Bill Davis
605.421

1. 34.2.1 GRAPH-ISOMORPHISM
   We can show that GRAPH-ISOMORPHISM $\in$ NP by demonstrating a polynomial-time algorithm to verify. So assume that f is a function from G_1 to G_2. We can verify f is an isomorphism if the graphs can be represented as an adjacency matrix by

   ```
   Function-Verify(G1, G2, f)
   for x ∈ G1
       for y ∈ G1
           if G1(x, y) ≠ G2(f(x), f(y))
               return false
   return true
   ```

   This algorithm works in $\mathbf{O}(V^2)$ time. A similar algorithm is possible for an adjacency matrix, but would require two passes, once through each list, to verify that every edge in G_1 has an equivalent edge in G_2 and vice versa.

2. 34.2-8 TAUTOLOGY
   We can show that TAUTOLOGY $\in$ co-NP by demonstrating that for a given boolean formula, there exists an arrangement of boolean values for which the boolean formula is false. This is pretty easy to verify, all we have to do given a set of input values is perform the boolean operation using them as input and observe the output. If it's 0 then the formula is not a tautology. This can happen in $\mathbf{O}$(n) time if n is the number of input values.

3. 34-3 Graph Coloring

   (a) You can determine if a 2-coloring is sufficient by performing a breadth first search on each connected component of a graph labeling each depth a different color. If during the DFS a node a level contains a previously colored node of a different color then a 2 coloring is insufficient.

   (b) A reformulation of the graph coloring problem as a decision problem could be; Can the graph G be colored using at most k colors?

Assume we can solve graph coloring in polynomial time, then obviously we can solve the decision problem in polynomial time. All we would have to do is solve the graph-coloring problem with solution n colors. Then for any formulation of the decision problem with k<=n, the answer would be yes, and for any formulation with k>n the answer is no.

Assume we can solve the decision problem in polynomial time. For a graph with |V|nodes, we can solve the decision problem for k=1 to |V|times in polynomial time, and the first time we get a no from the decision problem we have solved the graph coloring problem.

(c) Since we can construct a polynomial time transformation from the decision problem to the 3-COLOR problem by simply solving the decision problem at most |V|times, we know that if 3-COLOR is NP-Complete then so is the decision problem

(d) Since we know that x and $\bar{x}$ are connected, $c(x) \neq c(\bar{x})$. Also, since both x and $\bar{x}$ are connected to RED, and TRUE and FALSE are connected to RED either $c(x)$ equals c(TRUE) or c(FALSE), and $c(\bar{x})$ equals the other.

From above, if our three colors are r,g,b we can have c(x) = c(TRUE) = g , c($\bar{x}$) = c(FALSE) = b, and c(RED) = r. These colors any three graph containing the literal edges.

(e) Assume, by way of contradiction, that x,y,z are all colored c(FALSE) and the graph is 3-colorable. As a result neither node attached to TRUE can be colored c(FALSE). Also, since they are both attached to TRUE, neither can be colored c(TRUE). Therefore they must both have the same color, since there are only 3 colors available, and this contradicts that the graph was 3-Colored.

(f) We can use the above items to construct a graph out of the widget for a given boolean statement , using 1 widget for each clause. As a result 3-COLOR has a solution if and only if 3-CNF-SAT has a solution.

We proved in e that the graph is 3-colorable if one of x,y,z is colored c(TRUE). Therefore the clause is satisfiable.

Also if the clause is only satisfiable iff one of x,y,z is labeled true, which means that the graph is also 3-colorable.