Homework 6: Smartups

Due: Oct. 30, 2018 at 6:00 PM (early)
Nov. 2, 2018 at 6:00 PM (on time)
Nov. 4, 2018 at 6:00 PM (late)

This is a partner homework as usual: work on all problems together; you are responsible for everything you and your partner submit and it is an academic code violation to submit something that is not yours. Check for your partner in the Google doc that was sent out. **If you don't have a partner, please let us know.**

One of the features of good writing style is to say everything once and no more than once. When you are writing up these problems, please find a way to organize your presentation so that similar repeated parts of your argument are instead compressed into a single unit. This will make it easier for you to write and for us to read, and will sound more professional.

> Trainder was a huge success. User satisfaction was on the rise, and our daily active users increased tenfold! Unfortunately, an angry engineer caused a crash, which caused Trainder to stop in its tracks. Our CEO knew we had to pivot, but after many failed attempts, she has run out of ideas! Inspired by her lack of ideas, she decides to abandon ideas altogether, pivoting to Smartups, a startups-as-a-service company.

At Smartups, we have a standard way for an employee to prove that problem P' is NP-hard:

1. First (carefully) pick an NP-hard problem (for example, graph 3-coloring).
2. Then you need to show that the employee's problem is *at least as hard as graph 3-coloring*: show that if the employee can magically solve their problem P' (for any input), then you could use her to solve graph 3-coloring:
   (a) Consider an arbitrary instance x of the graph 3-coloring problem.
   (b) For any instance x of the graph 3-coloring problem, show how to translate x into an instance y of the employee's problem P', making sure that you describe a everything needed to define an instance of the employee's problem.
   (c) Assume that the employee solves this instance y of P'.
   (d) Then you need to show that solutions to the two problems are equivalent, that the instance x of graph 3-coloring can be solved **if and only if** the translated instance y of the employee's problem P' can be solved. You can prove this by showing:
      i. Every valid 3-coloring of x corresponds to a valid solution to the instance y of P'.
      ii. Every valid solution to the instance y of P' corresponds to a valid 3-coloring of x.

   (Parts i and ii are essentially showing that the answer to the 3-coloring instance is yes if and only if the answer to the corresponding instance of P' is yes; there are other ways of proving that "yes maps to yes and no maps to no" besides the method in i and ii.)

Note that most NP-hard problems are phrased as *decision problems*, namely "is a graph 3-colorable", but these are polynomially equivalent to the corresponding *search problems*, "find a 3-coloring of the graph", and this distinction is not going to be emphasized here.

Here are some NP-hard problems relevant to this assignment (google them if you are unfamiliar!): Independent-Set, even when every node has degree at most 3; Set-Packing.

Make sure that you clearly indicate on each problem which track you're in (either **1-credit** or **full**). **1-credit track:** do problems 1, 2, and 4.

Problem 1

(20 points total) Our CEO is hosting a networking event to help people come up with startup ideas. She has certain ideas about how her guests should interact. From a list of n potential guests, she knows which guests already know each other, and she wants to use this information to figure out who to invite to this event. In each of the following cases, determine whether there is a polynomial time algorithm for her, or whether the problem is NP-hard. State and prove the algorithm, or prove NP-hardness in each case.

1. Our CEO wants to ensure that everyone has a group to hang out with at the mixer: each guest should know at least 4 other guests. She wants to maximize the number of guests at the party.
2. She wants to ensure both that everyone has a group to hang out with *and* that everyone can meet new people: each person should (1) know at least 4 other guests; and (2) *not know* at least 4 other guests. She wants to maximize the number of guests at the event.
3. But her main goal is to help her guests "network": each guest must know as *at most 4* other guests at the event. She wants to maximize the number of guests at the event.

Hint: Does changing the number "4" to a smaller number simplify the problem and give you some helpful intuition?

Problem 2

Our networking event was a great success and we now have n startup teams eager to work with us! In order to get more VC funding our CEO wants to work with as many of these teams as possible. Each team has a schedule for when our CEO would need to show up: the 12th team might say "to train us you need to attend a session from 2:37 PM to 3:19 PM 5 days from now, AND a session from 8:08 AM to 8:51 AM 7 days from now, AND a session from 1 PM to 2 PM 10 days from now." To properly mentor each startup, our CEO cannot miss any of the sessions for a team she wants to work with. Of course, this means the our CEO cannot meet with teams that have overlapping sessions.

