NAME: ____________________

CSE 431
Complexity Theory
Midterm Exam

Anup Rao

DIRECTIONS:

- Answer the problems on the exam paper.
- If you need extra space use the back of a page
- You have 80 minutes to complete the exam.
- Good Luck!

1	/50
2	/15
3	/15
Total	/90

1. (50 points, 5 each) For each of the following statements, say whether they are true, false or open according to our current state of knowledge. Briefly justify your answers. Don't worry too much about the difference between open and true or false. Just discuss what you know that is relevant to each statement.

   (a) **NL** is contained in **PSPACE**.

   True. **NL** is even contained in $\mathbf{L}^2$ by Savitch's algorithm.

   (b) If f, g are in P, then $f \leq_p g$.

   True. The algorithms for f, g can be used to make reductions between the problems.

   (c) $\mathbf{L} \neq \mathbf{EXP}$.

   True. $\mathbf{L} \subseteq \mathbf{P} \neq \mathbf{EXP}$, where the first relationship we discussed in class, and the second is because of the time-hierarchy theorem.

   (d) There is a polynomial time algorithm for finding a 3-coloring of a graph.

   Open. This is true if and only if $\mathbf{P} = \mathbf{NP}$.

   (e) If $\mathbf{P} \neq \mathbf{NP}$, then for every oracle A, $\mathbf{P}^A = \mathbf{NP}^A$.

   False. We have shown that there is an oracle A for which $\mathbf{P}^A \neq \mathbf{NP}^A$.

   (f) Every function $f : \{0,1\}^n \to \{0,1\}$ can be computed by a Boolean circuit of size $O(2^n/n)$.

   True. We showed this in class.

   (g) If $n > 10$, there is a binary string $x \in \{0,1\}^n$ whose Kolmogorov complexity is at least $n/2$.

   True. The number strings with complexity at most $n/2$ is at most the number of programs of that length, which is $2^{n/2+1}$. But hte number of binary strings of length n is much bigger: 2^n.

   (h) If a Turing machine M has space complexity $s(n) = O(\log n)$, then $M(x)$ must halt for every x.

   False. There can be a machine that has an infinite loop but does not use any space.

   (i) Every NP-hard function is computable by a Turing machine.

   False. We showed this on the homework.

   (j) **PSPACE** = **NPSPACE**

   True. This follows from Savitch's algorithm.

2. (15 points) Recall that we defined the Halting problem by

$$\mathsf{HALT}(\alpha, x) = \begin{cases} 1 & \text{If } M_\alpha(x) \text{ halts,} \\ 0 & \text{otherwise.} \end{cases}$$

Give an example of a function that cannot be computed by any Turing machine with oracle access to HALT. Prove that your function cannot be computed by such machines. HINT: Use a diagonalization argument!

Solution:

Consider the function:

$$f(\beta) = \begin{cases} 1 & \text{If } M_\beta^{\mathsf{HALT}} \text{ halts with output 0,} \\ 0 & \text{otherwise.} \end{cases}$$

Here M_β^{HALT} denotes the machine with oracle access to HALT and code β. I claim that this function cannot be computed by any machine with oracle access to HALT.

Suppose γ was the code of a machine that computes f. Then if $M_\gamma^{\mathsf{HALT}}(\gamma) = 0$, we see that $f(\gamma) = 1$, and conversely if $M_\gamma^{\mathsf{HALT}}(\gamma) = 1$, $f(\gamma) = 0$. This is a contradiction.

3. Prove that for large n, the set of functions $f : \{0,1\}^n \to \{0,1\}$ that can be computed with circuits of size n^5 is strictly smaller than the set of functions that can be computed by circuits of size $2^{n/100}$. HINT: Consider the space of functions that depend only on the first ℓ bits of the input, for some carefully chosen parameter ℓ.

   Solution: Let $\ell = \sqrt{n}$. Consider all of the function g that depend on the first ℓ bits.

   By what we have seen in class, there is such a function that cannot be computed by circuits of size $2^\ell/(3\ell) = 2^{\sqrt{n}-O(\log n)}$, which means there certainly cannot be a circuit of size $n^5 = 2^{5\log n}$ that computes this function g, for n large.

   On the other hand, every such function can be computed by a circuit of size $O(2^\ell) = O(2^{\sqrt{n}})$, which is less than $2^{n/100}$ for n large.