CSCI 570 - Fall 2016 - HW 11

Due: *Nov 26th*

1. Given an n bit positive integer, the problem is to decide if it is composite. Here the problem size is n. Is this decision problem in NP ?

   Yes. For every "yes" instance (the number is composite), a factor of the number is a certificate. Certification proceeds by dividing the number by the factor and making sure that the reminder is zero and also making sure that the certificate is neither 1 nor the input number itself. The factor is at most n bits and verification can be done in time polynomial in n. Thus deciding if a number is composite is in NP.

2. Show that vertex cover remains NP-*complete* even if the instances are restricted to graphs with only even degree vertices.

   Let $\langle G, K\rangle$ be an input instance of $VERTEX$-$COVER$, where $G = (V, E)$ is the input graph.

   Because each edge in E contributes a count of 1 to the degree of each of the vertices with which it is incident, the sum of the degrees of the vertices is exactly $2|E|$, an even number. Hence, there is an even number of vertices in G that have odd degrees.

   Let U be the subset of vertices with odd degrees in G.

   Construct a new instance $\left\langle \bar{G}, k+2\right\rangle$ of $VERTEX$-$COVER$, where $\bar{G} = (V_0, E_0)$ with $V_0 = V \cup \{x, y, z\}$ and $E_0 = E \cup \{(x, y), (y, z), (z, x)\} \cup \{(x, v)|v \in U\}$. In words, we make a triangle with the three new vertices, and then connect one of them (say x) to all the vertices in U.

   The degree of every vertex in V_0 is even. Since a vertex cover for a triangle is of (minimum) size 2, it is clear that $\bar{G}$ has a vertex cover of size $k+2$ if and only if G has a vertex cover of size k.

3. Given an undirected graph $G = (V, E)$, the $HALF$-$CLIQUE$ problem is to decide if there is a subset $A \subseteq V$ of vertices satisfying the following two conditions:

   - $|A| \geq \frac{|V|}{2}$
   - For every pair of vertices $u, v \in A$, if $u \neq v$, then $(u, v) \in E$.

   Show that $HALF$-$CLIQUE$ is in NP-*complete*. You are allowed to use the fact that $INDEPENDENT$-SET is NP-*complete*.

Given a set of vertices A as the certificate, it is easy to verify that the two conditions listed in the question are satisfied. Thus *HALF-CLIQUE* is in *NP*.

We will reduce *INDEPENDENT-SET* to *HALF-CLIQUE* in two steps. The intermediate step utilizes the *CLIQUE* problem.

We begin by defining a few terms that we require. A subset of vertices is called a clique if and only if every distinct pair of vertices in the subset is connected by an edge. Given a graph and an integer m, the *CLIQUE* problem is to decide if the graph has a clique of size m. For a graph G_1, its complement (denoted by $\bar{G_1}$) is defined as the graph that has the same vertex set as G_1, but with the edge incidence inverted. That is, an edge e is in G_1 if and only if e in not in $\bar{G_1}$.

Observe that a set of vertices B is an independent set in G_1 if and only if it is a clique for its complement $\bar{G_1}$. Thus an *INDEPENDENT-SET* instance $\langle G_1, k\rangle$ can be reduced to the *CLIQUE* problem by mapping it to the *CLIQUE* instance $\left\langle \bar{G_1}, m = k\right\rangle$. Thus:

$$\textit{INDEPENDENT-SET} \leq_p \textit{CLIQUE}.$$

Given a *CLIQUE* instance $\langle G_2 = (V_2, E_2), m\rangle$ we next reduce it to a *HALF-CLIQUE* instance. If $m = \frac{|V_2|}{2}$, then we already have a *HALF-CLIQUE* instance.

If $m < \frac{|V_2|}{2}$, then add $|V_2|2m$ new vertices to G_2. Then add an edge between every distinct pair of new vertices. Also, add an edge between every new vertex and every existing vertex as well. The new graph has $2(|V_2|m)$ vertices. The new graph has a clique of size at least $|V_2|m$ if and only if G_2 had a clique of size m (prove this on your own!).

If $m > \frac{|V_2}{2}$, then add $2m|V_2|$ new vertices to G_2 and do not introduce any new edges. The new graph has $2m$ vertices. The new graph has a clique of size at least m if and only if G_2 had a clique of size m (prove this on your own!).

Thus we can conclude that:

$$\textit{CLIQUE} \leq_p \textit{HALF-CLIQUE}$$

From the transitivity of polynomial time reductions, it follows that:

$$\textit{INDEPENDENT-SET} \leq_p \textit{HALF-CLIQUE}$$

4. Given an undirected graph G and a positive integer k, consider the decision problem which asks if a simple path (no repeating vertices) of length at least k exists.

   Is this decision problem in *NP*? Assuming $P \neq NP$, is it in P ?

Yes, the decision problem is in *NP*. A simple path of length k is a certificate and can be verified by traversing the path (making sure all the edges in the path are indeed in G, the length is indeed k and that there are no repeated vertices).

We claim that the problem is *NP-complete* (which assuming $P \neq NP$ would imply that it is not in P).

Call the decision problem in question *K-PATH*.

