International Institute of Information Technology Hyderabad

Modern Complexity Theory (CS1.405)

Assignment 2
Deadline: September 18, 2023 (Monday), 17:00 PM
Venue for Submission: CSTAR, A3-110, Vindhya Block, IIIT Hyderabad
Total Marks: 100

NOTE: It is strongly recommended that no student is allowed to copy from others.
No assignment will be taken after deadline. For ANY DOUBTs, please contact the TAs.
Write the following while submitting ONLY HARDCOPY:

Modern Complexity Theory (CS1.405)
Assignment 2
Name:
Roll No.:

1. Suppose a given problem X is **NP-complete**, which is proved through a polynomial-time reduction that maps $size - n$ instances of **SAT** to $size - n^3$ instances of the problem X. Also, suppose that some genius manages to prove **SAT** requires $\Omega(c^n)$ time, for some constant $c > 1$. What can you conclude about the time complexity of problem X? Justify.

[10]

2. An **oracle** is a language $L \subseteq \{0, 1\}^*$. An **oracle Turing machine** is the same as a normal **Turing machine**, only with the addition of a second tape, called the **oracle tape**. The cells on the **oracle tape** can contain either blanks, 0's, or 1's. **Cook Reduction** is a reduction computed by a deterministic polynomial time oracle Turing machine. **Karp-reduction** is a polynomial-time many-one reductions. Show that, if **NP** $\neq$ **P**, there exists an infinite sequence of sets $\{S_1, S_2, \cdots\}$ in **NP**$\setminus$ **P** such that S_{i+1} is **Karp-reducible** to S_i, but S_i is not **Cook-reducible** to S_{i+1}.

   - Prove that a set S is **Karp-reducible** to some set in **NP** if and only if S is in **NP**.
   - If every set in **NP** can be **Cook-reduced** to some set in **NP** $\cap$ **CoNP** then **NP** = **CoNP**.

[10 + 10 = 20]

3. If **NP** $\neq$ **CoNP**, then show that **NP** $\cap$ **CoNP**$\setminus$**P** is a subset of **NPI**, where **NPI** is the class of such problems which are not in class **P** nor **NP-Complete**.

[10]

4. Given **NEXP** = $\cup_{k \in \mathbf{N}}$**NTIME**(2^{n^k}). Show that the following problem is **NEXP-complete**: Given $< M, x, n >$, consisting of description of a **NTM M**, input x and an integer n in binary, does M have an accepting computation on x in n steps.

[10]

5. Show that for every time constructible $t : \mathbf{N} \to \mathbf{N}$, if $L \in$ **TIME**$(t(n))$, then there is an oblivious **TM** that decides L in time $O(t(n) \log t(n))$.

[10]

6. For any real numbers r_1, r_2 such that $1 \leq r_1 \leq r_2$, there is s set of strings which has nondeterministic time complexity n^{r_2} but not nondeterministic time complexity n^{r_1}.

[10]

7. Prove that if a unary language is **NP-Complete** then **P=NP**.

[10]

8. Imagine you and your group of friends are on a covert mission. You are the brain of the entire operation. One of the areas that your team has to cross, is scattered with mines. You, being the brain of the operation, have to ensure that your team moves swiftly without setting it off and alerting the enemy (also, avoid the death of anyone in the operation). So essentially, you become the $minesweeper$, like the game. You decide to mathematically model the area and stumble across the result that $MINESWEEPER \in$**NP-Complete**. Your superior doubts your capabilities and decides to replace you- but you are adamant and decide to hand him the proof to prove your capabilities. Show the mathematical model and the main result of $MINESWEEPER \in$**NP-Complete** you would show your superior.

[10]

9. The last mission was easy. This time though, you have to deal with people. For some important information, you're required to meet a spy- but this spy is a bit moody. According to to the spy's file, he's a big fan of the Philosopher's Football. What this means is that you could challenge the spy for a game and get the information. The problem is that you have never heard of this game! This gives room for your superior to doubt your capabilities AGAIN. But owing to your previous success, your superior gives you some time to figure it out. After the studying the game, you realize that the game is a bit tricky- but you figure something out anyway. It is this- the problem of determining whether a player has a move that immediately wins the game is **NP-Complete**. To put it differently- looking at a configuration of the game and determining if you can win the game in one move is **NP-Complete**. Show this proof so that you can convince your superior and complete this mission.

[10]

All the best!!!