Exercise 1 (2pt). Define the complexity class

$$DP = \{ A \cap B \mid A \in NP, B \in coNP \}.$$ 

Let $G = (V, E)$ be an undirected graph. A subset $D \subseteq V$ of vertices is called a dominating set if for every $v \in V$ it holds that (i) $v \in D$ or (ii) there is some $u \in D$ and there is an edge between $u$ and $v$ in $G$. Consider the problem Exact-DS:

$$\text{Exact-DS} = \{ \langle G, k \rangle \mid G \text{ is an undirected graph that has a dominating set of size } k \text{ but has no dominating set of size } k - 1 \}.$$ 

(a) Prove that Exact-DS $\in$ DP.

(b) Prove that if DP $\subseteq$ NP, then the Polynomial Hierarchy collapses.

Exercise 2 (3pt). Prove that there is no polynomial-time algorithm $A$ with access to a SAT oracle such that, on each input $x \in \{0, 1\}^*$ representing a propositional formula $\varphi$:

- $A$ makes at most $O(\log |x|)$ queries to the oracle, and
- if $\varphi$ is satisfiable, $A$ outputs a truth assignment $\alpha$ that satisfies $\varphi$,

unless $P = NP$.

Exercise 3 (3pt). Define:

$$P_{/\log} = \bigcup_{c,d \in \mathbb{N}} \text{DTIME}(n^c)/(d \log n).$$

That is, $P_{/\log}$ is the class of all languages that can be decided in polynomial time with $O(\log n)$ bits of advice. Prove that SAT $\not\in P_{/log}$, unless $P = NP$.

- **Hint**: iterate over all possible advice strings of length $O(\log n)$.
- **Hint**: you may assume that for any string $x$ that represents a propositional formula $\varphi$ and any truth assignment $\alpha$ to (some of) the variables of $\varphi$, one can in polynomial-time encode the formula $\varphi[\alpha]$ as a string $x'$ that is of the same length as $x$—where $\varphi[\alpha]$ is obtained from $\varphi$ by instantiating each variable $z$ in the domain of $\alpha$ by $\alpha(z)$.

Exercise 4 (2pt). Let $P_{/1}$ be the class of languages that can be decided in polynomial time with a single bit of advice. Prove that $P \not\subseteq P_{/1}$.

- **Hint**: use the following undecidable language UHALT:

$$\text{UHALT} = \{ 1^n \mid n's \text{ binary expansion encodes a pair } \langle M, x \rangle \text{ such that } M \text{ is a Turing machine that halts on input } x \}.$$