Problem Set 1

- Due date: **18 Feb, 2024** (released on 25 Jan, 2024; last addition on 6 Feb, 2024).
- The points for each problem is indicated on the side.
- More problems will be released gradually as we cover more material in class. At the moment, the total for this set is **65** points.
- The problem set has a fair number of questions so please do not wait until close to the deadline to start on them. Try and do one question every couple of days.
- Turn in your problem sets electronically (PDF; either LaTeXed or scanned etc.) on Piazza via private post.
- Collaboration with other students taking this course is encouraged, but collaboration with others is not allowed. Irrespective of this, all writeups must be done individually and must include names of all collaborators (if any).
- Referring to sources other than the text book and class notes is STRONGLY DISCOURAGED. But if you do use an external source (eg.,other text books, lecture notes, or any material available online), ACKNOWLEDGE all your sources (including collaborators) in your writeup. This will not affect your grades. However, not acknowledging will be treated as a serious case of academic dishonesty.
- Be clear in your writing.

---

1. **[Classification of problems]** (5)

   For each of these problems, mention (with justification) if they are in P, or not (known to be) in P, or in NP, or in coNP, or is NP-hard, or is coNP-hard etc.

   (a) Factoring $= \left\{(1^n, 1^k) \ : \ n \text{ has a prime factor less than } k\right\}$.

   (b) Contradiction $= \{\langle\varphi\rangle \ : \ \langle\varphi\rangle$ encodes a formula that is false for every assignment.$\}$ (For example, $\langle x_1 \wedge \neg x_1\rangle$ is in the language.)

   (c)

   $$\text{EquivalentFormulas} = \left\{ (\langle\varphi_1\rangle, \langle\varphi_2\rangle) \ : \ \begin{array}{c} \langle\varphi_1\rangle, \langle\varphi_2\rangle \text{ encode two formulas such that} \\ \text{for all } x \text{ we have } \varphi_1(x) = \varphi_2(x). \end{array} \right\}$$

2. **[Properties of reductions]** (5)

   We'll use $L_1 \leq_m^{\text{poly}} L_2$ to denote that there is a polynomial-time many-one reduction from L_1 to L_2.

   Answer each of the questions with True/False with brief justifications.

   (a) If $L_1 \leq_m^{\text{poly}} L_2$, then $\overline{L_2} \leq_m^{\text{poly}} L_1$.

   (b) If $L_1 \leq_m^{\text{poly}} L_2$, then $\overline{L_1} \leq_m^{\text{poly}} L_2$.

   (c) If L is NP-hard, then $\overline{L}$ is coNP-hard.

   (d) If $L_1 \leq_m^{\text{poly}} L_2$ and $L_2 \leq_m^{\text{poly}} L_3$, then $L_1 \leq_m^{\text{poly}} L_3$.

   (e) If $L_1 \leq_m^{\text{lin}} L_2$ and $L_2 \leq_m^{\text{lin}} L_3$, then $L_1 \leq_m^{\text{lin}} L_3$. (Here, $\leq_m^{\text{lin}}$ refers to linear-time many-one reductions).

(f) If $L_1 \leq_m^{\text{quad}} L_2$ and $L_2 \leq_m^{\text{quad}} L_3$, then $L_1 \leq_m^{\text{quad}} L_3$. (Here, $\leq_m^{\text{quad}}$ refers to quadratic-time many-one reductions).

3. **[Operations on languages]** (10)

(a) If L_1, L_2 are two languages in NP, show that the languages $L_1 \cap L_2$ and $L_1 \cup L_2$ are in NP as well.

(b) For any three languages L_1, L_2, L_3,

$$\text{Maj}(L_1, L_2, L_3) = \{x \ : \ x \text{ is in at least two of the } L_i\text{'s}\} .$$

Show that, if $L_1, L_2, L_3 \in \text{NP}$, then the language $\text{Maj}(L_1, L_2, L_3)$ is also in NP.

(c) For two languages L_1, L_2, let $L_1 \oplus L_2 = \{x \in \Sigma^* \ : \ x \text{ is in exactly one of } L_1, L_2\}$. If $L_1, L_2 \in \text{NP} \cap \text{coNP}$, show that $L_1 \oplus L_2 \in \text{NP} \cap \text{coNP}$ as well.

4. **[Not-all-equal-SAT]** (5 + 3 + 2)

The 'not-all-equal' function $\text{NAE}(x_1, \ldots, x_k)$ is the Boolean function that is true when not all the values of $x_1, \ldots, x_k$ are equal (that is, it is false only on the inputs $000\cdots0$ and $111\cdots1$).

Not-all-equal-SAT is the constrait satisfaction problem where each constraint is the above NAE-function. (Similar to how CNF-SAT had the contraints as the 'OR' function.)

