Instructor: Rocco A. Servedio

Computer Science 4236: Introduction to Computational Complexity
Midterm exam, Spring 2023

Due 11:59pm Friday, March 10, 2023

Allowed materials include any of the readings for this course (linked to from the course web page); Rocco's notes from lecture; the responses that Rocco and the other instructors have provided on Ed Discussion; and any notes that you produced (typed or handwritten) yourself for the course. **Disallowed materials** include everything else.

You are **not allowed** to collaborate or communicate with anyone else while taking the exam; it must reflect entirely your own work.

Be *clear, precise, and succinct* in your answers. You will be graded not only on the correctness of your answers, but also on the clarity with which you express them.

The midterm should be formatted just like HW assignments; see the course web page for details.

If you have questions during the exam, please make a **private post** on Ed Discussion.

Problem 0 (2 points) Summarize the most important aspects of the Computer Science Department's academic honesty policy, as relevant to this exam, in your own words, in one sentence. As a reminder, some key points of the policy are reproduced below:

> *The Department enforces the general university and school policies on academic honesty, as described in the Bulletins of the School of Engineering and Applied Science, of Columbia College, and of the School of General Studies; in the Honor Code booklet of Barnard College; and in other related publications ...*
>
> *Unless specifically authorized by the instructor, no external aids or electronic devices are allowed in exams ...*
>
> *In general, the academic penalty for a first offense of academic dishonesty within the Department is a grade of zero on the assignment, project, or exam, or reduction of the course grade at the discretion of the instructor ...*
>
> *SEVERE ACADEMIC DISHONESTY: ... A student planning or executing with another student a cooperative subterfuge during an exam ... A student making use of unauthorized material during an exam.*

Problem 1 What is the relationship between the following pairs of complexity classes? Give brief but thorough justifications; if you say one class is contained in the other you should explain whether (and why) the containment is proper.

(a) (6 points) $NSPACE(n^3)$ and $SPACE(n^6 \log n)$

(b) (7 points) $PSPACE^{NP}$ and $TIME(2^{n^{\log n}})$

(c) (7 points) $SPACE(n^2)$ and $SPACE(f(n))$, where $f(n) = n$ for n odd and n^3 for n even.

Problem 2 Given a deterministic one-tape Turing machine M, we say that M has *rewrite complexity* $f(n)$ if on any input of length n the total number of times that M changes a symbol of its worktape is at most $f(n)$. (We think of such a Turing machine as initially having its n-character input written on the left portion of the tape, followed by an infinite sequence of "blank" characters.) We assume that the machine always moves its head left or right in every time step; when it makes a move that does not involve changing a symbol, we say that that is a *writeless move*.

(a) (6 points) Let M be a 1-tape TM with s internal states of its finite control. Argue that if M makes a sequence of writeless moves that bring it more than s cells beyond the rightmost non-blank character on its tape, then it will loop forever.

(b) (6 points) Let M be as in part (a) and moreover have rewrite complexity $f(n)$. Argue that such a machine, if it does not run forever, can only ever have non-blank contents in at most the first $n + sf(n)$ tape cells on any input of length n. (Note that this implies that if M does not run forever, then it can only ever have its head in the first $n + sf(n) + s$ cells of the tape.)

(c) (8 points) Let $f(n)$ be any computable function. Show that there is some decidable language L that is not decided by any one-tape Turing machine with rewrite complexity $f(n)$. (This is the same theorem which we proved in class except with "REWRITE" in place of "TIME.").

Problem 3 (20 points) Recall that a complexity class $\mathcal{C}$ (which is a set of languages) is *closed under complement* if for every language L, we have that $L \in \mathcal{C}$ implies $\overline{L} \in \mathcal{C}$. (Recall that for a language $\bar{L} \subseteq \Sigma^*$, the complement language $\overline{L}$ is $\Sigma^* \setminus L$, the set of all strings that are not in L.)

True or false: There is an oracle $\mathcal{A}$ such that NP^A is not closed under complement. Justify your answer. (If your answer refers to a proof from class, it is okay to only explain what needs to be changed rather than recapitulating the entire proof.)

Problem 4 (20 points) Suppose there exists a function f mapping the integers $\{1, \ldots, 2^k\}$ onto the integers $\{1, \ldots, 2^k\}$ such that when these integers are represented in binary,

- f is computable in polynomial time, but
- f^{-1} is not computable in polynomial time.

Let L be the language $\{(x, y) : f^{-1}(x) < y\}$. Show that this would imply that

$$L \in (\mathrm{NP} \cap \text{co-NP}) \setminus \mathrm{P}$$

i.e. that L is in NP and in co-NP but is not in P.

Problem 5 (20 points) Describe (informally but precisely) a Turing machine that has the following properties:

- On every input string x of length n, it uses $O(\log \log n)$ space;
- For infinitely many values of n, on every input string of length n it uses $\Omega(\log \log n)$ space.

Hint: The following fact from number theory may be helpful: the product of the primes between 2 and k (inclusive) is at least 2^k.