1. Recall that \( L_u = \{ \langle M, w \rangle \mid M \text{ accepts } w \} \) is language of a UTM, and \( L_{\text{HALT}} = \{ \langle M \rangle \mid M \text{ halts on blank input} \} \) is the Halting language.
   - Let \( L_{374A} = \{ \langle M \rangle \mid M \text{ accepts at least 374 distinct input strings} \} \). Prove that \( L_{374A} \) is undecidable.
   - Prove that \( L_u \leq L_{\text{HALT}} \)
   - **Not to submit:** Prove that \( L_{\text{HALT}} \leq L_u \).

2. Consider an instance of the Satisfiability Problem, specified by clauses \( C_1, \ldots, C_m \) over a set of Boolean variables \( x_1, \ldots, x_n \). We say that the instance is monotone if each term in each clause consists of a nonnegated variable; that is each term is equal to \( x_i \), for some \( i \), rather than \( \overline{x}_i \). Monotone instance of Satisfiability are very easy to solve: They are always satisfiable, by setting each variable equal to 1.

   For example, suppose we have the three clauses
   \[
   (x_1 \lor x_2), (x_1 \lor x_3), (x_2 \lor x_3)
   \]
   This is monotone, and indeed the assignment that sets all three variables to 1 satisfies all the clauses. But we can observe that this is not the only satisfying assignment; we could also have set \( x_1 \) and \( x_2 \) to 1 and \( x_3 \) to 0. Indeed, for any monotone instance, it is natural to ask how few variables we need to set to 1 in order to satisfy it.

   Given a monotone instance of Satisfiability, together with a number \( k \), the problem of **Monotone Satisfiability with Few True Variables** asks: Is there a satisfying assignment for the instance in which at most \( k \) variables are set to 1? Describe a polynomial time reduction from Vertex Cover to this problem. You should also prove the correctness of the reduction.

3. Given an undirected graph \( G = (V, E) \), a partition of \( V \) into \( V_1, V_2, \ldots, V_k \) is said to be a clique cover of size \( k \) if each \( V_i \) is a clique in \( G \). **CLIQUE-COVER** is the following decision problem: given \( G \) and integer \( k \), does \( G \) have a clique cover of size at most \( k \)?
   - **Not to submit:** Prove that **CLIQUE-COVER** is NP-Complete? For this part you just need to describe the reduction clearly, no proof of correctness is necessary. **Hint:** Use variable \( x(u, i) \) to indicate that node \( u \) is in partition \( i \).
   - Describe a polynomial-time reduction from \( k \)-Color to **CLIQUE-COVER**.
You should also prove the correctness of the reductions.

4. **Not to submit:** We call an undirected graph an *eight-graph* if it has an odd number of nodes, say $2n - 1$, and consists of two cycles $C_1$ and $C_2$ on $n$ nodes each and $C_1$ and $C_2$ share exactly one node. See figure below for an eight-graph on 7 nodes.

![Eight-graph diagram](image)

Given an undirected graph $G$ and an integer $k$, the EIGHT problem asks whether or not there exists a subgraph which is an eight-graph on $2k - 1$ nodes. Prove that EIGHT is NP-Complete.