CS41 Homework 8

This homework is due **11am Wednesday November 9**. Submit this homework using **github**. For this homework, you will work with a partner. It’s ok to discuss approaches with others at a high level, but most of your discussions should just be with your partner. The only exception to this rule is work you’ve done with a lab partner **while in lab**. In this case, note who you’ve worked with and what parts were solved during lab. If there are questions about academic integrity, please visit the section on Academic Integrity on the course website.

0. Before final submission, make sure to fill out the README file.

1. In this problem, you will prove that **Three-Coloring** is **NP-Complete**.

   (a) Consider the following verifier for **Three-Coloring**. The witness we request is a valid three coloring of the undirected graph $G = (V, E)$, which is specified as a list of two-digit binary strings $w = w_1w_2 \ldots w_k$ where we interpret

   $w_i = \begin{cases} 
   00, & \text{vertex } i \text{ is colored BLUE} \\
   01, & \text{vertex } i \text{ is colored GREEN} \\
   10, & \text{vertex } i \text{ is colored RED}
   \end{cases}$

   **threeColoringVerifier**($G = (V, E), w$)
   1  for each $w_i$ in $w$
   2      if $w_i = 11$
   3          return NO
   4  for $j$ from $i + 1$ to $|w|$
   5      if $w_i = w_j$ and $(i, j) \in E$
   6          return NO
   7  return YES

   This verifier is not quite right.
   Give an example witness $w$ and graph $G$ which is *not* three-colorable, such that

   **threeColoringVerifier**($G, w$) = YES

   (b) Rewrite **threeColoringVerifier** so that it is a valid verifier for **Three-Coloring**.

   (c) Prove that **Three-Coloring** $\in$ **NP**.

   (d) Given an input $x$ for 3-Sat, create an input for **Three-Coloring** using the gadgets below (Figures 1 through 5). For each clause in $x$, you should create a piece of the graph $G$ which will be an input for **Three-Coloring**.

   Describe how to do this, and what the final graph $G$ consists of. How is the satisfiability of the clause related to the colorability of the piece of the graph?

   Recall from lab that our gadgets are three-colorable graphs which include at least three vertices marked $a, b, c$. Except for the specified property, the remaining vertices are **unconstrained**. For example, unless the problem states that, *e.g.*, $a$ cannot be red, it
must be possible to color the graph in such a way that $a$ is red. Colors for other vertices may be fixed, just not $a, b, c$.

Figure 1: A graph such that $a, b, c$ all have different colors.

![Diagram](image1)

Figure 2: A graph such that $a, b, c$ all have the same color.

![Diagram](image2)

Figure 3: A graph such that $a, b, c$ do NOT all have the same color.

![Diagram](image3)

Figure 4: A graph such that none of $a, b, c$ can be green.

![Diagram](image4)
Figure 5: A graph such that none of $a, b, c$ are green, and they cannot all be blue.

(e) Run the Three-Coloring algorithm on the input $G$ you create, and output yes iff the Three-Coloring algorithm outputs yes. Argue why this procedure gives you a correct answer for 3-Sat. (Hint: Associate the color red with true and the color blue with false.)

2. A social networking site wants to find out whether its users form big groups of friends, or spread out in disparate clusters. To begin studying this topic, the site is interested in answering the following question: for some $k$, is there a group of $k$ people who are all friends?

Given a set of people $p_1, p_2, \ldots, p_n$, a list of friendships $F$ (so if $p_i$ and $p_j$ are friends, then $\{p_i, p_j\} \in F$), and an integer $k$, the problem is to decide whether there is a group of $k$ people who are all friends with each other.

Prove that this problem is NP-Complete. (Hint: reduce from Independent-Set.)

3. (Kleinberg and Tardos, 8.9) Consider the following problem. You are managing a communication network, modelled by a directed graph $G = (V, E)$. There are $c$ users who are interested in making use of this network. User $i$ (for each $i = 1, 2, \ldots, c$) issues a request to reserve a specific path $P_i$ in $G$ on which to transmit data.

You are interested in accepting as many of these path requests as possible, subject to the following restriction: if you accept both $P_i$ and $P_j$, then $P_i$ and $P_j$ cannot share any nodes.

Thus, the Path-Selection problem asks: Given a directed graph $G = (V, E)$, a set of requests $P_1, P_2, \ldots, P_C$ (each of which must be a path in $G$), and a number $k$, is it possible to select at least $k$ of the paths so that no two of the selected paths share any nodes?

Prove that the Path-Selection problem is NP-Complete.

4. Extra credit. (Kleinberg and Tardos, 8.20) You and a friend have been trekking through various far-off parts of the world and have accumulated a big pile of souvenirs. At the time you weren’t really thinking about which of these you were planning to keep and which your friend was planning to keep, but now the time has come to divide everything up.

Here’s a way you could go about doing this. Suppose there are $n$ objects, labelled $1, 2, \ldots, n$, and object $i$ has an agreed-upon value $x_i$. (We could think of this, for example, as a monetary...
resale value; the case in which you and your friend don’t agree on the value is something we won’t pursue here.) One reasonable way to divide things would be to look for a partition of the objects into two sets, so that the total value of the objects in each set is the same.

This suggests solving the following Number Partition Problem. You are given positive integers \( x_1, \ldots, x_n \); you want to decide whether the numbers can be partitioned into two sets \( S_1 \) and \( S_2 \) with the same sum:

\[
\sum_{x_i \in S_1} x_i = \sum_{x_j \in S_2} x_j
\]

Show that the Number Partition Problem is NP-Complete.

5. **Extra extra credit.** Is \( P = NP \)? Prove or give a counterexample.