COMS 3261, Computer Science Theory (Fall 2023): Assignment 4
Due on Gradescope 11:59pm, Tuesday Dec 5, 2023

Instructions

- The total number of points is 52. Submit your solutions in pdf format. Late homeworks will **not** be accepted. You can discuss with TAs, the prof, and other students, but please acknowledge them at the beginning of each problem. All solutions must be written in your own words.
- Note the material for Problems 3b and 5b will be covered on Monday Nov 27. You should be able to solve the other problems based on what has been covered so far, with the exception of the definition of NP. The definition can be found in the book and is also included on the second page of this homework.

Problems

1. (5 points) Prove that the class P is closed under complement.

2. (10 points) Let FACTOR be the function that takes as input a natural number x in decimal notation, and outputs the prime factorization of x, also in decimal. For example, if the input is 13, then the output would be 13, and if the input is 12, the output would be 2,2,3. (You can output the prime factors in any order.) Prove that if $P = NP$ then there is a polynomial-time algorithm for FACTOR. Note that NP is a class of *languages* (or decision problems), so factoring is not in NP. Therefore, you should give an algorithm for FACTOR, assuming that every language in NP is solvable in polynomial-time.

3. (10 points) State whether you think the following statements are True or False and give a one sentence justification/explanation of your answer.

   (a.) For any language L, if L is in NP, then the complement of L is also in NP.

   (b.) For any NP-complete language L, if the complement of L is in P, then $P = NP$.

4. (12 points) A Boolean formula is in CNF form if it is the logical AND of a set of clauses, where each clause is the OR of a set of literals, and each literal is a variable or its negation. A CNF formula ϕ over variables $x_1, \ldots, x_n$ is *satisfiable* if there exists a Boolean assignment $\alpha \in \{0,1\}^n$ to the variables such that $\phi(\alpha)$ evaluates to true.

   (a.) Is the following formula satisfiable? Prove your answer.

   $$\phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_4) \wedge (x_4)$$

   (b.) Give a polynomial-time algorithm that takes as input a 3-CNF formula ϕ, and outputs another 3-CNF formula ϕ', such that: (i) ϕ' contains exactly 3 distinct literals per clause; (ii) the runtime of your algorithm is polynomial-time in n, where n is the number of underlying variables in ϕ; and (iii) ϕ is satisfiable if and only if ϕ' is satisfiable. (Note that the variables of ϕ' can include new variables in addition to the variables of ϕ.)

5. (15 points) The problem k-minSAT takes as input a CNF formula f over n Boolean variables $x_1, \ldots, x_n$ and a number $k \leq n$, and accepts if and only if f has a satisfying assignment with at most k 1's.

   (a.) Prove for any constant k, k-minSAT is in P. (e.g., 3-minSAT is in P.)

   (b.) Prove that k-minSAT is NP-complete.

Definition of P and NP and an Example

A language $L \subseteq \{0,1\}^*$ is in P if there is a TM M that accepts L and furthermore M halts on all inputs in polynomial-time (in the length of the input). (See Book or Lecture notes for definition of polynomial-time.)

A language $L \subseteq \{0,1\}$ is in NP if there is a nondeterministic TM N that accepts L and furthermore for all inputs and all nondeterministic computation paths, N halts in polynomial-time (in the length of the input).

An example of a language in NP is the language Clique. Recall that an input to Clique is a pair (G, k) where G is an undirected graph. The input (G, k) is in Clique iff G contains a clique of size k. To see that it is in NP, the nondeterministic algorithm on input (G, k) would first guess a subset V' of the vertices of V, such that $|V'| = k$. Then the algorithm would check whether G contains all edge between vertices in V', and if so then halt and accept, and otherwise halt and reject. Note that if G contains a k-clique then there exists a computation path (a good guess) such that G will halt and accept, and if G does not contain a k-clique then every computation path will halt and reject.