29. (2 points)

    (a) Show that the Vertex Cover Problem is self-reducible. The decision problem is to take a graph G and an integer k and decide if G has a vertex cover of size k or not. The optimization problem takes a graph G, and returns a smallest vertex cover in G. So you must show that if the decision problem has a polynomial time algorithm then the optimization problem also has a polynomial time algorithm. Recall that a vertex cover is a collection S of vertices with the property that every edge is incident to a vertex in S.

    (b) In the dominating set problem the input is an undirected graph G, the problem is to find the smallest dominating set in G. A dominating set is a collection S of vertices with the property that every vertex v in G is either in S, or there is an edge between a vertex in S and v. Show that the dominating set problem is NP-hard using the fact that the vertex cover problem is NP-complete. Hint: Local replacement will work.

30. (4 points)

    (a) Show that the 3SAT problem is self-reducible. The input to the decision problem and the optimization problem is the same, namely a Boolean formula F in conjunctive normal form with three literals per clause that derive from unique variables. So the same variable can not appear more than once in any clause in F, either negated or unnegated. The decision problem is to determine if F is satisfiable. The optimization problem is to determine an arbitrary satisfying assignment for F if one exists, and a statement that F is unsatisfiable otherwise.

    (b) Show the decision version of 3SAT is NP-hard using the fact that 3-coloring is NP-complete. The 3-coloring problem is to determine whether a undirected graph G has a vertex coloring with 3 colors. A vertex coloring assigns a color to each vertex so that no pairs of adjacent vertices are colored the same color.

31. (6 points) For each of the following problems, either prove that it is NP-hard by reduction (from either the standard clique problem, or from the standard independent set problem, or from one of the previous subproblems), or give a polynomial time algorithm. Some of the reductions will be trivial, and its ok to dispose of these problems with a sentence or two explaining why the reduction is easy.

    (a) The input is an undirected graph G. Let n be the number of vertices in G. The problem is to determine if G contains a clique of size $3n/4$. Recall that a clique is a collection of mutually adjacent vertices.

    (b) The input is an undirected graph G. Let n be the number of vertices in G. The problem is to determine if G contains an independent set of size $3n/4$. Recall that an independent set is a collection of mutually nonadjacent vertices.

    (c) The input is an undirected graph G and an integer k. The problem is to determine if G contains a clique of size k **AND** an independent set of size k.

    (d) The input is an undirected graph G and an integer k. The problem is to determine if G contains a clique of size k **OR** an independent set of size k.

    (e) The input is an undirected graph G. Let n be the number of vertices in G. The problem is to determine if G contains a clique of size $3n/4$ **AND** an independent set of size $3n/4$.

    (f) The input is an undirected graph G. Let n be the number of vertices in G. The problem is to determine if G contains a clique of size $3n/4$ **OR** an independent set of size $3n/4$.