CS 330 – Summer 2019 – Lab 11 Solutions

Question 1 (Set Packing)**.**
Given a universe $U = i_1, i_2, \ldots, i_n$ of n items and m subsets $S_j \subseteq U$, find the maximum number of subsets that are independent (that is, they don't have an item in common.). Prove the complexity of this problem.

Solution: Firstly, we can tell that the decision version of this problem is as following: Given a universe $U = i_1, i_2, \ldots, i_n$ of n items and m subsets $S_j \subseteq U$, is there a set of k subsets that are independent? We can solve this problem by trying all the sets of k subsets. The certificate in this case is a set S of k subsets. We check if S is a correct solution by checking every subset in S and seeing if it shares a common item with any other subset in S. Thus, the set packing problem is in NP. Next, we show that set packing problem is $NP - complete$ through reduction between the independent set problem and set packing problem: Credit to Wikipedia: Given a set packing problem on a collection S which contains m subsets S_j, we can create a graph where for each subset $S_j \in S$ there is a vertex V_{S_j} , and there is an edge between V_{S_i} and V_{S_j} if $S_i \cap S_j \neq \varnothing$. Then, because vertices in an independent set do not share any edge, any independent set in the graph represents a set of subsets that do not share an item. We can create this graph in polynomial time using the same method we used to show this problem was in NP. Thus, every independent set of vertices in the generated graph corresponds to a set packing in S and whenever there is a set packing in S, there is an independent set in G.
Since the independent set problem is NP-complete and the independent set problem $\leq_P$ set packing problem, the set packing problem is also NP-complete.

Question 2 (Seating problem)**.**
Professor X is throwing a big party for his colleagues and is expecting n visitors. Lately some controversial issues have come up that resulted in many fights and some of his guests are not speaking to each other anymore. The professor wants to know whether he can seat his guests at a round table so that each guest is on speaking terms with both of her neighbors. We may assume that Professor X has exact knowledge about the friend-foe relationships among the guests. Prove the complexity of the Professor's task!

Solution:

We can solve this problem by trying all the ways of arranging n guests in a table. The certificate s in this case is a cyclic permutation P of n guests. The way that we check if P is a correct solution is by checking every guest in P and see if she is on speaking terms with both of her neighbors. (The neighbors of the last guest in P is the first guest and the second last guest in P.) Thus, seating problem is in NP. Then, we show that set packing problem is $NP - complete$ through reduction between Hamiltonian cycle and seating problem: We create a graph G in which each vertex represents a guest. Then, we add an edge between two vertices if and only if their representative guests talk to each other. In the worst case scenario, it would take $O(n^2)$ time to create this graph.
$\Rightarrow$ If there is a Hamiltonian cycle in G, then there is a solution to the seating problem: Fix an order for the vertices $v_1, v_2, \ldots, v_n$ of the Hamiltonian cycle. There is an edge e between two nodes v_i, v_{i+1} and v_n, v_1. Thus, we have a way to arrange the guests at the table such that any two adjacent guests at the table are on speaking terms.
$\Leftarrow$ If there is a solution to the seating problem, then there is a Hamiltonian cycle on G: The solution is a permutation P, where any two adjacent guests in this permutation are on speaking terms. Hence, P represents in G, a sequence of vertices where there is an edge between any v_i, v_{i+1} and v_n, v_1, thus a hamiltonian cycle.
Since the Hamiltonian cycle problem is NP-complete and the Hamiltonian cycle problem $\leq_P$ the seating problem, the seating problem is NP-complete.