NAME: ____________________________

CSE 431
Complexity Theory
Sample 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) $\mathsf{ISET} \leq_p \mathsf{3SAT}$ (here ISET is the independent set problem).
   True. Since $\mathsf{ISET} \in \mathbf{NP}$ and $\mathsf{3SAT}$ is NP-complete.

   (b) $\mathbf{P}^{\mathsf{3SAT}} = \mathbf{NP}$.
   Open. $\mathbf{P}^{\mathsf{3SAT}}$ contains $\mathbf{NP}$, but it may also contain other problems.

   (c) Either $\mathbf{P} = \mathbf{NP}$ or $\mathbf{NP} = \mathbf{EXP}$.
   False/Open. It could be that $\mathbf{NP}$ is not equal to $\mathbf{P}$ but also not equal to $\mathbf{EXP}$.

   (d) $\mathbf{L} \subseteq \mathbf{PSPACE}$.
   True. Everything that can be computed in space $O(\log n)$ can also be computed in polynomial space.

(e) If $\mathbf{P} = \mathbf{PSPACE}$, then $\mathbf{NL}$ must be strictly smaller than $\mathbf{NP}$.

True. $\mathbf{NL}$ is contained in space $O(\log^2 n)$ by Savitch's algorithm. By the space hierarchy theorem, $\mathbf{NL}$ is strictly contained in $\mathbf{PSPACE}$, and if $\mathbf{P} = \mathbf{PSPACE}$, then $\mathbf{NP}$ (which contains $\mathbf{P}$) must be strictly larger than $\mathbf{NL}$.

(f) If $\mathsf{3SAT} \in \mathsf{DTIME}(2^{\log^2 n})$, then $\mathbf{NP} \neq \mathbf{EXP}$.

True. Since $\mathsf{3SAT}$ is NP-complete, this would imply that every problem in $\mathbf{NP}$ is in $\mathsf{DTIME}(2^{\log^2 n})$, but this class is strictly smaller than $\mathbf{EXP}$ by the time hierarchy theorem.

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

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

(h) If $f \in P$, then f can be computed with polynomial sized circuits.

True. We showed this in class.

(i) If 3SAT does not have polynomial sized circuits, then $\mathbf{P} \neq \mathbf{NP}$.

True. This means $3\mathsf{SAT} \notin \mathbf{P}$.

(j) There is a function $f : \{0,1\}^* \to \{0,1\}$ that cannot be computed by any Turing machine, yet for every n, $f(x)$ can be computed by a branching program of width n^{10} on inputs of length n.

True. You can obtain such a function by padding the Halting function, just as we did on the homework to show that there is a function that cannot be computed by Turing machines but can be computed by polynomial sized circuits.

2. (15 points) Give an example of a function that is computable in exponential space (space $2^{n^{O(1)}}$) yet does not have polynomial sized circuits. HINT: The space is enough to enumerate over all circuits of exponential size.

   Solution:

   Consider the function $f(x)$ defined as follows:

   On input x, enumerate over all functions $g : \{0,1\}^{|x|} \to \{0,1\}$, and find the one that cannot be computed by a circuit of size $2^{|x|/100}$. Output $g(x)$.

   This function can be computed in exponential space, since we can enumerate over all functions g by writing down their truth tables, and we can enumerate over all circuits of size $2^{|x|/100}$ within that budget for space. However, by construction this function can only be computed by circuits of exponential size.

3. (15 points) Prove that for n large enough, there is a function that can be computed by a circuit of depth $n/2$, but cannot be computed by a circuit of depth $n/3$.

   Solution: Every circuit of depth $n/3$ has size at most $2^{n/3+1}$, since the worst case is that the circuit is a full binary tree. In class we discussed that every function that depends on k variables can be computed by a 3SAT formula with at most 2^k clauses: this gives a circuit of depth at most $k+3$ for computing every such function. Now, let $k = n/2 - 3$. Then by what we showed in class, there is a function that depends on the first k input bits that requires circuit size at least $2^k/(3k)$, which is greater than $2^{n/3}$ for n large, and yet this function can be computed in depth $n/2$ by the above argument.