Problem 1

Proof. Does not apply.

Problem 2

Answer. Consider the language \( A = \{1\} \). This language is clearly in \( P \). We now show that 3SAT can be reduced to \( A \) under the new reduction.

Since 3SAT \( \in NP \subseteq EXP \), 3SAT can be decided in exponential time. Thus, give any 3CNF formula \( \varphi \), we can let \( f \) be the following function: first decide in exponential time if \( \varphi \) is satisfiable. If it is, then let \( f(\varphi) = 1 \), otherwise let \( f(\varphi) = 0 \). Note that \( f \) can be computed in exponential time, and \( \varphi \in 3SAT \) if and only if \( f(\varphi) \in A \).

Problem 3

Answer. First, QUADEQ is in \( NP \) since a certificate is just a satisfying assignment to the variables. We now reduce 3SAT to QUADEQ.

To do this, given any 3CNF formula \( \varphi \), we view each variable \( x_i \) as a variable in \( \{0, 1\} \), and we create another \( n \) variables \( x_i', i = 1, \ldots, n \) such that \( \forall i, x_i' = 1 - x_i \). This gives us \( n \) equations. Now, for each clause in \( \varphi \), we create a system of quadratic equations such that this clause is satisfiable if and only if the system of equations is satisfiable. Finally we combine all the equations we have obtained. Note that \( \varphi \in 3SAT \) if and only if there is an assignment that satisfies all clauses, thus this happens if and only if there is an assignment that satisfies the whole system of equations.

Consider the \( t \)'th clause, without loss of generality assume it is \( x_i \lor x_j \lor x_k \) (if the literals are negations then we use \( \overline{x_i} \) instead). We create a new variable \( z_t \) which is used to represent \( z_t = x_i \lor x_j \). We can represent this as a quadratic equation: \( z_t = 1 - (1 - x_i)(1 - x_j) \). Now to ensure that \( z_t \lor x_k = 1 \), we add the equation \( 1 = 1 - (1 - z_t)(1 - x_k) \). If there is an assignment that satisfies \( x_i \lor x_j \lor x_k \), then the corresponding values of \( x_i, x_j, x_k, z_t \) satisfy these two equations, and vice versa. Note that if the clause contains less than 3 literals, then we can skip the new variable \( z_t \) and use just one equation. The whole reduction is in polynomial time and thus 3SAT \( \leq_p \) QUADEQ. Therefore QUADEQ is NP-complete.

Problem 4

Answer. (a) First, let’s prove the “only if” part. Assume graph \( G \) is bipartite. Suppose it contains a cycle \( C \) that has an odd number of nodes. Let \( v_i, i = 1, 2, \ldots, n \) be the nodes in \( C \) where \( v_i \) connects \( v_{i+1} \) for \( i = 1, 2, \ldots, n - 1 \) and \( v_n \) connects \( v_1 \). Assume \( G \) can be divided into two sets \( U, V \).

Without loss of generality, assume \( v_1 \in U \). As all the edges are between \( U \) and \( V \), we know that \( \forall i \in [n] \) such that \( i \) is odd, \( v_i \) is in \( U \). Also \( \forall i \in [n] \) such that \( i \) is even, \( v_i \) is in \( V \). Now consider the edge \((v_n, v_1)\). As \( C \) has an odd number of nodes, \( n \) is odd. So \( v_n \) and \( v_1 \) are both in \( U \). This edge contradicts the assumption that \( G \) is bipartite.

Second, let’s prove the “if” part. Assume graph \( G \) does not contain a cycle that has an odd number of nodes. Divide the nodes of \( G \) into \( U \) and \( V \) in the following way.
Consider each connected component $C$ of $G$. Choose an arbitrary node $u$ in $C$. Put $u$ into $U$. For any other node $v$ in $C$, if the length of the shortest path from $u$ to $v$ in $C$ is odd, put $v$ into $V$. Otherwise, put $v$ into $U$. Suppose after we have done this, there is an edge $e = (u_e, v_e)$ such that $u_e, v_e$ are both in $U$ or $V$. Without loss of generality, suppose that they are both in $U$. Consider the cycle formed by the shortest path $P_1$ from $u$ to $u_e$, the shortest path $P_2$ from $u$ to $v_e$ and $e$. The length of $P_1$ and $P_2$ are both even. So the length of the cycle is odd which contradicts our assumption.

(b) We know that $NL = coNL$. So we are going to show that $\overline{BIPARTITE} \in NL$.

According to the definition of $BIPARTITE$, $x \in BIPARTITE$ if and only if the graph $G_x$ (which is the graph described by $x$) contains a cycle that has an odd number of nodes. So the certificate is a cycle that has an odd number of nodes. This cycle can be represented by a string $u$ that records the nodes in this cycle one by one. That is, if the cycle $C = \langle v_1, \ldots, v_n \rangle$ such that $v_i$ connects to $v_{i+1}$ for $i = 1, 2, \ldots, n - 1$, then $u = s_{v_1} \circ \cdots \circ s_{v_n}$ where $s_{v_i}$ is the string representation of $v_i$. The verifier $M$ reads these nodes one by one. It checks whether $v_i$ is connected to $v_{i+1}$ for $i = 1, 2, \ldots, n - 1$ and whether $v_n = v_1$. Also $M$ counts the number of total nodes in this cycle. If it is a cycle with odd number of nodes, $M$ outputs 1. Otherwise, it outputs 0. The total space used by $M$ is $O(\log n)$ because it only needs to record the number of all nodes in the cycle and the indices of at most two nodes of the cycle. The time complexity of $M$ is a polynomial of the input length. Also $M$ only reads $u$ once. This proves that $\overline{BIPARTITE} \in NL$. So $\overline{BIPARTITE} \in coNL$. Thus $BIPARTITE \in NL$. □

Problem 5

Proof: Consider the complement language of $MIN − FORMULA$, that is $\overline{MIN − FORMULA}$. We know that given any Boolean formula $\varphi$, $\varphi \in \overline{MIN − FORMULA}$ if and only if the following holds:

\[ \exists \text{ a Boolean formula } \phi \text{ such that } \forall x \in \{0, 1\}^n, \phi \text{ is shorter than } \varphi \text{ and } \phi(x) = \varphi(x). \]

Note that the condition $\phi$ is shorter than $\varphi$ and $\phi(x) = \varphi(x)$ can be verified by a polynomial time TM. Thus $\overline{MIN − FORMULA} \in \Sigma_2^P$. Hence $MIN − FORMULA \in \Pi_2^P$. If $P = NP$, then we know $PH = P$. Thus $\Sigma_2^P = \Pi_2^P = P$ and $MIN − FORMULA \in P$. □