CS 535: Complexity Theory, Fall 2026

Homework 2

Due: 11:59PM, Friday, September 18, 2026.

Reminder. Homework must be typeset with LaTeX preferred. Problems 2 (all parts) and 3 are required. Problems 1 and 4 are optional.

Problem 1 ((*Optional*) Unique Path)**.** Show that the problem

$$\mathsf{UNIQUE-PATH} = \{\langle G, s, t\rangle \mid \text{directed graph } G \text{ has exactly one simple path from } s \text{ to } t\}$$

is in $\mathbf{P}$.

Problem 2 ((*Required*) $\mathbf{coNP}$)**.** Recall that the complexity class $\mathbf{coNP}$ consists of languages L such that $\overline{L} \in \mathbf{NP}$.

(a) Show that a language L is $\mathbf{NP}$-complete if and only if $\overline{L}$ is $\mathbf{coNP}$-complete. Recall that completeness for both classes is defined with respect to polynomial-time (Karp) reductions.

(b) Recall that the set difference operation on two languages A, B is defined by $A \setminus B = \{x \mid x \in A, x \notin B\}$. Is $\mathbf{NP}$ closed under the set difference operation? That is, is it the case that for every $A, B \in \mathbf{NP}$, we have $A \setminus B \in \mathbf{NP}$? Justify your answer.

(c) Define the language

$$\mathsf{FACTOR} = \{\langle m, k\rangle \mid m \text{ has a prime factor } p \text{ such that } p \leq k\}.$$

Here, the numbers m and k are written in binary. Show that if FACTOR is $\mathbf{NP}$-complete, then $\mathbf{NP} = \mathbf{coNP}$.

You can use without proof that there is a polynomial time algorithm for testing whether a number p (written in binary) is prime.

(d) Find the first error in the following "proof" that $\mathbf{NP} = \mathbf{coNP}$, and explain why it is an error: Let M be a nondeterministic polynomial-time algorithm deciding

$$\mathsf{FSAT} = \{\varphi \text{ a Boolean formula } \mid \exists x\ \varphi(x) = 1\}.$$

We design a nondeterministic polynomial-time algorithm deciding

$$\mathsf{TAUT} = \{\varphi \text{ a Boolean formula } \mid \forall x\ \varphi(x) = 1\}$$

as follows. On input φ (an instance of TAUT), evaluate $b = M(\neg\varphi)$. If $b = 0$, output 1, and if $b = 1$, output 0. This runs in nondeterministic polynomial-time as long as M does, and $\varphi \in \mathsf{FSAT}$ iff $\neg\varphi \notin \mathsf{TAUT}$, so it decides TAUT. Therefore, $\mathsf{TAUT} \in \mathbf{NP}$. Since TAUT is $\mathbf{coNP}$-complete, it follows that $\mathbf{coNP} \subseteq \mathbf{NP}$. A similar argument shows that $\mathbf{NP} \subseteq \mathbf{coNP}$, hence $\mathbf{NP} = \mathbf{coNP}$.

Problem 3 ((*Required*) Class Assignments)**.** As the principal of an elementary school, one of your jobs is to determine class assignments for the n students entering 2nd grade. The first grade teachers generated a list of "disruptive friend groups," which are subsets $G_1, G_2, \ldots, G_m \subseteq [n]$ such that the presence of any complete friend group G_i in a classroom will be...problematic.

Is it possible to partition the n students into two classrooms such that every disruptive friend group is split between the two classes? That is, do there exist disjoint sets S_1, S_2 such that $S_1 \cup S_2 = [n]$ and for every friend group G_i, we have $G_i \not\subseteq S_1$ and $G_i \not\subseteq S_2$?

Define the language PC (short for "peaceful classrooms") such that an instance $\langle n, G_1, \ldots, G_m \rangle \in \mathsf{PC}$ if and only if there exists an appropriate partition of the n students described above. Show that the language PC is $\mathbf{NP}$-complete via a poly-time reduction from $\mathsf{3SAT}$.

Problem 4 ((*Optional*) $\mathbf{NEXP}$ via Verifiers)**.** Recall that we used nondeterministic Turing machines to define the complexity class $\mathbf{NEXP}$ via

$$\mathbf{NEXP} = \bigcup_{c=1}^{\infty} \mathbf{NTIME}(n^c).$$

Modify Definition 2.1 in Arora-Barak to give an equivalent characterization of $\mathbf{NEXP}$ in terms of certificates and deterministic verifiers. Prove that your characterization is correct. Pay close attention to certificate length and verifier runtime.