Due: Friday, March 13, beginning of tutorial

1. Given an undirected graph $G = (V, E)$ and a subset $V' \subseteq V$, we say $V'$ is a node cover
for $G$ if for every node $v \in V - V'$ there is $v' \in V'$ such that $v$ and $v'$ are connected by
an edge in $G$.

Consider the following decision problem:

**Node-Cover**

Instance:
$\langle G, k \rangle$, where $G$ is an undirected graph and $k$ is a positive integer.

Question:
Does $G$ have a node cover of size $k$?

Prove that **Node-Cover** is NP-complete.

(Do not confuse **Node-Cover** with **Vertex-Cover**, but you may use the fact that the
latter is NP-complete.)

**Suggestion:** Modify a given graph $G$ as follows: If $(u, v)$ is an edge in $G$, put in a
new vertex $w_{uv}$ and new edges connecting $w_{uv}$ with both $u$ and $v$.

**Solution:**
Note: The usual term for node cover is dominating set.

To show that **Node-Cover** is in NP we use the following ‘Guess and Check’ algorithm:

To verify that an instance $\langle G, k \rangle$ of **Node-Cover** is a YES instance we guess a subset $V' \subseteq V$ (where $V$ is the set of vertices of $G$) and verify that $V'$ has size $k$ and for every vertex $v \in V \setminus V'$ there is $v' \in V'$ such that there is an edge between $v$ and $v'$.

(In other words, $V'$ is the certificate showing that that $\langle G, k \rangle$ is in **Node-Cover**.)

To show that **Node-Cover** is NP-hard we show

$$\text{Vertex-Cover} \leq_p \text{Node-Cover}$$

Given an instance $(G, k)$ of **Vertex-Cover**, with $G = (V, E)$, we construct (in polynomial time) an instance $(\hat{G}, \hat{k})$ of **Node-Cover** such that $G$ has a vertex cover of size $k$ iff $\hat{G}$ has a node cover of size $\hat{k}$.

First note that if $G$ has any isolated nodes (i.e. nodes not touching any edge) then these can be deleted from any vertex cover for $G$ and the result is still a vertex cover. Hence we may assume that $k$ is at most the number of non-isolated nodes in $V$, since otherwise $(G, k)$ is trivially a YES instance of **Vertex-Cover** (if $k \leq |V|$) or trivially a NO instance (if $k > |V|$).

Let $\hat{G}$ be obtained from $G$ by first deleting all the isolated nodes of $G$ and then applying the modification given by the **Suggestion:** in the question.

Let $\hat{k} = k$. 
CORRECTNESS PROOF: Note that deleting the isolated nodes of \( G \) does not change the answer to the question of whether \((G, k)\) is a YES instance of VERTEX-COVER, as explained above.

Thus we may assume that \( G \) has no isolated nodes.

Suppose that \( G \) has a vertex cover \( V' \) of size \( k \). Then we claim that \( V' \) is a node cover (i.e. dominating set) of \( \hat{G} \). This is because that every node in \( \hat{G} \) is either in \( V' \) or is touching an edge in \( \hat{G} \) whose other end is a \( V' \).

Conversely, suppose that \( V' \) is a node cover of size \( k \) of \( \hat{G} \). Then modify \( V' \) to form a subset \( V'' \) of \( V \) of size at most \( k \) as follows. For each node in \( V' \) of the form \( w_{uv} \) (i.e. one of the nodes added to \( V \)), delete it if both \( u \) and \( v \) are in \( V' \), and otherwise replace it by a node in \( \{u, v\} \) which is not in \( V' \). Note that each of the triangles \( \{u, w_{uv}, v\} \) must have at least one node in \( V' \) since otherwise the node \( w_{uv} \) would not be adjacent to any node in \( V' \), contrary to the definition of node cover. Hence \( V'' \) is a vertex cover for \( G \) of size at most \( k \). (To increase the size of \( V'' \) to exactly \( k \), add any nodes in \( V \). We may assume that \( k \leq |V| \), since otherwise \((G, k)\) is trivially a NO instance of Vertex-Cover.)

