CS Theory (Spring '24)	Assigned: April 15, 2024
Homework 6	
Instructor: *Josh Alman*	Due: April 24, 2024 at noon

Please note the due time of noon! We will start grading that afternoon.

Collaboration on homework is allowed. However, you must write up solutions by yourself and understand everything that you hand in. You may also consult other reference materials, but you may not seek out answers from other sources.

Write the solution to each problem on a separate page. This includes writing each part of each problem on a separate page. Do not write your name or any other identifying information on any page of the submission; we are using anonymous grading in GradeScope.

Be sure to answer the final question listing your collaborators and other sources used. See the course webpage for more details.

There are 120 total possible points, and also bonus questions worth an additional 40 points.

1 Primeish (45 points; each part 15 points)

For positive integers n, m with $m \leq n$, we say that n is "m-primeish" if there is no positive integer k with $1 < k < m$ such that n is divisible by k.

Define the two languages:

$$PRIME = \{\langle a \rangle \mid \ a \text{ is a positive integer which is prime.}\}$$

$$PRIMEISH = \{\langle a, b \rangle \mid \ a, b \text{ are positive integers with } b \leq a, \text{ and } a \text{ is } b\text{-primeish.}\}$$

Note that $PRIME$ is a special case of $PRIMEISH$ when you set $b = a$. It's known that $PRIME \in P$ (and you may use this fact, without proving it, when solving this problem), although the algorithm achieving this is very complicated. By contrast, it's unknown whether $PRIMEISH$ is in P or NP-complete. In this problem, we'll prove that if $PRIMEISH$ is NP-complete, then a surprising consequence follows.

(a) Prove that $\overline{PRIMEISH} \in NP$.

(b) Prove that $PRIMEISH \in NP$. (You may want to use the fact that $PRIME \in P$ for this.)

(c) Use parts a and b to prove that if $PRIMEISH$ is NP-complete, then NP is closed under complement. (Recall that "NP is closed under complement" means that, for any language L, if $L \in NP$ then $\overline{L} \in NP$.)

In all problems throughout this homework, if you're asked to prove a language is in NP, be sure to clearly explain what the witness for your verifier is and why your verifier runs in polynomial time.

2 Visionary (13 points part a, 20 points part b)

For an undirected graph G with vertex set V and edge set E, we say a subset $S \subseteq V$ is "visionary" if every vertex $v \in V$ is either in S or adjacent to a vertex in S (or both). Define the language

$$VIS := \{\langle G, k\rangle \mid G \text{ is an undirected graph, } k \text{ is an integer, and } G \text{ has a visionary set } S \text{ of size } |S| \le k\}$$

In this problem, we'll show that VIS is NP-complete.

(a) Prove that $VIS \in NP$.

(b) Prove that $3SAT \le_P VIS$. (You may want to read Theorem 7.44 starting on page 312 in the textbook, which proves that VERTEX-COVER is NP-complete, and use ideas similar to that proof.)

3 Emptyness (40 points; each part 20 points)

Recall from class the languages

$$E_{DFA} := \{\langle D\rangle \mid D \text{ is a DFA that does not accept any strings}\},$$

$$E_{NFA} := \{\langle N\rangle \mid N \text{ is an NFA that does not accept any strings}\}.$$

(a) Prove that $E_{DFA} \in P$.

(b) Prove either one of the two following claims:

- $E_{NFA} \in P$.
- E_{NFA} is NP-complete.

(c) (20 BONUS points) Prove or disprove the other claim from part b (that you didn't already prove).

4 BONUS: Time Lower Bounds from Communication Complexity (20 BONUS points; each part 10 BONUS points)

Recall from class the language PAL over $\Sigma = \{0, 1\}$ defined by

$$PAL := \{w \in \{0, 1\}^* \mid w \text{ is a palindrome}\}.$$

We observed in class that there is a Turing Machine that decides PAL in $O(n^2)$ time. In this problem, we'll prove that no Turing Machine decides PAL in $o(n^2)$ time. (As always, when we say "Turing Machine", we're referring to single-tape Turing Machines.)

Actually, we'll prove a version of this for a slightly restricted model of computation: a "#-limited Turing Machine" is a Turing Machine which is not allowed to write over any #s on its tape. (A typical Turing Machine can write on any cell of its tape; a #-limited Turing Machine cannot write on cells that contain #s.) The same proof idea that we'll use here can work for regular Turing Machines too, but it gets messier.

Recall that for a language L over $\{0,1\}$, there is an associated communication problem $f_L : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$, where $f_L(x,y) = 1$ if $xy \in L$ and $f_L(x,y) = 0$ if $xy \notin L$. (Also recall that we assume, in communication problems, that the input x to Alice and the input y to Bob have the same length.)

For language L over $\{0,1\}$, we will also use an associated language $L^{\#}$ over $\{0,1,\#\}$ defined by

$$L^{\#} := \{x\#^{2n}y \mid n \text{ is a positive integer, } x,y \in \{0,1\}^n \text{ and } xy \in L\}.$$

(a) (10 BONUS points) Prove that, for any language L over $\{0,1\}$, if $L^{\#}$ is decided by a #-limited Turing Machine which runs in time $t(n)$, then the communication complexity of f_L is at most $O(t(n)/n)$.

(b) (10 BONUS points) Use part a to prove that there is no #-limited Turing Machine which decides $PAL^{\#}$ in time $o(n^2)$.

5 Collaborators and References and Rules (2 points)

Do all three of the following:

- List who you collaborated with on each problem, including any TAs or students you discussed the problems with in office hours.
- List any reference materials consulted other than the lectures and textbook for our class.
- Follow the formatting rules listed at the beginning of the assignment!