Announcements

- HW solution session on Sunday as usual

New Phrases

- We say a problem is in **NP** if there exists a polynomial time verifier TM $V$ and a constant $k > 0$ such that for all $x \in \Sigma^*$:
  
  - if $x \in L$, then there exists a certificate $u$ with $|u| \leq |x|^k$ such that $V(x, u)$ accepts.
  
  - if $x \notin L$, then for all $u \in \Sigma^*$, $V(x, u)$ rejects.

- We say there is a polynomial-time many-one reduction from $A$ to $B$ if there is a polynomial-time computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that $x \in A$ if and only if $f(x) \in B$. We write this as $A \leq^P_m B$. (We also refer to these reductions as Karp reductions.)

- A problem $Y$ is **NP-hard** if for every problem $X \in \text{NP}$, $X \leq^P_m Y$.

- A problem is **NP-complete** if it is both in **NP** and **NP-hard**.

No Privacy

**DOUBLE-CLIQUE**: Given a graph $G = (V, E)$ and a natural number $k$, does $G$ contain two vertex-disjoint cliques of size $k$ each?

Show **DOUBLE-CLIQUE** is **NP-complete**.

**Edge Cover-Up**

Let $G = (V, E)$ be a graph. A vertex covering of $G$ is a set $C \subseteq V$ such that for every edge $\{x, y\} \in E$, either $x \in C$ or $y \in C$ (a set of vertices such that every edge is incident to at least one vertex in the set). An independent set in $G$ is a set $S \subseteq V$ such that for any $u, v \in S, \{u, v\} \notin E$ (a set of vertices such that no edge connects two vertices in the set). Define the following languages:

**VERTEX-COVER**: $\{\langle G, k \rangle : G$ is a graph, $k \in \mathbb{N}^+, G$ contains a vertex covering of size $k$ $\}$

**IND-SET**: $\{\langle G, k \rangle : G$ is a graph, $k \in \mathbb{N}^+, G$ contains an independent set of size $k$ $\}$

Show that **VERTEX-COVER** $\leq^P_m$ **IND-SET** and **IND-SET** $\leq^P_m$ **VERTEX-COVER**

What is nondeterministic about NP?

A **nondeterministic TM** (NDTM) is a normal Turing machine where the transition function $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is replaced by a transition function of a different type $\delta' : Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\})$. This means that if an NDTM is in state $q$ and is reading the letter $\sigma$, there could be any number of choices for the next state, the letter to write, and the direction to move (in this sense, NDTMs are to normal TMs as MFAs are to DFAs). One way to think about how an NDTM computes is that at each step, it makes all the possible choices for the next state/letter/direction and continues simulating the results of all of those choices in parallel. Thus, at any point during its computation, an NDTM has not just one but many active computation paths. An NDTM accepts an input $x$ when any of its computation paths terminate in an accepting state, and it rejects when all of its computation paths terminate in a rejecting state.

Prove that NP is the class of languages decided by some polynomial-time NDTM.
(Extra) Looping Around
Show that the HALTS is NP-hard.

(Bonus) Hard Cut
Define (the decision version of) the MAX-CUT problem as follows:
MAX-CUT: \{\langle G, k \rangle : G's vertices may be colored with two colors in a way that cuts at least \( k \) edges\}.
Prove that MAX-CUT is NP-hard. This is slightly difficult; try reducing from IND-SET.