2. Consider the following decision problem:

\textbf{Marking-Set}

\textbf{Instance:}
\[ \langle k, n, S_1, \ldots, S_m \rangle, \] where \( k \) and \( n \) are positive integers given in unary notation, and \( S_i \subseteq \{1, \ldots, n\} \) for \( 1 \leq i \leq m \).

\textbf{Question:}
Is there a subset \( T \subseteq \{1, \ldots, n\} \) such that \( |T| \leq k \) and \( T \cap S_i \neq \emptyset \) for \( 1 \leq i \leq m \).

(i) Show that \textbf{Marking-Set} is in \textbf{NP}.

\textbf{Solution:}
Note that \textbf{Marking-Set} is usually called \textbf{Hitting-Set}.

A certificate for the above instance of \textbf{Marking-Set} is a set \( T \subset \{1, \ldots, n\} \) such that \( |T| \leq k \) and \( T \cap S_i \neq \emptyset \) for \( 1 \leq i \leq m \). Given \( T \), it is easy to verify the conditions in polynomial time.

(ii) Define the corresponding (optimization) search problem \textbf{Marking-Set-Search}.

\textbf{Solution:}
\textbf{Marking-Set-Search}

\textbf{Instance:}
\[ \langle n, S_1, \ldots, S_m \rangle, \] where \( n \) is a positive integer and \( S_i \subseteq \{1, \ldots, n\} \) for \( 1 \leq i \leq m \).

\textbf{Output:}
A set \( T \subseteq \{1, \ldots, n\} \) such that \( |T| \) is as small as possible subject to the conditions \( T \cap S_i \neq \emptyset \) for \( 1 \leq i \leq m \).

(iii) Show that \textbf{Marking-Set-Search} \( p \rightarrow \textbf{Marking-Set} \). (For a definition of \( p \rightarrow \), see page 8 of the Lecture Notes “Turing Machines and Reductions”.)
Solution:
Given an instance \( \langle n, S_1, \ldots, S_m \rangle \) of \textbf{Marking-Set-Search} we are to find (in polynomial time) a smallest hitting set \( T \subseteq \{1, \ldots, n\} \), using an oracle for the decision problem \textbf{Marking-Set}.

The algorithm has two parts. The first part is to find the size \( k \) of the smallest hitting set. This is easily done as follows:

\[
k \leftarrow n \\
\text{for } i : n \text{ downto } 1 \\
\quad \text{if } \langle i, n, S_1, \ldots, S_m \rangle \in \text{Marking-Set} \\
\quad \quad \text{then } k \leftarrow i \\
\text{end for}
\]

Now a hitting set of size \( k \) can be found by removing each element \( i \) one at a time from each \( S_j \) and checking whether there is still a hitting set of size \( k \). If not, then \( i \) must be replaced.

\[
T \leftarrow \{1, \ldots, n\} \\
\text{for } j = 1..m \\
\quad S'_j \leftarrow S_j \\
\text{end for} \\
\text{for } i : 1..n (\ast) \\
\quad \text{if } \langle k, n, S'_1 - \{i\}, \ldots, S'_m - \{i\} \rangle \in \text{Marking-Set} \\
\quad \quad \text{then for } j : 1..m \\
\quad \quad \quad S'_j \leftarrow S'_j - \{i\} \\
\quad \quad \quad T \leftarrow T - \{i\} \\
\quad \text{end for} \\
\text{end for}
\]

Output \( T \)

(\ast) Loop Invariant: There is a hitting set \( T' \subseteq T \) for the original input such that \( |T'| = k \) and \( T - T' \subseteq \{i, \ldots, n\} \).

The Loop Invariant is proved by induction on the number of times the loop has been executed. After exiting from the Loop, in effect \( i = n + 1 \), so \( T - T' = \emptyset \), so \( T = T' \) is a hitting set of size \( |T| = k \).

3. Recall that if \( G \) is an undirected graph, then a \textit{Hamiltonian path} in \( G \) is a path that hits each node exactly once, and a \textit{Hamiltonian cycle} in \( G \) is a cycle that hits each node exactly once.

Let \textbf{UHamPath} be the set of triples \( \langle G, s, t \rangle \) such that \( G \) is an undirected graph with a Hamiltonian path from \( s \) to \( t \), and let \textbf{UHamCycle} be the set of undirected graphs \( G \) which have a Hamiltonian cycle.

