Exam 2

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 discussed in the Exam1 Review Class, our goal is to design an exam that does 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 we hope 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 **11** questions, each of which awards a good answer with 9 points (you can also get up to **6 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 105 points. For each question, there is ample space provided 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.

Asymptotic Operators

Recall these definitions of asymptotic operators from class:

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 $f(n) \in \Omega(g(n))$*

Let $f(n) = n^{0.5} \log n$ and $g(n) = n(\log(n))^2$, which of the following are true? Support your answer to each part with a convincing argument.

1. $f \in O(g)$

2. $f \in \Omega(g)$

3. $f \in \Theta(g)$

Counting Functions

Recall that $SIZE_n(s)$ is the set of all n-input, 1-output Boolean functions that can be implemented with s or fewer NAND gates.

The notation $\subsetneq$ means proper subset. If $A \subsetneq B$ it means that every element of A is an element of B, but that there is at least one element of B that is not in A.

Provide a **brief but convincing proof** for each of the statements below.

4. (Corrected!) If $2 < s \leq t$ then $SIZE_1(s) = SIZE_1(t)$. (This is corrected: the original question, with $1 < s \leq t$, is not true!)

5. $SIZE_{10}(9) \subsetneq SIZE_{10}(15)$.

6. $SIZE_{10}(3102) \subsetneq SIZE_{11}(3102)$.

Turing Machines

7. A Turing Machine's configuration contains all the information needed to describe the current status of its computation (i.e., if I paused my computation then wrote the configuration down, I could resume the computation using what I had written). List three necessary components of a Turing Machine's configuration.

8. A *Linear-Tape Turing Machine* is a variant of a Turing Machine where instead of having access to an unbounded tape, the machine can only use as many cells as the length of the machine's input (i.e., if it received a k-bit input, it is able to use k cells).

Is a Linear-Tape Turing Machine less powerful than a standard Turing Machine? Support your answer with a convincing argument.

9. Grace created a brand new programming language, Hopper, with a useful property: for any program written in this language, the compiler will warn you when there is some input which would cause the program to run forever.

Prove that there is no way to write a Hopper program that behaves as a Universal Turing Machine (i.e., that can simulate any given Turing Machine).

Halting Beavers

10. We defined the the Busy Beaver Problem as:

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.*

In class we demonstrated that $BB_2(n)$ was not computable by showing that we could use a decider for $BB_2(n)$ to decide $HALTS$, concluding that $BB_2(n)$ is "at least as hard as" $HALTS$.

Complete the proof that the two problems are "equivalent" in difficulty by showing how one could use a decider for $HALTS$ to decide $BB_2(n)$.

Beyond Turing Machines

11. We have discussed several models of computation so far this semester (e.g., *NAND-CIRC* programs, Finite State Automata, Turing Machines). Each of these models only allows for finite-length representations, and for each we have demonstrated functions not computable by that model.

Prove that *any* model of computation which only allows finite-length representations cannot compute all infinite boolean functions (i.e., all functions of the form $\{0, 1\}^* \rightarrow \{0, 1\}$).

Optional Feedback

This question is optional and will not negatively affect your grade. (In unusual circumstances, it might lead us to provide you with some additional opportunity to do so.)

Do you feel your performance on this exam will fairly reflect your understanding of the course material so far? If not, explain why. (Feel free to provide any other comments you want on the exam, the course so far, your hopes for the rest of the course here, or just draw a picture or your favorite computing machine.)

Score:	

End of Exam 2