COL351 Holi2023: Tutorial Problem Set 11

1. Let U be a finite set and, let $\mathcal{S} \subseteq 2^U$ be a collection of subsets of U. We say that a subset $\mathcal{C} \subseteq \mathcal{S}$ is a *set cover* of U if $\bigcup_{A \in \mathcal{C}} A = U$. In the `SetCover` problem, we are given U, $\mathcal{S} \subseteq 2^U$, and an integer k as input. We are required to determine whether there is a set cover $\mathcal{C} \subseteq \mathcal{S}$ such that $|\mathcal{C}| \leq k$. Prove that `SetCover` is `NP-complete`.

2. In the `SubgraphIsomorphism` problem, the input consists of two graphs, say G and G', and we are required to determine whether G' is isomorphic to some subgraph of G. Prove that `SubgraphIsomorphism` is `NP-complete`.

   (We do not know whether `GraphIsomorphism` is `NP-hard`, or whether it is in `P`. This is another problem which, if you solve, you will get fame, admission in top universities for a PhD, and needless to say, an A in COL351 irrespective of your performance.)

3. Recall the $0-1$ knapsack problem: you are given the weights and values of n items and an integer W, and the goal is to find a max value subset of the items with total weight at most W. Define the *natural* decision problem associated with the $0-1$ knapsack problem and prove that it is `NP-complete`.

4. Recall that a boolean formula is said to be a 3CNF formula if it is an AND of clauses, where each clause is an OR of at most 3 literals. Let us call a boolean formula a *strict 3CNF* formula if it is an AND of clauses, where each clause is an OR of exactly 3 literals derived from 3 distinct boolean variables. In the `Strict3SAT` problem, we are given a strict 3CNF formula, and we need to determine whether it is satisfiable. Prove that `Strict3SAT` is `NP-complete`.

5. Consider a graph $G = (V, E)$. Recall that if G is connected, then G contains a spanning tree. Now, instead of connecting all vertices, suppose our task is to build a tree using edges of G and connect a given subset $S \subseteq V$ of vertices to one another, possibly through vertices not in S. Such a tree is called a *Steiner tree* of S in G. Obviously, such a tree must have at least $|S| - 1$ edges. However, depending on the graph, we might need many more. An extreme case is where if G is a path and S contains only the two endpoints, in which case our tree must include all the edges. Thus, it makes sense to find a Steiner tree which has as few edges as possible. Define the *natural* decision problem associated with this problem and prove that it is `NP-complete`.

6. The crazy instructor of COL351 has an equally crazy criterion for passing the course. The instructor gives n tutorial problems over the semester, and asks his students to submit their answers within 24 hours after the major exam. Every student gets zero or one mark in every problem. For a student to pass the course, their marks must satisfy all of m constraints specified by the instructor. Specifically, a student who gets marks $x[1], \ldots x[n]$ passes the course if and only if for each $i \in \{1, \ldots, m\}$, $\sum_{j=1}^{n} A[i][j] \times x[j] \geq b[i]$. Here $A[i][j]$ and $b[i]$ are integers.

   Given the matrix of integers $(A[i][j])$ where $i = 1, \ldots, m$ and $j = 1, \ldots, n$, and the array $b[1], \ldots, b[m]$, the students would like to know whether it is at all possible to pass the course, that is, whether there exist $x[1], \ldots, x[n] \in \{0, 1\}$ such that for each $i \in \{1, \ldots, m\}$, $\sum_{j=1}^{n} A[i][j] \times x[j] \geq b[i]$. Prove that this problem is `NP-complete`.