Guidelines:

Please submit your assignment to the TA/lecturer at the beginning of the appropriate lecture. **Solutions should be typed**, drawings can be done by hand.

You are encouraged to work in groups of up to three people, however we ask that you dedicate enough time to think about each problem by yourself before consulting others. Moreover, you must write up each problem solution by yourself without assistance. You are asked to identify your collaborators on problem sets. If you did not work with anyone, please write “Collaborators: none”.

You may consult the course notes and the text (Sipser).

The assignment questions may be similar to questions on problem sets from past offerings of this course or courses at other universities. Using any preexisting solutions from these or any other sources is strictly prohibited.

Each of the Questions 1-6 is worth 10 points, please attempt all problems. Please provide full proofs, unless stated otherwise.

In all that follows, in order to prove that a language $L$ is NP-hard you may reduce any of the languages that were shown to be NP-hard in class (3SAT, CLIQUE, SUBSET SUM), or any of the problems in this problem set that you proved to be NP-hard, to $L$.

Question 1.

In this problem, $K(x)$ denotes the Kolmogorov complexity of the string $x$.

1. (5 points) Use Kolmogorov complexity to give an alternate proof for the claim that there are exist infinitely many primes.

   **Hint:** Assume that there are finitely many primes, and use this assumption to prove that every integer of length $n$ has a representation using less than $n$ bits. Derive a contradiction.

2. (5 points) Let $f : \Sigma^* \to \mathbb{Z}$ be a computable function. Show that if $\forall x : f(x) \leq K(x)$, then $f$ must be bounded. That is, there must be some $C$ such that $\forall x : f(x) \leq C$.

   **Remark:** This shows that the Kolmogorov complexity of a string cannot even be lower bounded in a meaningful way by a computable function.
Question 2.

Recall that the Cook-Levin Theorem proves that 3SAT is \textbf{NP}-complete, and therefore is highly unlikely to be in \textbf{P}. In this question you will prove that 2SAT is in fact in \textbf{P}. (Recall that 2SAT is the set of satisfiable 2CNF formulas).

Consider the following construction: Let $G = G_\varphi$ be a directed graph on $2m$ nodes labeled with $x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_m, \bar{x}_m$. For each clause $(x \lor y)$ of $\varphi$, place the two edges $(\bar{x}, y)$ and $(\bar{y}, x)$ in the graph $G$. A cycle of $G$ containing both $x_i$ and $\bar{x}_i$, for some $i \in [m]$, is called an \textit{inconsistency cycle} of $G$.

1. (3 points) Prove that if $G$ contains an inconsistency cycle, then $\varphi$ is unsatisfiable.

2. (5 points) Prove that if $G$ does not contain an inconsistency cycle, then $\varphi$ is satisfiable.

    \textit{Hint:} First show that if there is a path from $\alpha$ to $\beta$ in $G$, where $\alpha$ and $\beta$ are literals, there is also a path from $\bar{\beta}$ to $\bar{\alpha}$ in $G$.

3. (2 points) Give a polynomial time algorithm for 2SAT. What is the running time of your algorithm?

Question 3.

1. (3 points) Let $G = (E, V)$ be an undirected graph. A set $S \subseteq V$ is an \textit{independent set} of $G$ if no two vertices in $S$ are connected by an edge. Define the language

   \[ \text{IND SET} = \{(G, k) : G \text{ is an undirected graph with an independent of size } \geq k\} \]

   Prove that \text{IND SET} is \textbf{NP}-complete by showing that CLIQUE $\leq_P$ \text{IND SET}.

2. (3 points) Let $G = (E, V)$ be an undirected graph. A set $S \subseteq V$ is a \textit{vertex cover} of $G$ if every edge of $G$ touches at least one of vertices in $S$. Define the language

   \[ \text{VERTEX COVER} = \{(G, k) : G \text{ is an undirected graph with a vertex cover of size } \leq k\} \]

   Prove that \text{VERTEX COVER} is \textbf{NP}-complete by showing that \text{IND SET} $\leq_P$ \text{VERTEX COVER}.

3. (4 points) Consider the following \textit{polynomial time} algorithm for finding a vertex cover of a graph: Start with $S = \emptyset$. While there is an edge for which neither of its endpoints is in $S$, select any such edge and insert both of its endpoints to $S$.

   Prove that the suggested algorithm always gives a 2-approximation of the minimal vertex cover. That is, show that if the minimal vertex cover of $G$ is of size $k$, the algorithm produces a vertex cover of $G$ of size at most $2k$. 
**Question 4.**

Let $\mathcal{C} = \{S_1, S_2, S_3, \ldots, S_n\}$ be a collection of sets and define $U = \bigcup_i S_i$. We say that $T \subseteq \mathcal{C}$ is a *cover* if every $u \in U$ occurs in at least one $S_i \in T$. Define the language

$$ \text{SET COVER} = \{(\mathcal{C}, k) : \text{there is a cover } T \subseteq \mathcal{C} \text{ with } |T| \leq k\}. $$

Prove that \text{SET COVER} is \textbf{NP}-complete.

**Question 5.**

A list of positive integers $(a_1, a_2, \ldots, a_n)$ is *partitionable* if there is a subset $T \subseteq \{1, 2, \ldots, n\}$ for which $\sum_{i \in T} a_i = \sum_{i \not\in T} a_i$. Define the language

$$ \text{PARTITION} = \{S = \{a_1, a_2, \ldots, a_n\} : S \text{ is partitionable}\} $$

Prove that \text{PARTITION} is \textbf{NP}-complete.

**Question 6.**

1. (4 points) Assume that $P = \text{NP}$. Show that there exists a polynomial time algorithm that given $\phi \in 3\text{sat}$, produces a satisfying assignment for $\phi$.
   
   *Hint:* Use the satisfiability tester repeatedly to find the assignment bit-by-bit.

2. (6 points) Let $A \subseteq \{0, 1\}^*$ be a language. Define

   $$ A_{<n} = \{x \in A : |x| < n\}. $$

   Assume that $A$ is *downward self-reducible*, meaning that it is possible to determine in polynomial time if $x \in A$ using the results of queries of the form “$y \in A_{<|x|}$?”. The queries may be *adaptive*, meaning that the polynomial time procedure may choose later queries depending on the results of earlier ones. Show that $A \in \text{PSPACE}$.

   *Remark:* Observe that $\text{tqbf}$ satisfies the above property.