# CS1510 Reduction Problems 7, 9, & 13

Rebecca Negley, Sean Myers

October 21, 2011

7. Show that if one of the following three problems has a polynomial time algorithm, then they all do:

   - The input is two undirected graphs $G$ and $H$. The problem is to determine if the graphs are isomorphic.
   - The input is two directed graphs $G$ and $H$. The problem is to determine if the graphs are isomorphic.
   - The input is two undirected graphs $G$ and $H$ and an integer $k$. The problem is to determine if the graphs are isomorphic and all the vertices in each graph have degree $k$.

   Intuitively, two graphs are isomorphic if one can name/label the vertices so that the graphs are identical. More formally, two undirected graphs $G$ and $H$ are isomorphic if there is a bijection $f$ from the vertices of $G$ to the vertices of $H$ such that $(v, w)$ is an edge in $G$ if and only if $(f(v), f(w))$ is an edge in $G$. More formally, two directed graphgs $G$ and $H$ are isomorphic if there is a bijection $f$ from the vertices of $G$ to the vertices of $h$ such that $(v, w)$ is a directed edge in $G$ if and only if $(f(v), f(w))$ is a directed edge in $G$. The degree of a vertex is the number of edges incident to that vertex.

   Solution: First, we will reduce the undirected graph isomorphism problem to the directed graph isomorphism problem. For the input graphs $G$ and $H$, replace every undirected edge with two directed edges going opposite directions. There are at mot $O(n^2)$ edges, so this takes $O(n^2)$ time. We can then input our modified graphs $G'$ and $H'$ to the directed graph isomorphism problem. Since, in an undirected graph, an edge can be taken in either direction, $G'$ and $H'$ are isomorphic to $G$ and $H$, respectively. If an isomorphism exists between $G'$ and $H'$, an isomorphism must therefore exist between $G$ and $H$. Simiarly, if no isomorphism exists between $G'$ and $H'$, no isomorphism can exist between $G$ and $H$. Hence, we can just return the output of the directed graph isomorphism problem as our output for the undirected graph isomorphism problem. Using this polynomial time reduction, we can see that if a polynomial time algorithm exists for the directed graph isomorphism problem, one exists for the undirected graph isomorphism problem.

   Now, we will reduce the directed graph isomorphism problem to the undirected graph isomorphism problem. For the input directed graphs $G$ and $H$, we will create two undirected graphs $G'$ and $H'$ as follows: For each vertex in $G$, create two new vertices in $G'$. One vertex will represent the edges directed into the original vertex and the other will represent edges directed out of the original vertex. We wiill then "copy" the edges over from the original graph so that they connect to the appropriate "outgoing" and "incoming" vertices. To prevent an obvious isomorphism with the inverse of the graph, we connect all of the "outgoing" vertices in a $n$-clique, $n$ being the total number of vertices in the original graph. These edges will not duplicate any of the original edges because no edge in the original graph could have had to outgoing vertices or no incoming vertex. We construct $H'$ from $H$ the same way. We create $2n$ vertices in our new graph and then copy over $O(n^2)$ edges from the original graph. We also create an $n$-clique, taking $O(n^2)$ time as well. Hence, our conversion is $O(n^2)$ (so polynomial). We then input $G'$ and $H'$ to the undirected graph isomorphism problem and return

its output as the output for the directed graph isomorphism problem. To see this will be correct: Suppose the directed graphs are isomorphic. Then, there exists $f$ such that $(v, w)$ is a directed edge in $G$ iff $(f(v), f(w))$ is a directed edge in $G'$. Let $v_1$ indicate the outgoing vertex, $v_2$ the incoming vertex in the undirected graph. Then, by construction there is an edge $(v_1, w_2)$ in $G'$ iff there is an edge $(f(v)_1, f(w)_2)$ in $H'$. Similarly, an edge $(v_1, w_2)$ existing in $G'$ iff an edge $(f(v)_1, f(w)_2)$ exists in $H'$ implies $(v, w)$ a directed edge in $G$ iff $(f(v), f(w)$ a directed a edge in $H$. The $n$-clique with the outgoing $(u_2)$ vertices will be in every $G'$ and $H'$, so the graphs will be isomorphic as long as $G$ and $H$ have the same number of edges and are isomorphic. Hence, if a polynomial time algorithm exists for the undirected graph isomorphism problem, one exists for the directed graph isomorphism problem.

Next, we reduce the undirected graph isomorphism problem to the undirected graph with degree $k$ isomorphism problem. For our input graphs $G$ and $H$, we construct new graphs $G'$ and $H'$ as follows: Find the maximum degree $j$ of vertices in $G$ and $H$. (A polynomial time way to do this is outlined in the next paragraph.) Then, for every vertex in $G$ create a $j$-clique in $G'$. For each edge in $G$, create two edges connecting one of the vertices the each of the cliques corresponding to the original vertices. We must use vertices that are only connected to the other points in a clique (so do not connect the same vertex to two different vertices outside of the clique). When we have copied over (and doubled) each edge, we put loops connecting each vertex that is only connected to its $j$-clique to itself ($O(n)$ time). We create $H'$ from $H$ the same way. Then, each vertex in $G'$ and $H'$ have degree $j + 2$. We input $G'$, $H'$, and $k = j + 2$ into the undirected graph with degree $k$ isomorphism problem. We return the output from that problem as the output of the undirected graph isomorphism problem. We can see this is correct because the cliques are connected in the same way as the vertices, so an isomorphism exists between $G$ and $H$ iff an isomporphism exists between $G'$ and $H'$. The $k$ part of the problem will be true automatically, so this is the only part that matters. Hence, if a polynomial time algorithm exists for the undirected graph isomorphism problem with degree $k$, one exists for the undirected graph isomorphism problem.

Last, we will reduce the undirected graph isomorphism with degree $k$ problem to the undirected graph isomorphism problem. If we have an algorithm for the undirected graph isomorphism problem, we take our input graphs $G$ and $H$ and input them directly into that algorithm. If the algorithm returns false, we return false. If the algorithm returns true, we must see if every vertex is incident to $k$ edges. We can run through every edge in $O(n^2)$ time and increment the degree of each vertex as we encounter incident edges. Then, we simply run through all the vertices ($O(n)$ time) to check that the degree of each equals $k$. If it does, return true. Otherwise, return false. Hence, if a polynomial time algorithm exists for the undirected graph isomorphism problem, one exists for the undirected graph isomorphism with degree $k$ problem.

9. The input to the Hamiltonian Cycle Problem is an undirected graph $G$. The problem is to find a Hamiltonian cycle, if one exists. A hamiltonian cycle is a simple cycle that spans $G$. Show that the Hamiltonian cycle problem is self reducible. That is, show that if there is a polynomial time algorithm that determines whether a graph has a Hamiltonian cycle, then there is a polynomial time algorithm to find Hamiltonian cycles.
   Solution:

13. Show that the following problem is $NP$-hard:
    INPUT: A graph $G$. Let $n$ be the number of vertices in $G$.
    OUTPUT: 1 if $G$ contains a simple cycle with $\sqrt{n}$ edges, and 0 otherwise.
    Use the fact the following problem is $NP$-hard: INPUT: A graph $G$.
    OUTPUT: 1 if $G$ contains a simple cycle that spans $g$, and 0 otherwise.

    Solution: