G and H are isomorphic. Thus, we can determine whether G and H are isomorphic by calling EvenGraphIsomorphism(G', H'). ■



Transforming an arbitrary graph (left) into an even graph (middle or right).

(c) Describe a polynomial-time reduction from GraphIsomorphism to SubgraphIsomorphism.

Solution: Suppose we are asked whether two graphs G and H are isomorphic. We first check if the graphs have the same number of edges; if not, we immediately return False. If the graphs do have the same number of edges, then G is isomorphic to H if and only if G is isomorphic to a subgraph of H. (Recall that every graph is a subgraph of itself, just as every set is a subset of itself.) Thus, we can determine whether G and H are isomorphic by calling SubgraphIsomorphism(G, H). The reduction clearly takes polynomial time. ■

(d) Prove that SubgraphIsomorphism is NP-complete.

Solution: We can easily verify in polynomial time that one graph is a subgraph of another if we are given the correspondence between vertices, so SubgraphIsomorphism is in NP.

We prove the problem is NP-hard by a reduction from the Hamiltonian cycle problem in in undirected graphs. (See problem 2.) To determine whether an arbitrary graph G with n vertices contains a Hamiltonian cycle, we construct a cycle C_n of n vertices and call SUBGRAPHISOMORPHISM(C_n, G). The reduction clearly takes polynomial time. Thus, SUBGRAPHISOMORPHISM is NP-hard.

We can also reduce from MAXCLIQUE. Let K_r denote the complete graph with r vertices.

```
MaxClique(G)
  for r ← 1 to ∞
      if SubgraphIsomorphism(K_r, G) = False
          return r − 1
```

The loop must stop before $r = n + 1$, so this reduction runs in polynomial time.

In fact, the reduction in the lecture notes implies that the following decision problem is NP-hard: Given a graph G with n vertices, does it contain a clique with $n/3$ vertices? We can solve this problem by calling SUBGRAPHISOMORPHISM$(K_{n/3}, G)$. ■

(e) What can you conclude about the NP-hardness of GRAPHISOMORPHISM? Justify your answer.

Solution: ***Absolutely nothing!*** The reduction in part (c) is in the wrong direction to imply that GRAPHISOMORPHISM is NP-hard. ■

Rubric: ½ point for part (a); ½ point for part (e); 1 point for each of the other parts.

2. Prove that the following problems are NP-hard.

   (a) Given an undirected graph G, does G have a spanning tree with at most 473 leaves?

Solution: We prove the problem is NP-hard by reduction from the undirected Hamiltonian path problem. Given an arbitrary undirected graph G, we construct a graph H by adding 473 new vertices $x, y_1, y_2, \ldots, y_{472}$ to G, with edges connecting x to each new vertex y_i and to every vertex of G.

Suppose G has a Hamiltonian path through vertices $v_1, v_2, \ldots, v_n$. Adding the edges $v_n x$ and $x y_i$ for all i to this path creates a spanning tree of H with 473 leaves, namely $v_1, y_1, y_2, \ldots, y_{472}$.

On the other hand, suppose H has a spanning tree T with at most 473 leaves. Every spanning tree of H must contain the edges $x y_i$ for all i, and each vertex y_i must be a leaf. Thus, at most one node in G is a leaf in T, and therefore *exactly* one in G is a leaf in T. It follows that removing vertices $x, y_1, \ldots y_{472}$ from T leaves a spanning tree of G with exactly two leaves, or in other words, a Hamiltonian path in G.

We conclude that G has a Hamiltonian path if and only if H has a spanning tree with at most 473 leaves. Given G, we can easily construct H in polynomial time. ■

   (b) Given an undirected graph $G = (V, E)$, what is the size of the largest subset of vertices $S \subseteq V$ such that at most 2015 edges in E have both endpoints in S?

Solution: We prove the problem is NP-hard by reduction from the maximum independent set problem. Given an arbitrary undirected graph G, we construct a graph H by adding 4030 new vertices $x_1, x_2, \ldots, x_{2015}, y_1, y_2, \ldots, y_{2015}$ and edges $x_i y_i$ for each index i; these new vertices are not connected to the original vertices in G.

Suppose G has an independent set S of size k. Then the set $S \cup \{x_1, x_2, \ldots, x_{2015}, y_1, y_2, \ldots, y_{2015}\}$ is a subset of $k + 4030$ vertices of H that induces exactly 2015 edges.

On the other hand, suppose there is a subset of ℓ vertices of H that induces at most 2015 edges in H. Let S be such a subset that indices as few edges of the original graph G as possible. There are two cases to consider:

- If S contains all vertices x_i and y_i, then $S \setminus \{x_1, x_2, \ldots, x_{2015}, y_1, y_2, \ldots, y_{2015}\}$ is an independent set of size $\ell - 4030$ in G.
- Suppose S induces an edge uv in G. Then S must exclude at least one of the new vertices, say x_i. Let S' be the set obtained from S by removing u and inserting x_i. This new set S' induces at most 2015 edges in H, and induces fewer edges in G, which contradicts our choice of S.

We conclude that for any integer k, the largest independent set in G has size k if and only if the largest 2015-edge set in H has size $k + 4030$. Given G, we can easily construct H in polynomial time. ■

3. (a) Describe a polynomial-time reduction from UndirectedHamCycle to DirectedHamCycle.

Solution: Given an arbitrary undirected graph G, we construct a directed graph H by replacing each undirected edge uv with two directed edges $u \to v$ and $v \to u$.

