- The homework is due on May 16, 23:59pm. Please submit your solutions to Gradescope.
- Starting from Homework 1, all homework sets allow *group submissions* up to 2 people. Please write down the names of the members *very clearly* on the first page of your solutions.
- Answer the questions in a way that is clear, correct, convincing, and concise. The level of details to aim for is that your peers in this class should be convinced by your solutions.
- You can use any statements proved during the working sessions/lectures without proofs in your solutions.
- You might notice the difficulty of the homework problems are much higher than the worksheets. *This is by design.* These problems are meant to stretch your ability and solidify your understanding of the core concepts.
- You are expected to spend a reasonable amount of time (measured in hours) working on these problems. Remember you are allowed to utilize any resources. Make sure to cite all the people/webpages/source of infomation that helped.
- Some problems are marked with a *star*; these are more challenging (and fun) extra credit problems. They are optional and do not count toward raw grades.

---

1. ***Bogus proof for P=NP.*** A boolean formula is in ***disjunctive normal form*** (or DNF) if it consists of a *disjunction* (OR) of several terms, each of which is the *conjunction* (AND) of one or more literals. For example, the formula

$$(\overline{x} \wedge y \wedge \overline{z}) \vee (y \wedge z) \vee (x \wedge \overline{y} \wedge \overline{z})$$

is in disjunctive normal form. Consider the satisfiablility problem for DNFs:

> DNFSAT
> - ***Input:*** *A boiolean formula ϕ in disjunctive normal form*
> - ***Output:*** *Is the formula ϕ satisfiable?*

(a) Describe a polynomial-time algorithm to solve DNFSAT.

(b) What is the error in the following argument that P=NP?

> We reduce the NP-hard problem 3Sat to DNFSat. Suppose we are given a boolean formula in conjunctive normal form with at most three literals per clause, and we want to know if it is satisfiable. We can use the distributive law to construct an equivalent formula in disjunctive normal form. For example,
>
> $$(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y}) = (x \wedge \overline{x}) \vee (y \wedge \overline{x}) \vee (\overline{x} \wedge \overline{x}) \vee (x \wedge \overline{y}) \vee (y \wedge \overline{y}) \vee (\overline{x} \wedge \overline{y})$$
>
> Now we can use the algorithm from part (a) to determine, in polynomial time, whether the resulting DNF formula is satisfiable. We have just solved 3Sat in polynomial time. Since 3Sat is NP-hard, we must conclude that P=NP.

2. ***Someone that compares.*** We know that CIRCUITSAT is *self-reducible*. That is, if there is an oracle that decides whether a given circuit is satisfiable, then we can find the satisfying assignment in polynomial time. Not surprisingly, oracles probably do not exist in real-life.

   One day, a monk visits your door and claims that, while they have no way to decide if a given circuit is satisfiable or not, they hold the power to decide between two given circuits, which one is more likely to have a satisfying assignment. In other words, if you hand the monk two circuits C and C':

   - the monk will correctly point out which circuit is satisfiable if exactly one of C and C' is satisfiable;
   - the monk will choose an arbitrary circuit if *both* or *neither* C and C' are satisfiable.

   Most importantly, the monk themselves do not know which case it is; since neither you nor them can check satisfiability, it is impossible to tell if the circuit the monk pointed out has a satisfying assignment or not.

   Prove that you can find a satisfying assignment for any given circuit in polynomial time, with the help of the monk.

*3. ***Unary language cannot be NP-hard.*** Prove that if any unary language $A \subseteq 1^*$ is NP-complete, then P=NP.