Give explicit reductions showing each of the following (and justify your reductions):
i) UHamCycle \leq_p CNF-SAT

Solution:
Given an undirected graph $G = (V, E)$ we want to find (in polynomial time) a Boolean CNF formula $\varphi$ such that $G$ has a Hamiltonian cycle iff $\varphi$ is satisfiable.

Suppose $V = \{v_1, \ldots, v_n\}$. The formula $\varphi$ has variables $p_{ij}, 1 \leq i, j \leq n$, where $p_{ij}$ is intended to mean that node $v_i$ is the $j$-th node in the Hamiltonian cycle.

Then $\varphi = \psi_1 \land \psi_2 \land \psi_3$, where the $\psi_i$ are defined as follows:

$\psi_1$ asserts that each node $v_i$ occurs somewhere on the cycle:

$$\psi_1 = \bigwedge_{i=1}^{n} (p_{i1} \lor \cdots \lor p_{in})$$

$\psi_2$ asserts that no two nodes can both be at the same position in the cycle:

$$\psi_2 = \bigwedge_{1 \leq i < k \leq n} \bigwedge_{1 \leq j \leq n} (\overline{p_{ij}} \lor \overline{p_{kj}})$$

$\psi_3$ asserts that successive nodes in the cycle must form an edge in $G$:

$$\psi_3 = \bigwedge_{(v_i, v_k) \notin E} \left[ (\overline{p_{in}} \lor \overline{p_{k1}}) \land \bigwedge_{j=1}^{n-1} (\overline{p_{ij}} \lor \overline{p_{k,j+1}}) \right]$$

Justification: If $G$ has a Hamiltonian cycle, then we assign each variable $p_{ij}$ to be 1 (true) iff $v_i$ is the $j$-th node in the cycle. Then each of $\psi_1, \psi_2, \psi_3$ are true under this assignment, because the English description preceding each of these formulas is true, and the formulas correctly implement the description. Hence $\varphi$ is satisfiable.

Conversely, suppose that $\varphi$ is satisfied by some assignment $\tau$. Then since $\tau$ satisfies $\psi_1$ and $\psi_2$ it follows that for each $j, 1 \leq j \leq n$ there is precisely one $i = i_j$ such that $\tau$ satisfies $p_{i_j,j}$. Then the sequence of nodes

$$v_{i_1}, v_{i_2}, \ldots, v_{i_n}$$

forms a permutation of the sequence $v_1, v_2, \ldots, v_n$, and $\psi_3$ assures that there is an edge between adjacent nodes in the sequence, and between $v_{i_n}$ and $v_{i_1}$.

ii) UHamCycle \leq_p UHamPath

Solution:
Given an instance $\langle G \rangle$ of UHamCycle we must compute (in polynomial time) an instance $\langle G', s, t \rangle$ of UHamPath such that

$G$ has a Hamiltonian cycle iff $G'$ has a Hamiltonian path from $s$ to $t$.

Suppose $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$. We assume that $n \geq 3$, since every cycle must have at least 3 nodes, so if $n < 3$ then make $\langle G', s, t \rangle$ any NO instance of UHamPath.
The idea for $G'$ is to split node $v_1$ into the two nodes $s$ and $t$, and connect both $s$ and $t$ to the same nodes that $v_1$ is connected to.

Thus let $G' = (V', E')$, where $V' = \{s, t, v_2, \ldots, v_n\}$, and

$$E' = E \cup \{(s, v_j), (t, v_j) \mid (v_1, v_j) \in E\} - \{(v_1, v_j) \mid (v_1, v_j) \in E\}$$

**Proof of correctness:**

($\Rightarrow$): Suppose the sequence $v_1, v_{i_2}, \ldots, v_{i_n}, v_1$ forms a Hamiltonian cycle in $G$. Then $s, v_{i_2}, \ldots, v_{i_n}, t$ forms a Hamiltonian path from $s$ to $t$ in $G'$.

($\Leftarrow$): Suppose $s, v_{i_2}, \ldots, v_{i_n}, t$ forms a Hamiltonian path from $s$ to $t$ in $G'$. Then $v_1, v_{i_2}, \ldots, v_{i_n}, v_1$ forms a Hamiltonian cycle in $G$. 