Consider the *HAMILTONIAN-PATH* problem (*HAM-PATH*) where given a graph with n vertices, we have to decide if it contains a simple path that visits all nodes. Clearly *HAM-PATH* $\leq_p$ *K-PATH*, since *HAM-PATH* is a special case of *K-PATH* (with $k = n - 1$).

It turns out that *HAM-CYCLE* $\leq_p$ *HAM-PATH*, where *HAM-CYCLE* is the *HAMILTONIAN-CYCLE* problem. We show this by the following reduction. Let $\bar{G}$ be the graph input to *HAM-CYCLE*. Choose one vertex $u \in \bar{G}$ and duplicate it, i.e. add another vertex u_0 and for each edge (u, v) add the edge (u_0, v). Also add two more vertices t and t_0 and the edges (t, u) and (t_0, u_0). It is fairly easy to see that the new graph has a Hamiltonian path if and only if G has a Hamiltonian cycle.

By transitivity of polynomial time reductions, it follows that:

$$HAM\text{-}PATH \leq_p K\text{-}PATH$$

Thus *K-PATH* is *NP-complete* which assuming $P \neq NP$ implies that *K-PATH* is not in P.

5. Given an undirected graph with positive edge weights, the *BIG-HAM-CYCLE* problem is to decide if it contains a Hamiltonian cycle C such that the sum of weights of edges in C is at least half of the total sum of weights of edges in the graph. Show that *BIG-HAM-CYCLE* is *NP-complete*. You are allowed to use the fact that deciding if an undirected graph has a Hamiltonian cycle is *NP-complete*.

   The certifier takes as input an undirected graph (the *BHC* instance) and a sequence of edges (certificate). It verifies that the sequence of edges form a Hamiltonian cycle and that the total weight of the cycle is at least half the total weight of the edges in the graph. Thus *BIG-HAM-CYCLE* is in *NP-complete*. We claim that *HAM-CYCLE* is polynomial time reducible to *BIG-HAM-CYCLE*. To see this, given an undirected graph $G = (V, E)$ (instance of *HC*), pick an edge e and set its weight to $|E|$ and assign the rest of the edges a weight of 1. When this weighted graph is fed into the *BIG-HAM-CYCLE* decider black box, it returns "yes" if and only if G has a Hamiltonian cycle containing the edge e. By repeating the above procedure once for every edge e in the graph G, we can decide if the graph has a Hamiltonian cycle.

6. You are given an undirected graph $G = (V, E)$ and for each vertex $v \in V$, you are given a number $p(v)$ (which is either 0 or 1 or 2) that denotes the number of pebbles (stones) placed on v. We will now play a game where the following move is the only move allowed. You can pick a vertex u that contains at least two pebbles, and remove two pebbles from u and add one pebble to one (your choice) of the neighboring vertices of u. The objective of the game is to perform a sequence of moves such that we are left with exactly one pebble in the whole graph. Show that the problem of deciding if we can reach the objective is *NP-complete*.

   Call our decision problem *PEBBLE*. For an instance, given a sequence of moves as a certificate, we can verify efficiently if each move is valid and that we are left with a single pebble in the graph. Thus *PEBBLE* is in *NP*.

   We claim that *HAM-PATH* $\leq_p$ *PEBBLE*. Unlike in most other problems that we have encountered, we will use a black box that solves *PEBBLE* more than once.

   Let $G = (V, E)$ be an instance of *HAM-PATH*. Pick a vertex $s \in V$ as the starting vertex, place 2 pebbles at s and at every other vertex, place 1 pebble. We claim that this instance of the *PEBBLE* problem is an "yes" instance if and only if there is a Hamiltonian path starting from s. To see this, observe that the only allowed move in the first step is to go from s to a neighboring vertex (say u). After this move, there are no pebbles left at s and 2 pebbles at u. By induction, we see that for every vertex that we leave, there are no pebbles left and at every new vertex that we arrive at, there are 2 pebbles. Thus we can never revisit a vertex (otherwise, we get stuck since that vertex would only have a single pebble after arrival). To be left with only a single pebble in the graph, prior to the last move, the whole graph should have had exactly 2 pebbles, both of which on a single vertex. This can happen and can only happen if there is a Hamiltonian path starting from s.

   Thus, by calling the black box once for every starting vertex s, we can decide is G has a Hamiltonian path.

7. Assume that you are given a polynomial time algorithm that decides if a directed graph contains a Hamiltonian cycle. Describe a polynomial time algorithm that given a directed graph that contains a Hamiltonian cycle, lists a sequence of vertices (in order) that forms a Hamiltonian cycle.

   Let $G = (V, E)$ be the input graph. Let A be an algorithm that decides if a given directed graph has a Hamiltonian cycle. Hence $A(G) = 1$.

   Pick an edge $e \in E$ and remove it from G to get a new graph $\bar{G}$. If $A(\bar{G}) = 1$, then there exists a Hamiltonian cycle in $\bar{G}$ which is a subgraph of G, set $G = \bar{G}$. If $A(\bar{G}) = 0$, then every Hamiltonian cycle in G contains e. Put e back into G.

Iterate the above three lines until we are left with exactly $|V|$ edges. Since after each step we are left with a subgraph that contains a Hamiltonian cycle, at termination we are left with the set of edges that forms a Hamiltonian cycle. Starting from an edge, do a BFS to enumerate the edges of the Hamiltonian cycle in order.