

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

- Choose a problem Y that you already know is NP-hard (because we told you so in class).
- 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 always requires two separate steps, which are usually of the following form:
 - **Prove** that your algorithm transforms “good” instances of Y into “good” instances of X .
 - **Prove** that your algorithm transforms “bad” instances of Y into “bad” instances of X . Equivalently: Prove that if your transformation produces a “good” instance of X , then it was given a “good” instance of Y .
- Argue that your algorithm for Y runs in polynomial time. (This is usually trivial.)

1. Recall the following k COLOR problem: Given an undirected graph G , can its vertices be colored with k colors, so that every edge touches vertices with two different colors?

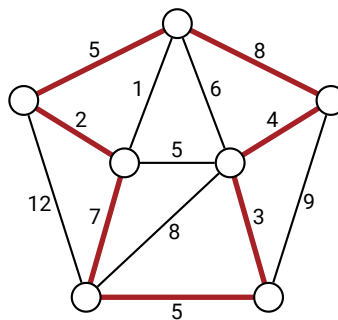
- (a) Describe a direct polynomial-time reduction from 3COLOR to 4COLOR.
- (b) Prove that k COLOR problem is NP-hard for any $k \geq 3$.

2. A *Hamiltonian cycle* in a graph G is a cycle that goes through every vertex of G exactly once. Deciding whether an arbitrary graph contains a Hamiltonian cycle is NP-hard.

A *tonian cycle* in a graph G is a cycle that goes through at least *half* of the vertices of G . Prove that deciding whether a graph contains a tonian cycle is NP-hard.

To think about later:

3. Let G be an undirected graph with weighted edges. A Hamiltonian cycle in G is **heavy** if the total weight of edges in the cycle is at least half of the total weight of all edges in G . Prove that deciding whether a graph contains a heavy Hamiltonian cycle is NP-hard.



A heavy Hamiltonian cycle. The cycle has total weight 34; the graph has total weight 67.