Exam 3

Read this page and fill in your name, pledge, and email ID now.
Do not open past this page until instructed to do so.

Name: ______________________________

UVA Email ID: ______________________________

For this exam, you must **work alone**. You are not permitted to obtain help from people other than asking clarifying questions of the course staff. You are not permitted to provide help to others taking the exam. **You may not use any resources other than your brain and body, the one page of notes you prepared, and a simple writing implement like a pen or pencil.**

Sign below to indicate that you understand these expectations and can be trusted to behave honorably:

Signed: ____

As in previous exams, our goal is to design exams that do not incentivize the intellectual dishonesty that is typically incentivized by school assignments and that you are all experts at, as demonstrated by your ability to achieve the level of success needed in high school to be admitted to the University. Hence, please keep in mind that the exam will be graded in a way that will not reward intentionally obfuscated or deceptive answers — if you do not know how to solve a problem, or get stuck at a step in a proof, it is much better to state that clearly and explain what you know that might be relevant or useful towards solving the problem, than to fabricate an answer that you know is wrong.

Although fairly generous partial credit will be awarded for answers that state that you do not know how to solve the asked problem and either solve an easier one or show something you can do that is related to the given problem, answers that we believe are deliberately deceptive will receive negative scores (worse than that 0 that a blank answer receives for any question).

The exam has 7 questions, each of which awards a good answer with 10 points (you can also get up to **30 points for filling in the three blanks** on this cover page well enough so we can read your name and id), so an exam with all good answers would be worth 100 points. For each question, we either identify the length an excellent answer should have, or else provide ample space to hold an excellent answer. If you need more space, you can use the backs of pages, but include clear markings and arrows to indicate the answer that should be graded. We will assume anything not inside an answer box or clearly marked from one, is your scratch work that should not be considered in scoring your answers.

Useful Definitions

Recall these definitions from class. We provide them here, and you can (carefully) rip this page out of the exam if it is helpful to you. These are all definitions you have seen before, and nothing should be surprising in them.

Definition 1 (O) A function $f : \mathbb{N} \to \mathbb{R}$ is in the set $O(g(n))$, defined for any function $g : \mathbb{N} \to \mathbb{R}$ iff there exist two constants $c \in \mathbb{R}^+, n_0 \in \mathbb{N}$ such that: $\forall n > n_0 . f(n) \leq cg(n)$.

Definition 2 (Ω) A function $f : \mathbb{N} \to \mathbb{R}$ is in the set $\Omega(g(n))$, defined for any function $g : \mathbb{N} \to \mathbb{R}$ iff there exist two constants $c \in \mathbb{R}^+, n_0 \in \mathbb{N}$ such that: $\forall n > n_0 . f(n) \geq cg(n)$.

Definition 3 (Θ) A function $f : \mathbb{N} \to \mathbb{R}$ is in the set $\Omega(g(n))$, defined for any function $g : \mathbb{N} \to \mathbb{R}$ iff $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$

Definition 4 (Busy Beaver Problem) For any $n \in \mathbb{N}$, define $BB_2(n)$ as the maximum number of steps for which a Turing Machine with n states and 2 symbols can execute and halt, starting from a blank tape.

Definition 5 (CNF) We say that a formula is in conjunctive normal form (CNF for short) if it is an AND of ORs of variables or their negations. E.g. $(x_7 \vee \overline{x_{22}} \vee x_{15}) \wedge (x_{37} \vee x_{22} \vee \overline{x_7})$ is in CNF. We say that it is k-CNF if there are exactly k variables per clause (group of variables combined with OR).

Definition 6 (DNF) We say that a formula is in disjunctive normal form (DNF for short) if it is an OR of ANDs of variables or their negations. E.g. $(x_7 \wedge \overline{x_{22}} \wedge x_{15}) \vee (x_{37} \wedge x_{22} \wedge \overline{x_7})$ is in DNF. We say that it is k-DNF if there are exactly k variables per clause (group of variables combined with AND).

True, False, or Unknown

1. For each of the following, circle one of the choices to indicate whether the statement is known to be *True*, is known to be *False*, or *Unknown* if its validity depends on something that is either currently unknown or not specified in the question.

Then, write a short justification to support your answer. When your answer is *Unknown*, your answer should make it clear what unknown the validity of the statement depends on (for example, that it is equivalent to a statement whose truth is currently unknown to anyone).

(a) The function, $XOR : \{0, 1\}^* \rightarrow \{0, 1\}$, which outputs the logical exclusive or of all the input bits, is in class P.

Circle one:

True *False* *Unknown*

Justification ($\leq$ 5 words):

(b) The function, XOR (from the previous question), is in class NP.

