Problem Set 9

This problem set is due **at 10:00pm** on **Wednesday, May 6, 2020**.

Please make note of the following instructions:

- This assignment, like later assignments, consists of *exercises* and *problems*. **Hand in solutions to the problems only.** However, we strongly advise that you work out the exercises for yourself, since they will help you learn the course material. You are responsible for the material they cover.
- Remember that the problem set must be submitted on Gradescope. If you haven't done so already, please signup for 6.046 Spring 2020 on Gradescope, with the entry code MNEBKP, to submit this assignment.
- We require that the solution to the problems is submitted as a PDF file, **typeset on LaTeX**, using the template available in the course materials. Each submitted solution should start with your name, the course number, the problem number, your recitation section, the date, and the names of any students with whom you collaborated.
- You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your essay should provide the following:
  1. A description of the algorithm in English and, if helpful, pseudocode.
  2. A proof (or indication) of the correctness of the algorithm.
  3. An analysis of the asymptotic running time behavior of the algorithm.
  4. Optionally, you may find it useful to include a worked example or diagram to show more precisely how your algorithm works.

EXERCISES (NOT TO BE TURNED IN)

NP Completeness

- Do Exercise 34.2-6 in CLRS on page 1066.
- Do Exercise 34.5-1 in CLRS on page 1100.
- Do Exercise 34.5-3 in CLRS on page 1101.

Problem 9-1. 3-SAT quickies [30 points] Recall that in the decision problem 3-SAT, we are given the input $C = c_1 \wedge c_2 \wedge \ldots \wedge c_n$ where each clause contains three variables or literals: $c_i = a \vee b \vee c$ for some a, b, and c from the set of variables and literals $\{v_1, \overline{v_1}, v_2, \overline{v_2}, \ldots v_m, \overline{v_m}, \text{TRUE}, \text{FALSE}\}$, and we wish to output whether there exists an assignment of TRUE, FALSE to variables $v_1, v_2 \ldots, v_m$ such that C is true.

(a) [15 points] Let MST be the decision problem, where given the input $\langle G, k\rangle$, we wish to output whether G contains a minimum spanning tree with total weight at most k. Give a reduction that is *not necessarily polynomial*[1] from 3-SAT to MST.

(b) [15 points] Given an oracle $\mathcal{A}$ that can solve the decision 3-SAT problem in $O(1)$ time, give a polynomial-time algorithm to find a satisfying assignment for an input to 3-SAT, or output "not satisfiable."

[1]Note that this is unusual as we normally require our reductions to run in polynomial time.

Problem 9-2. Scheduling Issues [30 points]

At BIT, the Bitsville Institute of Technology, many students have final exams to take at the end of the semester. For a given semester, administrators at BIT are trying to schedule n examinations in only four time slots. Administrators also have a list of classes that each student is taking. They must construct a schedule such that each class is assigned to one of the four time slots so that no student is taking two exams assigned to the same time slot. We define the decision problem SCHEDULING that corresponds to checking whether there exists a feasible exam schedule for a given list of students S and their lists of classes C. Concretely, let

$$\text{SCHEDULING} = \{\langle S, C\rangle | \langle S, C\rangle \text{ is an input with a feasible exam schedule}\}$$

(a) [5 points] Prove that SCHEDULING is in NP.

We now want to prove that SCHEDULING is NP-hard. We will do this in two parts.

(b) [10 points] Consider the 4-COLOR problem in which we aim to determine whether every node in a graph can be colored with one of four colors so that the endpoints of each edge must have different colors. Argue that SCHEDULING and graph 4-COLOR are equivalent (in terms of hardness) by providing a direct reduction from SCHEDULING to 4-COLOR and one from 4-COLOR to SCHEDULING.

(c) [15 points] Prove that 4-COLOR is NP-hard. (Note that in light of (b), this immediately proves that SCHEDULING is NP-hard too.)

Problem 9-3. Dominating Set [30 points]

Given an undirected graph $G = (V, E)$ with V vertices and E edges, let a dominating set S be a set of vertices such that every vertex is either in S or adjacent (connected via an edge) to a vertex in S. We wish to capture the problem of determining if there exists a dominating set of size k. To this end, let us define

$$\text{DOMINATING-SET} = \{\langle G, k\rangle \mid G \text{ has a dominating set of size k}\}$$

Prove that DOMINATING-SET is NP-Complete. **Hint:** Consider a reduction from VERTEX-COVER. You can assume the graph has no isolated vertices.

Problem 9-4. Feedback Form [10 points] Please fill out a feedback form about this problem set at

`https://forms.gle/nhXmRSFm3sZWhUNB9`.

It should not take more than a few minutes and will greatly help us improve teaching and material for future semesters!