

General Instructions: Please write concisely, but rigorously. Non-rigorous solutions won't be graded. For each problem, only "nearly flawless" solutions earn 2 points. Solutions that contain the key insights but are flawed in execution earn only 1 point. Solutions that are correct but needlessly long (usually this means over 1.5 pages long) will earn only 1 point. The purpose of this strict grading scheme is to dissuade you from writing up half-baked ideas in the hope of getting "some" credit. You are budding researchers: your writing should reflect that.

Honor Principle: You are allowed to discuss the problems and exchange solution ideas, *but not complete solutions*, with your classmates. But when you write up any solutions for submission, you must work alone. You may refer to any textbook you like, including online ones. However, you may not refer to published or online solutions to the specific problems on the homework, if you intend to turn it in for credit. *If in doubt, ask the professor for clarification!*

1. For this problem, first read up the discussion about coNP in Sipser (or any other textbook).

For a complexity class \mathcal{C} , define two new complexity classes " $\exists\mathcal{C}$ " and " $\forall\mathcal{C}$ " as follows.

$$\begin{aligned}\exists\mathcal{C} &= \{ \{x \in \Sigma^* : \exists y \in \Sigma^* \text{ with } |y| = \text{poly}(|x|) \text{ such that } \langle x, y \rangle \in L_0\} : L_0 \in \mathcal{C} \} \\ \forall\mathcal{C} &= \{ \{x \in \Sigma^* : \forall y \in \Sigma^* \text{ with } |y| = \text{poly}(|x|) \text{ we have } \langle x, y \rangle \in L_0\} : L_0 \in \mathcal{C} \} \end{aligned}$$

The notation $|y| = \text{poly}(|x|)$ means that there is some fixed polynomial p such that $|y| = O(p(|x|))$.

Prove, rigorously, that $\exists\text{P} = \text{NP}$ and $\forall\text{P} = \text{coNP}$. Use only the basic definitions, where NP is defined using NDTMs and coNP is defined as $\{L \subseteq \Sigma^* : \bar{L} \in \text{NP}\}$ (as in Sipser).

This problem is simply asking you to write out a rigorous version of ideas we have already discussed in class, when we talked about the verifier-prover view of NP. In your proofs, make sure you precisely defined appropriate languages L_0 used in the definitions above. [2 points]

2. Define the following two complexity classes

$$\begin{aligned}\text{EXPTIME} &= \bigcup_{i=1}^{\infty} \text{DTIME}(2^{n^i}) \\ \text{NEXPTIME} &= \bigcup_{i=1}^{\infty} \text{NTIME}(2^{n^i}). \end{aligned}$$

Prove that $\text{P} = \text{NP}$ implies $\text{EXPTIME} = \text{NEXPTIME}$. The key trick is to "pad" an input with a lot of extra symbols. [2 points]

3. (Solve this one without looking at a textbook!) Prove that $\text{L} \subseteq \text{NL} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PSPACE} = \text{NPSPACE}$. Give concise reasons for each step; the whole thing should not take more than half a page or so. [2 points]

4. For this problem, assume that Boolean formulas are encoded as strings over the alphabet $\{0, 1, \vee, \wedge, \neg, (,)\}$, fully parenthesized to resolve ambiguities. Note that the negation operator (\neg) has higher priority than the other two. The variables in a formula ϕ are represented as binary substrings of ϕ with no leading zeros. For instance, the formula

$$((x_1 \wedge \neg x_2) \vee \neg x_1 \vee (x_3 \wedge \neg x_1)) \wedge \neg x_4 \wedge (\neg x_3 \vee x_5)$$

is represented as the string

$$((1 \wedge \neg 10) \vee \neg 1 \vee (11 \wedge \neg 1)) \wedge \neg 100 \wedge (\neg 11 \vee 101).$$

Define $\text{SATISFIES} = \{\langle \phi, \alpha \rangle : \phi \text{ is a Boolean formula and the assignment } \alpha \text{ satisfies } \phi\}$. Our proof that $\text{SAT} \in \text{NP}$ boiled down to showing that $\text{SATISFIES} \in \text{P}$. Indeed, we can say that

$$\text{SAT} = \{\langle \phi \rangle : \exists \alpha \text{ such that } \langle \phi, \alpha \rangle \in \text{SATISFIES}\}.$$

Anyhow, as we were saying, we easily have $\text{SATISFIES} \in \text{P}$. Prove the stronger result that $\text{SATISFIES} \in \text{L}$.

This problem is all about careful implementation, so take care to specify exactly how you use the work tape of your TM. Some naïve implementations end up requiring $\Omega(\log^2 n)$ space. [2 points]