Circle one:

True *False* *Unknown*

Justification (3 symbols):

(c) If a function A in NP has an exponential lower bound (e.g., requires $\Omega(2^n)$ time to compute), then no function in NP-Complete can be computed in polynomial time.

Circle one:

True *False* *Unknown*

Justification ($\leq$ 3 sentences):

Proving Uncomputability

2. In this question, your goal is to show that the function $REENTERS$ defined below is uncomputable.

> **Input:** A string w that describes a Turing Machine.
>
> **Output:** **1** if the machine described by w would re-enter its start state when executed on a tape that is initially all blank. Otherwise, **0**.

That is, a machine which computes $REENTERS$ outputs **1** when the input describes a Turing Machine which, when run on a blank tape, enters the start state as a result of some transition.

(a) Which strategy would show that $REENTERS$ is uncomputable? (Circle one, no explication needed.)

Use a machine that computes $REENTERS$ to compute $HALTS$.	Use a machine that computes $HALTS$ to compute $REENTERS$.

(b) Employ the strategy you chose in the previous question to show that $REENTERS$ is uncomputable.

Complexity Classes

3. For some function $F : \{0,1\}^* \rightarrow \{0,1\}$, given that each of the following is True, identify the first class that F is *guaranteed* to belong to. We list the classes so each is a subset of the class to its right, so circle the leftmost class which F must belong to. If it is not guaranteed to belong to any class, don't circle anything. Briefly justify your choice.

(a) There is an algorithm that computes F with running time in $\Theta(n^{5.5})$.

$F \in \mathsf{TIME}_{\mathsf{TM}}(O(n^5 \log n))$ $\qquad$ $F \in \mathsf{P}$ $\qquad$ $F \in \mathsf{NP}$

Justification ($\leq$ 3 sentences):

(b) There is a function $R \in \mathsf{TIME}_{\mathsf{TM}}(O(n^{3102}))$ such that, for all $x \in \{0,1\}^n$ that is a well-formed input to 3-SAT, 3-$SAT(x) = F(R(x))$.

$F \in$NP-Complete $\qquad$ $F \in$NP-Hard $\qquad$ $F \in$ Computable

Justification ($\leq$ 4 sentences):

Always, Sometimes, Never

4. For a function $f : \{0, 1\}^{3102} \to \{0, 1\}$ that can be implemented by a NAND circuit with s gates, which of the statements that follow would be *Always True*, *Possibly True* (meaning there are some functions f for which the statement is true and others for which is it false), or *Never True* (circle one option). Give a brief statement to justify your answer.

(a) f is computable

Always True *Possibly True* *Never True*

Justification ($\leq$ 5 words):

(b) $f \in \mathsf{NP}$

Always True *Possibly True* *Never True*

Justification ($\leq$ 5 words):

(c) There is some function $g : \{0, 1\}^{3102} \to \{0, 1\}$ that can be implemented using $s + 10$ NAND gates, but cannot be implemented using s NAND gates.

Always True *Possibly True* *Never True*

Justification ($\leq$ 3 sentences):

Constant Time

5. Show whether $O(1) = \Theta(1)$. (It is left up to you to determine if the two sets are equivalent, and provide a convincing proof to support your answer.)

3-TAUT

We know that 3-SAT is NP-Complete, where 3-SAT requires determining whether there exists at least one way to assign Boolean values to each variable in a 3-CNF formula so that the formula evaluates to True.

For this question we will show 3-TAUT is NP-Hard. This problem requires determining whether *every* assignment causes a 3-DNF formula (see definitions page) to evaluate to True (i.e., no assignments will cause the formula to evaluate to False).

6. Show 3-TAUT is NP-Hard.

7. 3-TAUT is not known to belong to NP. Give an intuitive reason why it is difficult to show that 3-TAUT belongs to NP.

Oral Exam Request

If you are worried your performance on the in-class exams (including especially this one, on which a good performance can make up for difficulties on one of the first two exams), you may check fill in the minimum grade expectation box below to request an independently scheduled oral exam:

> If my course grade would be lower than ________,
> I would like to schedule an oral exam.

We will decide on (preliminary) final grades before looking at this. If you fill in a grade here, and your final grade would be lower than the grade you entered here, we may contact you to schedule an oral exam. At the exam, you will be expected to demonstrate your knowledge and understanding by explaining a few key definitions from the class and answering questions (including doing a proof). Your final grade may go either up or down as a result of the oral exam.

Optional Feedback

(Optional) This question is optional and will not affect your grade.

Do you feel your performance on this exam will fairly reflect your understanding of the course material? If not, explain why. (Feel free to provide any other comments you want on the exam, the course, or just draw a picture or your favorite complexity class.)

Score:	

End of Exam 3

