This exam contains 6 pages (including this cover page) and 5 problems. Some problems are easier and others are harder, and they are not necessarily arranged from easy to hard. So, if you get stuck on one problem, it may be a good idea to try others first. Try to do as many problems as you can. However, it will probably be a better strategy to solve a few problems correctly, rather than to try all problems but end up doing them all wrong (in which case you will receive very little credit).

Unless otherwise stated, you are required to show your work on each problem on this exam. The following rules apply:

- You may use directly any results proved in the textbook, in class, or in your homework. Otherwise, please prove your claims.

- Organize your work, in a reasonably neat and coherent way. Try to be concise and accurate in your argument. Lengthy paragraphs that are hard to understand may receive very little credit.

- Mysterious or unsupported answers will not receive full credit. An incorrect answer supported by substantially correct explanations might still receive partial credit.

**Problem Points Score**

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>100</strong></td>
<td></td>
</tr>
</tbody>
</table>

Sign the following:

I agree to complete this exam without unauthorized assistance from any person, materials or device.

Sign and Date: ____________________________
1. (20 points) let \( \Sigma = \{0, 1\} \). For each \( k \geq 1 \), let \( D_k \) be the language consisting of all strings that have at least one 1 among the last \( k \) symbols. Thus \( D_k = \Sigma^*1(\Sigma \cup \varepsilon)^{k-1} \). Describe a DFA with at most \( k + 1 \) states that recognizes \( D_k \), both in terms of a state diagram and a formal description. You don’t need to prove your construction works.
2. (20 points) Recall that when we define polynomial-time Karp reduction \( \leq_p \), we say that \( L \leq_p L' \) if there exists a polynomial time computable function \( f : \{0, 1\}^* \rightarrow \{0, 1\}^* \) such that for every \( x \in \{0, 1\}^* \), \( x \in L \) iff \( f(x) \in L' \). Suppose that we relax the restriction and allow \( f \) to be computable in exponential time (i.e., \( 2^{\text{nc}} \) for any constant \( c > 0 \) if \( n \) is the input size), with all the other requirements remaining the same. Show that under this new definition 3SAT can be reduced to some problem in \( P \).
3. (20 points) Given a system of quadratic equations over variables \( x_1, \cdots, x_n \), each of the form

\[
\sum_{i<j} a_{ij} x_i x_j + \sum_i b_i x_i = c,
\]

where all the variables and coefficients are in \( \{0, 1\} \), and all arithmetic is modulo 2, we say that the system is satisfiable if there is an assignment to the variables satisfying all the equations.

Let \( \text{QUADEQ} \) denote the language of all satisfiable systems of quadratic equations over 0/1 variables. Show that \( \text{QUADEQ} \) is \( \text{NP} \)-complete.

**Hint:** Reduce from 3SAT.
4. (20 points) An undirected graph is bipartite if its nodes may be divided into two sets so that all edges go from a node in one set to a node in the other set.

(a) (10 points) Show that a graph is bipartite if and only if it doesn’t contain a cycle that has an odd number of nodes.

(b) (10 points) Use (a) to show that the following language is in NL:

\[ BIPARTITE = \{ \langle G \rangle | G \text{ is a bipartite graph.} \} \]

**Hint:** Use \( \text{NL} = \text{coNL} \).
5. (20 points) Say that two Boolean formulas are *equivalent* if they have the same set of variables and are true on the same set of assignments to those variables (i.e., they describe the same Boolean function). A Boolean formula is *minimal* if no shorter Boolean formula is equivalent to it. Let $\text{MIN} - \text{FORMULA}$ be the collection of minimal Boolean formulas.

Show that, if $P = NP$, then $\text{MIN} - \text{FORMULA} \in P$.

**Hint:** Use the relations between $P$, $NP$, and $coNP$. 