(a) Consider the following purported reduction from CNF-SAT to NAE-SAT:

> Consider a new variable z. For each clause of the form, $(x_i \vee x_j \vee x_k)$ in the CNF, add the constraint $\text{NAE}(x_i, x_j, x_k, z)$ to the NAE instance.

Prove that this is a legitimate reduction from CNF-SAT to NAE-SAT.

(b) If we begin with 3 CNF-SAT, then we end up with 4 NAE-SAT in the above reduction. Give a polynomial-time many-one reduction from 4 NAE-SAT to 3 NAE-SAT.

(c) What is the 2 NAE-SAT problem? Do you know it by a different name? Is it in P?

5. **[SAT with few negations]** (1 + 4)

(a) Consider the special case of CNF-SAT where you are promised that all literals are un-negated (i.e., there are no $\neg x_i$ in the expression at all). Show that satisfiability of such instances is in P.

(b) Consider the slightly more general specualial case of CNF-SAT where you are promised that every clause has at most 1 negated variable. Prove that checking satisfiability of such instances is also in P.

[Hint: A clause of the form $x_1 \vee x_2 \vee x_3 \vee \overline{x_4}$ is equivalent to "all of x_1, x_2, x_3 being false forces x_4 to be false as well".]

6. **[Integer solutions to an equation]** (5 + 0 + 0)

Consider the following language

$$L = \left\{ (a,b) \ : \ a,b \in \mathbb{Z} \ , \quad \begin{array}{c} \text{there exists integers } x, y \\ \text{such that } x^2 + ay + b = 0 \end{array} \right\}$$

Here, the inputs a, b are provided in binary.

(a) Prove that $L \in \text{NP}$.

(b) What is your guess on whether or not this problem is NP-complete, and why do you feel so? (Giving no answer here will cost you 5 points!)

(c) Above, we were checking if a given (by providing a, b) special quadratic polynomial $P(x, y) = x^2 + ay + b$ had integer solutions. What do you think is the complexity of the problem if you generalise it to arbitrary equations? For instance, given a polynomial $P(x_1, x_2, \ldots, x_{58})$ of degree 4 polynomial (again, provided by presenting its coefficients), what do you think is the complexity of checking if it has integer solutions? (Giving no answer here will cost you 5 points!)

7. **[Non-deterministic UTM simulations]** (10)

Assume that $T : \mathbb{N} \to \mathbb{N}$ is a time-constructible function. Consider the following task — you are given a source code of a non-deterministic machine M deciding a language L running in time $T : \mathbb{N} \to \mathbb{N}$ but M uses 81234 tapes. Show that there is a different machine M' deciding the same language L with just 2 tapes with the running time $T' : \mathbb{N} \to \mathbb{N}$ of M' satisfying $T'(n) = O(T(n))$.

In words, show that we can perform tape-reduction with just a constant overhead if we have the power of non-determinism.

[Hint: In the machine M', use one of the tapes to 'interleave' the 81234 tapes of M and use the second tape to write down a guess for what the heads of M' do in the T steps.]

8. **[Improving the time-hierarchy theorem]** (15)

For this problem, you may assume that any 'reasonable-looking' function is time-constructible. And whenever you see functions like $n^2 \log^{3/4}(n)$, assume that there is an implicit ceiling to make sure this is an integer etc. (Basically don't worry about technicalities!)

In class we proved that the deterministic time hierarchy theorem that stated the following:

> Suppose $t_1, t_2 : \mathbb{N} \to \mathbb{N}$ are non-decreasing time-constructible functions with $t_1(n), t_2(n) \geq n$. If we have $t_1(n) \log t_1(n) = o(t_2(n))$, then we have $\mathsf{DTIME}(t_1) \subsetneq \mathsf{DTIME}(t_2)$.

One of your classmates asked in class if this log-factor can be made smaller. You can now blame him for this question.

(a) Let $t_1, t_2, f : \mathbb{N} \to \mathbb{N}$ be time-constructible non-decreasing functions that satisfy $t_1(n), t_2(n), f(n) \geq n$. Show that $\mathsf{DTIME}(t_1(n)) = \mathsf{DTIME}(t_2(n))$ implies

$$\mathsf{DTIME}(t_1(f(n))) = \mathsf{DTIME}(t_2(f(n))).$$

[Hint: Padding.]

(b) Show that $\mathsf{DTIME}(n^2) \subsetneq \mathsf{DTIME}(n^2 \log^{3/4}(n))$.

[Hint: You may have to use the above part multiple times.]

(c) Extend this to show that for any rational number a, ε satisfying $a > 1$ and $0 < \varepsilon < 1$, we have

$$\mathsf{DTIME}(n^a) \subsetneq \mathsf{DTIME}(n^a (\log n)^\varepsilon).$$