

Problem Set #3

Prof. Michael A. Forbes

Due: Sat., 2026-10-10 12pm

All problems are of equal value.

1. (Sipser #8.17) Let A be the language of properly nested parentheses. For example, $(())$ and $(())(())()$ in A , but $)()$ is not. Show that A is in L .
2. (Sipser #9.19) Define the unique-sat problem to be $USAT = \{\langle \varphi \rangle : \varphi \text{ is a boolean formula that has a unique satisfying assignment}\}$. Show that $USAT \in P^{SAT}$.
3. (Arora-Barak Problem 6.3) Describe a decidable language in $P/poly$ that is not in P .
4. (Sipser #9.13, #9.14): Consider the function $pad : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \#^*$ that is defined as follows. Let $pad(s, \ell) = s\#^j$, where $j = \max(0, \ell - m)$ and m is the length of s . Thus $pad(s, \ell)$ simply adds enough copies of the new symbol $\#$ to the end of s so that the length of the result is at least ℓ . For any language A and function $f : \mathbb{N} \rightarrow \mathbb{N}$ define the language $pad(A, f(m))$ as

$$pad(A, f(m)) = \{pad(s, f(m)) : s \in A, m = |s|\}$$

- (a) Prove that, if $A \in TIME(n^6)$, then $pad(A, n^2) \in TIME(n^3)$.
 - (b) Recall that $EXP = TIME(2^{\text{poly}(n)})$, and define $NEXP = NTIME(2^{\text{poly}(n)})$. Prove that, if $EXP \neq NEXP$, then $P \neq NP$.
5. (Sipser #9.16) Prove that $TQBF \notin SPACE(n^{1/3})$.
Hint: Argue that $TQBF$ is $PSPACE$ -complete with respect to logspace-reductions.
6. (Sipser #9.24, #9.25)
 - (a) Define the function $majority_n : \{0, 1\}^n \rightarrow \{0, 1\}$ as $majority_n(x_1, \dots, x_n) = 1$ iff $\sum_i x_i \geq n/2$. Thus the $majority_n$ function returns the majority vote of the inputs. Directly show that the $majority_n$ function can be computed by size $O(n^2)$ size circuits.
Hint: Use dynamic programming.
 - (b) Recall that you may consider circuits that output strings over $\{0, 1\}$ by designating several output gates. Let $add_n : \{0, 1\}^{2n} \rightarrow \{0, 1\}^{n+1}$ take the sum of two n -bit binary integers and produce the $n + 1$ bit result. Show that you can compute the add_n function with $O(n)$ size circuits.
 - (c) By recursively dividing the number of inputs in half, and using part (b), show that the $majority_n$ function can be computed by size $O(n \log n)$ size circuits.