Ethics Mini-Project 3: P vs. NP

Due Date: This assignment is due at 11:59 PM Friday March 12 (Seattle time, i.e. GMT-8).
You will submit as a PDF to gradescope.

Algorithms have effects in the real-world. The assumption that $P \neq NP$ is already baked into a lot of the real world.

The goal of this exercise is for you to consider the effects of running algorithms in the real-world. This assignment is a mix of technical tasks and non-technical ones. The technical aspects can be "right" or "wrong", but the non-technical aspects are unlikely to be simply "right" or "wrong" – we won't have to **agree** with the non-technical aspects of your analysis to consider them a good analysis. Our evaluation will be based on how well they connect to the technical aspects, as well as the depth of reasoning demonstrated.

Collaboration Policy

You are to conduct your own thinking and analysis for this assignment. While you may get feedback form other students on your writing, you cannot just use the results of another student's work.

"real-world"

Most of the questions in this assignment are about dealing with a sudden resolution of the P vs. NP question. Most experts agree that $P \neq NP$ is **far** more likely than $P = NP$, and most experts think P vs. NP won't be resolved for decades. Nonetheless, it's informative (and hopefully fun) to imagine the possible effects, no matter how unlikely.

This assignment is intended to be open-ended (we don't have a list of expected answers here); feel free to add-in extra assumptions to the situations described if it helps you formulate a response.

1. Security Ethics

Most companies have ethical guidelines for security experts who might notice (or create) security issues with their code. In exchange for revealing vulneratilities to the company (allowing for a fix) instead of the general public, companies put restrictions on what testing they consider legitimate and how long someone must keep a vulnerability secret before publicizing it. As an example, read this one used by the U.S. Department of Defense or this sample one for a fake company.

2. P, NP, and Security

Most online applications (email, social media, banking accounts, shopping accounts,...) use passwords for security. "Is this password corect?" is an NP-problem, as currently implemented.

When you enter your password, it gets encrypted (a very complicated hash function is applied to it), and is sent to the service you want to log into. The service then compares the hashed password submitted to what it has stored (the hashed version of your true password). If the hashes match, then the service lets you log-in!

Passwords are fundamentally an NP-problem:
Input: A hash (supposedly of a password)
Output: True if there is a password whose hash of the input matches the stored hash of the true password.

The password check is (essentially) a verifier (is this certificate, the proposed password, a good certificate?), while someone trying to get unauthorized access is solving the search problem (what is the password?)

When databases of **hashed** passwords are leaked, that's bad for security, but often not *that* bad. There is still a difficult problem of figuring out what unhashed password would match the hashed stored password. In a world where $P = NP$ though, that problem would no longer be as difficult.

3. Passwords

(a) For normal password systems, making substantial progress on P vs. NP (e.g. an algorithm that efficiently solves a large fraction of 3-SAT instances that couldn't be solved before, or even a polynomial time algorithm for 3-SAT) could theoretically form a security issue for a lot of the modern internet. Suppose your company made some substantial progress on a 3-SAT algorithm, and you realize that your 3-SAT improvement has actually made it easier to find unhashed versions of the passwords used for logging into your own app. What should your company do? (Delete your 3-SAT improvement? Make it top-secret? Force everyone to use 2-factor authentication? Something else?) Why? (2-3 sentences).

(b) Your company uses standard password libraries. Should you disclose your new algorithm to the writers of the library? What if they need details of the algorithm to update the library? (2-3 sentences).

4. More P, NP

(a) Passwords aren't the only system that would be affected by substantial positive progress on P vs. NP. Think of at least one other potential **negative** side-effect of proving $P = NP$. Describe what NP problem you can now solve (by describing inputs, outputs, and the certificate) and what negative effect might result (3-4 sentences).

(b) Think of at least one potential **positive** side-effect of proving $P = NP$. Describe the NP problem you can now solve (by describing inputs, outputs, and the certificate) and what positive effect might result (3-4 sentences).

(c) For the most part we live our lives as though $P \neq NP$ (we don't know any efficient algorithm for an NP-complete problem, so it's pretty easy to live as though they don't exist). Suppose someone actually proved $P \neq NP$. From what you know now about P and NP, do you think this would affect you? If so (or not), describe how and why (3-4 sentences).