1. **Equivalence of Search and Decision.** Given an \( \text{NP} \) language \( L \), let a **witness-finder** for \( L \) be a polynomial-time algorithm \( M \) that actually outputs a yes-witness \( y \) for \( x \) whenever \( x \in L \), but could behave arbitrarily if \( x \notin L \). (In other words, if \( V(x,y) \) is the verifier for \( L \), then \( V(x,M(x)) \) accepts whenever \( x \in L \).

(a) Show that if every \( \text{NP} \) language has a witness-finder, then \( \text{P} = \text{NP} \).

(b) Show that if \( \text{P} = \text{NP} \), then every \( \text{NP} \) language has a witness-finder.

2. Let \( \text{EXACT4SAT} \) be the following problem:

- Given a Boolean formula \( \varphi \), consisting of an AND of clauses involving exactly 4 distinct literals each (such as \( (x_2 \lor \neg x_3 \lor \neg x_5 \lor x_6) \)), decide whether \( \varphi \) is satisfiable.

Show that \( \text{EXACT4SAT} \) is \( \text{NP} \)-complete. You can use the fact, which we proved in class, that \( \text{3SAT} \) is \( \text{NP} \)-complete.

3. Let \( \text{DOUBLESAT} \) be the following problem:

- Given as input a Boolean circuit \( C \), decide whether there are two or more input assignments \( x \in \{0,1\}^n \) such that \( C(x) = 1 \).

Show that \( \text{DOUBLESAT} \) is \( \text{NP} \)-complete.

4. Let \( G \) be an undirected graph with \( n \) vertices. Then a **Hamilton path** is a simple path in \( G \) that visits each vertex once (i.e., has \( n \) vertices and \( n - 1 \) edges), while a **Hamilton cycle** is a simple cycle in \( G \) that visits each vertex once (i.e., has \( n \) vertices and \( n \) edges). Let \( \text{HAMPATH} \) and \( \text{HAMCYCLE} \) be the problems of deciding whether \( G \) has a Hamilton path and Hamilton cycle respectively, given \( G \) as input.

(a) Show that if \( G \) has a Hamilton cycle, then \( G \) also has a Hamilton path.

(b) Give an example of a graph \( G \) that has a Hamilton path but no Hamilton cycle.

(c) Give a polynomial-time reduction from \( \text{HAMCYCLE} \) to \( \text{HAMPATH} \).

(d) Give a polynomial-time reduction from \( \text{HAMPATH} \) to \( \text{HAMCYCLE} \).

(Together, parts c and d imply that \( \text{HAMPATH} \) and \( \text{HAMCYCLE} \) are polynomial-time equivalent. Since \( \text{HAMCYCLE} \) is a famous \( \text{NP} \)-complete problem, this immediately implies that \( \text{HAMPATH} \) is \( \text{NP} \)-complete as well.)

5. In the **quadratic programming** (\( \text{QUADPROG} \)) problem, the input is a system of equalities and inequalities, each involving polynomials of degree at most 2 (with integer coefficients) in \( n \) real variables \( x_1, \ldots, x_n \). The “size” of the input can be taken to be \( n + m \), where \( n \) is the number of variables, and \( m \) is the number of bits needed to write down the constraints. The problem is to decide whether there
exists an assignment to $x_1, \ldots, x_n$ that satisfies all the constraints simultaneously. As an example, the system

$$x_1 + x_2 \leq 1$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$
$$4x_1x_2 \geq 1$$

can be satisfied by setting $x_1 = x_2 = 1/2$, but if we replaced the last inequality by $x_1x_2 \geq 1$, then the system would be unsatisfiable.

(a) Show that QUADPROG is NP-hard, by reduction from any problem that was already proved NP-hard in class. [Hint: 3COLORING would be a good choice.]

(b) What is a difficulty in showing that QUADPROG $\in$ NP (the other condition needed for QUADPROG to be NP-complete)?

6. Suppose problem $X$ is proved NP-complete, by a polynomial-time mapping reduction that maps size-$n$ instances of SAT to size-$n^3$ instances of problem $X$. And suppose that someday, some genius manages to prove that SAT requires $\Omega(c^n)$ time, for some constant $c > 1$. Then what can you conclude about the time complexity of problem $X$?

7. Recall that SUBSET SUM is the following problem: given as input positive integers $x_1, \ldots, x_n, k$ encoded in binary, decide whether there exists a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} x_i = k$. In class, we asserted without proof that SUBSET SUM is NP-complete. Now let UNARY SUBSET SUM be the same problem, except that $x_1, \ldots, x_n, k$ are encoded in unary notation, so that the “input length” is $x_1 + \cdots + x_n + k$ instead of $\log x_1 + \cdots + \log x_n + \log k$. Show that UNARY SUBSET SUM $\in$ P, by describing a polynomial-time algorithm to solve it.

8. The Polynomial Hierarchy. Let $L = \{ \langle C \rangle : \text{there is a smaller circuit that computes the same Boolean function as } C \}$, where $C$ is a circuit (say of NAND gates), and $\langle C \rangle$ is its description. Show that $L \in \text{NP}^{\text{NP}}$. 

2