1. Closure of NP
Show that the class $NP$ is closed under the Kleene star operation (i.e., for any language $L$, if $L \in NP$, then $L^* \in NP$ as well).

2. NP-completeness

(a) Define $\text{Root-CLIQUE} = \{\langle G \rangle \mid \text{graph } G \text{ has a clique of size at least } \sqrt{n}, \text{ where } n \text{ is the number of vertices in } G\}$. Prove that $\text{Root-CLIQUE}$ is $NP$-complete, using a reduction from $\text{CLIQUE}$.

(b) Define $\text{Twice-SAT} = \{\langle \phi \rangle \mid \phi \text{ is a cnf formula with at least two satisfying assignments}\}$. Prove that $\text{Twice-SAT}$ is $NP$-complete, using a reduction from $\text{SAT}$.

(c) Define $\text{PARTITION} = \{\langle a_1, \ldots, a_n \rangle \mid a_1, \ldots, a_n \text{ are positive integers in binary such that there is partition of } \{1, \ldots, n\} \text{ into two disjoint subsets } S \text{ and } T, \text{ where } S \cup T = \{1, \ldots, n\}, \text{ so that } \sum_{i \in S} a_i = \sum_{j \in T} a_j\}$. Prove that $\text{PARTITION}$ is $NP$-complete, using a reduction from $\text{SubsetSum}$.

3. PSPACE-completeness
For a language $A$, an $A$-oracle Turing machine is a Turing machine $M$ that may ask if $y \in A$, for any string $y$, and receive the correct answer in a single step. (That is, for such an $A$-oracle machine, checking membership in the language $A$ is free.) We say that a language $L$ is in $P^A$, if there is a deterministic polytime $A$-oracle TM that decides $L$. Similarly, we say that $L \in NP^A$ if there is a nondeterministic polytime $A$-oracle TM deciding $L$.

Prove that $NP^{TQBF} = P^{TQBF}$, where $TQBF$ is the language of true quantified boolean formulas (which we showed in class to be $PSPACE$-complete).
4. Randomized complexity
Suppose that $\text{SAT} \in \text{BPP}$. Under this assumption, argue that $\text{SAT} \in \text{RP}$.

5. NP-hardness of approximation
Recall that a Minimization problem is efficiently $\alpha$-approximable (for some $\alpha \geq 1$) if there is a polytime algorithm that finds an approximate solution whose value $\text{APPROX}$ satisfies: $\text{OPT} \leq \text{APPROX} \leq \alpha \cdot \text{OPT}$, where $\text{OPT}$ is the value of an optimal solution.

Consider the TSP problem:

Given a weighted complete graph $G = (V, E)$ on $n$ vertices, with positive integer weights (in binary) on its edges $w : E \to \mathbb{Z}^+$, find the cost of a minimal-cost tour (where a tour is a Hamiltonian cycle in $G$ and its cost is the sum of edge weights for the edges in the cycle).

Show that, for every polynomial $\alpha(n) = n^c$, for $c \geq 0$, this problem is $\text{NP}$-hard to $\alpha(n)$-approximate. That is, show that if for some $\alpha(n) = n^c$, there is a deterministic polytime $\alpha(n)$-approximation algorithm for TSP on $n$-vertex graphs, then $\text{P} = \text{NP}$.