Any cycle in G is also a cycle in H, and vice versa. In particular, any Hamiltonian cycle in G is also a Hamiltonian cycle in H, and vice versa.

Given G, we can clearly construct H in polynomial time. (In fact, this construction takes ***no time at all***. If G is represented using an adjacency list, then H is represented by the *exactly the same* adjacency list. Similarly, if G is represented using an adjacency matrix, then *exactly the same* adjacency matrix represents H.) ■

(b) Describe a polynomial-time reduction from DirectedHamCycle to UndirectedHamCycle.

Solution: Given an arbitrary directed graph G, we construct a directed graph H as follows.

- For each vertex v_i in G, the graph H contains three vertices x_i, y_i, z_i.
- For each vertex v_i in G, the graph H contains edges $x_i y_i$ and $y_i z_i$.
- For each directed edge $v_i \to v_j$ in G, the graph H contains the edge $z_i x_j$.

Given G, we can easily construct H in polynomial time.

Suppose G contains the Hamiltonian cycle C. Reindex the vertices of G so that $C = v_0 \to v_1 \to \cdots \to v_{n-1} \to v_0$. Then we can construct a Hamiltonian cycle in H by replacing each edge $v_i \to v_{i+1 \bmod n}$ in C with the corresponding path $x_i \to y_i \to z_i \to x_{i+1 \bmod n}$.

On the other hand, suppose H contains a Hamiltonian cycle C. Because each vertex y_i has degree 2, the vertices x_i, y_i, z_i must appear together in C, for every index i. Moreover, if we direct the edges of C so that C contains the directed path $x_i \to y_i \to z_i$ for *some* index i, then C must contain the directed path $x_i \to y_i \to z_i$ for *every* index i. Thus, we can reindex the vertices so that

$$C = x_0 \to y_0 \to z_0 \to x_1 \to y_1 \to z_1 \to x_2 \cdots \to x_{n-1} \to y_{n-1} \to z_{n-1} \to x_0.$$

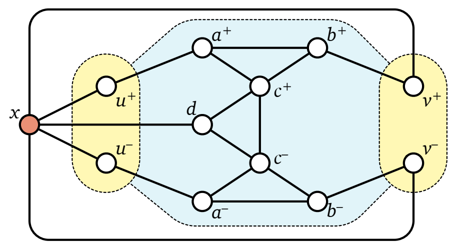
Each directed edge $z_i \to x_{i+1 \bmod n}$ in C corresponds to a directed edge $v_i \to v_{i+1 \bmod n}$ in G. It follows that the corresponding cycle $v_0 \to v_1 \to \cdots \to v_{n-1} \to v_0$ is a Hamiltonian cycle in G.

We conclude that G contains a Hamiltonian cycle if and only if H contains a Hamiltonian cycle. ■

4. ***[Extra credit]*** Describe a direct polynomial-time reduction from 4COLOR to 3COLOR.

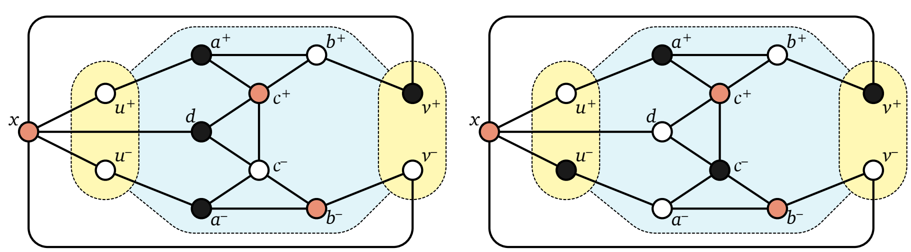
Solution: Given an arbitrary graph G, we construct a new graph H as follows.

- H contains a single *ground vertex* x. In any valid 3-coloring of H, the neighbors of x must use only the two colors different from x; call those two colors 0 and 1.
- For each vertex v in G, the graph H contains a vertex gadget, which consists of two vertices v^+ and v^- connected by edges to the ground vertex x, but not to each other. In any valid 3-coloring of H, the vertices v^+ and v^- are colored either 0 or 1. Thus, the vertex gadget has four different valid states, which intuitively correspond to possible colors of the original vertex v in G.
- For each edge uv in G, the graph H contains an *edge gadget*, which consists of seven vertices $a^+, b^+, c^+, a^-, b^-, c^-, d$ connected to the vertex gadgets of u and v and to the ground vertex x as shown below.



I claim that in any valid 3-coloring of H, any two vertex gadgets that are connected by an edge gadget must have different states. Straightforward case analysis implies that if nodes u^+ and v^+ have the same color, then node c^+ also has that color. On the other hand, vertex d must have color 0 or 1, so either c^+ or c^- must have color x. Thus, either v^+ and u^+ have different colors, or v^- and u^- have different colors, which proves the claim. It follows that, given any valid 3-coloring of H, we can construct a valid 4-coloring of G by using the state of each vertex gadget in H as the color of the corresponding vertex in G.

Now suppose G has a valid 4-coloring, with colors $0, 1, 2, 3$. For each vertex u in G, assign colors 0 or 1 to the corresponding vertices in H so that $color(u) = 2 \cdot color(u^+) + color(u^-)$. Straightforward case analysis implies that we can extend this color assignment to a valid 3-coloring of each edge gadget. There are only two essentially different cases to consider: Either three of the four vertices u^+, u^-, v^+, v^- have the same color, or two have color 0 and two have color 1.



We conclude that G is 4-colorable if and only if H is 3-colorable. Given G, we can easily construct H in polynomial time. ■