Given a list of meeting schedules for n startups, we want to find the maximum number of teams our CEO can meet with.

1. (10 points) Show this problem is NP-hard.
2. (10 points) Suppose that our CEO is now allowed to miss one training session, total, among all the teams. That is, our CEO's schedule may contain one overlap. Show that this problem is still NP-hard.

Problem 3

(20 points) Exhausted after weeks of meetings our CEO goes back to relax and solve some crossword puzzles. Concerned that our CEO may not be getting the relaxation she needs, you will show him that crossword puzzles are NP-hard. We consider a simplified variant of crosswords where there are no clues, and also no "black squares": a crossword solver is given a rectangular board, and a dictionary, and needs to find a way to fill the board with letters such that each row and each column is a valid word from the dictionary. (The "dictionary" is an input to the problem, and thus is not meant to be English, but just an arbitrary list of sequences of letters that you, the puzzle maker, get to choose. Remember that the dictionary is part of the input to the problem, so cannot be too large.) Show that this problem is NP-hard.

A few hints:

1. Remember, you need to show how to embed *some* NP-hard problem in the crossword problem, and in particular, your construction might only consider crosswords of a very particular form. (You get to write the dictionary.)
2. Consider a rectangular board (i.e., not square) so that in your dictionary: 1) words that can appear as a row and 2) words that can appear as a column form distinct sets.
3. For an $n \times m$ crossword, consider having the only valid length-n words be the $n + 1$ words consisting of A's and B's with *at most one B*.

Problem 4

(20 points) Now that we have so many startup teams with ideas, our CEO wants to decide on a ranking of these teams, so that she can give a ranked list to her Venture Capitalsts (VCs). Comparing startup ideas is tricky, and her VCs only trust "A/B testing", where two startup ideas are shown to a bunch of users, who then pick a favorite.

Given n teams, she compares *every pair* of teams to see who impresses users more in a "pitch-off". She wants to output a ranked list of the teams, where the best team beat all the others in A/B testing; the second-best team lost to the best team but beat all the remaining teams, etc.

However, after having all n-choose-2 pitch-offs, our CEO found the following issue: sometime team A beat team B, who beat team C, etc., who beat team Z, who beat team A. Such a cycle would mean that she cannot possibly produce a consistent final ranking to give to her VCs.

What she has to do in this case is "fire" some of the n teams, so that the k teams that remain can be ordered as desired. Her goal is to fire *as few teams as possible*, so that the teams that remain can be ranked unambiguously.

You have been hired as a consultant, to figure out which teams to fire. You soon realize that finding the *optimum* number is NP-hard, and instead you decide to come up with a *3-approximation*. Namely, your task is to find a polynomial time algorithm that fires at most 3 times as many teams as the optimal scheme would.

Hint: Show that if there is an "inconsistent sequence" $\mathrm{A}, \mathrm{B}, \mathrm{C}, \ldots, \mathrm{Z}$ where each one beats the one after it and the last one beats the first one, then there *must* exist an inconsistent sequence of length 3.

Problem 5

(20 points) Now that our CEO has her startup list finalized, she wants to spend a week pitching her list to as many Venture Capitalists (VCs) as possible! Find either a polynomial time algorithm or a proof of NP-hardness for the following problem (essentially a restricted version of problem 2.1, but a generalization of problem 1.1 from the previous homework): There are n VCs our CEO wants to meet with, and each VC wants to meet **twice**, between some arbitrary times (Tuesday from 11:13 AM to 12:54 PM, and Friday from 4 PM to 8:57 PM, for example). Find the maximum number of meetings she can take that do not overlap.

Important new rules for this problem: This problem is harder than usual, and we will post hints over the course of the week. However, *you* will have to come up with the hints! The game is as follows: when you think you have a significant idea for how to make progress on this problem, send a brief email to the TA list outlining the idea. If you email us an idea worth sharing, you and your partner will get **extra credit**! The **earlier** you send us the hint, the **more credit** you get! (If multiple groups email us the same idea before we send it out, they will all get credit; each partner group can receive extra credit at most once for this problem.) Start thinking about this problem **early**.