
Homework 8 – Due Tuesday, April 21, 2020 before 2:00PM

Reminder Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the course staff if asked. You must also identify your collaborators. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Exercises Please practice on the following exercises and solved problems in Chapter 7: 7.5, 7.7, 7.12, 7.16, 7.40. The material they cover may appear on exams.

Problems There are 3 mandatory problems.

1. **(NP)** Recall that we've seen two equivalent characterizations of the complexity class NP: One as the class of problems with polynomial-time verifiers, and the other as the class of problems which can be solved in polynomial-time on nondeterministic Turing machines. In part (a) you will show that a problem is in NP using the first characterization, and in part (b) you will use the second characterization.

- (a) **(Homework conundrum)** You are trying to plan a schedule for doing your homework next semester. You will be given n homework assignments. For each assignment a , you know the the release time r_a , the length of time ℓ_a you have to spend on it, and the deadline d_a . You would like to determine whether you can find an *uninterrupted schedule* where you are doing *exactly one* assignment at a time from the beginning to the end and satisfying all the deadlines. For simplicity, you can think of all numbers involved as positive integers. For example, if you have HW1 released at time 1, due at time 11, taking 2 time units, HW2 released at time 3, due at time 9, taking 4 time units, and HW3 released at time 2, due at time 7, taking 3 time units, you can have an uninterrupted schedule as follows: you do HW3 from 2 to 5, HW2 from 5 to 9, and HW1 from 9 to 11. We formalize this problem as a language as follows: $HC = \{\langle n, (r_1, \ell_1, d_1), \dots, (r_n, \ell_n, d_n) \rangle \mid \text{all numbers in the input are positive integers and there exists an uninterrupted schedule}\}$. Show that $HC \in \text{NP}$ by giving a **polynomial-time verifier**.

- (b) Consider the language

$$HASCERT = \{\langle M, x, 1^n, 1^k \rangle \mid \text{there exists } u \in \{0, 1\}^n \text{ such that the deterministic TM } M \text{ accepts input } \langle x, u \rangle \text{ within } k \text{ steps}\}.$$

Prove that $HASCERT$ is in NP by giving a **nondeterministic polynomial-time algorithm** for it.

- (c) If k was written in binary, would your solution to part (b) still work? Why or why not?

2. (**Search vs. decision**) You are consulting for a hospital that is looking to improve the morale of its surgery teams. A surgical team¹ consists of a surgeon, an anesthesiologist, and a surgical technician. Every year, the hospital sends you sets X, Y , and Z , with n individuals each, corresponding to the three roles. You also receive a set M of triples, where each triple corresponds to a potential surgery team where all 3 individuals can work together without fighting. An individual can appear in multiple triples. The hospital administration is asking you to find a way to arrange the largest possible number of happy surgical teams. That is, your goal is to select a largest subset of triples from M so that each individual appears in at most one triple. (There may be more than one largest subset, in which case, you are free to pick any largest subset.)

In this problem, you will prove that if $P=NP$, you can *find* a largest set of happy surgical teams in polynomial time.

- (a) Consider the following language:

$TEAM = \{ \langle n, X, Y, Z, M, k \rangle \mid |X| = |Y| = |Z| = n, \text{ each element of } M \text{ is a triple } (x, y, z) \text{ where } x \in X, y \in Y \text{ and } z \in Z, \text{ and } M \text{ contains a subset of size } k \text{ where each element appears in at most one triple} \}$.

Prove that $TEAM \in NP$.

- (b) If $P=NP$, the proof you just gave for part (a) implies that $TEAM \in P$, that is, in polynomial time you can decide whether it is possible to have k happy surgical teams. Assume you have a subroutine that does it, and use it repeatedly to *find* a largest set of happy surgical teams in polynomial time.

Hint: Similar to problem 7.40, solved in the book.

3. (**Systems of linear inequalities**) A *linear inequality* over variables x_1, \dots, x_k is an inequality of the form $c_1x_1 + \dots + c_kx_k \leq b$, where c_1, \dots, c_k and b are integers. E.g., $5x_1 - 3x_2 + x_3 \leq -1$ is a linear inequality. A *system of linear inequalities* is a set of inequalities over the same variables. Such a system *has an integer solution* if one can assign integer values to all variables in such a way that all inequalities are satisfied.

Formulate as a language LE the problem of deciding whether a given system of linear inequalities has an integer solution.

- (a) Prove that LE is in NP.

- (b) Give a polynomial time reduction from 3SAT to LE.

(*Careful:* make sure you are doing it in the direction specified.)

¹omitting several important roles for simplicity