CS41 Homework 11: NP-Completeness

This homework is due at 11:59PM on Sunday, December 2. Write your solution using LaTeX. Submit this homework using **github** as a **.tex** file. This is a **partnered homework**. You should primarily be discussing problems with your homework partner.

It's ok to discuss approaches at a high level with others. However, you should not reveal specific details of a solution, nor should you show your written solution to anyone else. The only exception to this rule is work you've done with a lab partner *while in lab*. In this case, note (in your **README** file) who you've worked with and what parts were solved during lab.

1. Multiple-Independent-Set (K&T 8.14) In this problem, there is a machine that is available to run jobs over some period of time, say 9AM to 5PM.

   People submit jobs to run on the processor; the processor can only work on one job at any single point in time. However, in this problem, each job requires a **set of intervals** of time during which it needs to use the machine. Thus, for example, one job could require the processor from 10AM to 11AM and again from 2PM to 3PM. If you accept this job, it ties up your machine during these two hours, but you could still accept jobs that need any other time periods (including the hours from 11AM to 2PM).

   Now, you're given an integer k and a set of n jobs, each specified by a set of time intervals, and you want to answer the following question: is it possible to accept at least k of the jobs so that no two of the accepted jobs have any overlap in time?

   In this problem, you are to show that Multiple-Independent-Set $\in$ NP-complete. To assist you, we've broken down this problem into smaller parts:

   (a) First, show that Multiple-Independent-Set $\in$ NP.

   (b) In the remaining two parts, you will reduce

   $$\text{Independent-Set} \leqslant_P \text{Multiple-Independent-Set} .$$

   Given input $(G = (V, E), k)$ for Independent-Set, create a valid input for Multiple-Independent-Set. First, divide the processor time window into m distinct and disjoint *intervals* $i_1, \ldots, i_m$. Associate each interval i_j with an edge e_j. Next, create a different job J_v for each vertex $v \in V$. What set of time intervals should you pick for job J_v?

   (c) Finally, run the Multiple-Independent-Set algorithm on the input you create, and output yes iff the Multiple-Independent-Set algorithm outputs yes. Argue that the answer to Multiple-Independent-Set gives you a correct answer to Independent-Set.

2. Intersection-Inference (K&T 8.16) Consider the problem of reasoning about the identity of a set from the size of its intersections with other sets. You are given a finite set U of size n, and a collection $A_1, \ldots, A_m \subset U$ of subsets of U. You are also given integers $c_1, \ldots, c_m$. The question is: does there exists $X \subset U$ such that for each $i = 1, 2, \ldots, m$, the cardinality of $X \cap A_i$ equals c_i? We will call this an instance of the Intersection-Inference problem, with input $U, \{A_i\}, \{c_i\}$.

   Prove that Intersection-Inference is NP-complete, **Hint:** reduce from the following problem, which you may assume is NP-complete:

   Problem One-In-Three-Sat:

   Inputs: n variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$ where each clauses is the or of three literals e.g., $c_i = (x_1 \vee \overline{x_2} \vee x_3)$.

   Output: yes iff there is a truth assignment to the variables such that for each clauses there is **exactly** one satisfied variable.

   Hint: Let U be the set of literals. You'll have to work to ensure that a variable and its negation cannot both end up in X.

3. PATH-SELECTION (K&T 8.9) Consider the following problem. You are managing a communication network, modeled by a directed graph $G = (V, E)$. There are c users who are interested in making use of this network. User i (for each $i = 1, \ldots, c$) 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 $P_1, \ldots, P_c$ (each of which is a path in G), and a number k, output YES iff it is possible to select at least k paths so that no two of the selected paths share any nodes.

   Prove that PATH-SELECTION is NP-COMPLETE.

4. **(extra challenge)** Show that ONE-IN-THREE-SAT is NP-COMPLETE.