31 (100 PTS.) My friend, parting time is pending.

The following question is long, but not very hard, and is intended to make sure you understand the following problems, and the basic concepts needed for proving NP-Completeness.

All graphs in the following have n vertices and m edges.

For each of the following problems, you are given an instance of the problem of size n. Imagine that the answer to this given instance is "yes", and that you need to convince somebody that indeed the answer to the given instance is `yes`. To this end, describe:

(I) An algorithm for solving the given instance (not necessarily efficient). What is the running time of your algorithm?
(II) The format of the proof that the instance is correct.
(III) A bound on the length of the proof (its have to be of polynomial length in the input size).
(IV) An efficient algorithm (as fast as possible [it has to be polynomial time]) for verifying, given the instance and the proof, that indeed the given instance is indeed `yes`. What is the running time of your algorithm?

(EXAMPLE)

Shortest Path

Instance: A weighted undirected graph G, vertices s and t and a threshold w.
Question: Is there a path between s and t in G of length at most w?

Solution:

(I) **Algorithm**: We seen in class the Dijkstra algorithm for solving the shortest path problem in $O(n \log n + m) = O(n^2)$ time. Given the shortest path, we can just compare its price to w, and return yes/no accordingly.
(II) **Certificate**: A "proof" in this case would be a path π in G (i.e., a sequence of at most n vertices) connecting s to t, such that its total weight is at most w.
(III) **Certificate length**: The proof here is a list of $O(n)$ vertices, and can be encoded as a list of $O(n)$ integers. As such, its length is $O(n)$.
(IV) **Verification algorithm**: The verification algorithm for the given solution/proof, would verify that all the edges in the path are indeed in the graph, the path starts at s and ends at t, and that the total weight of the edges of the path is at most w. The proof has length $O(n)$ in this case, and the verification algorithm runs in $O(n^2)$ time, if we assume the graph is given to us using adjacency lists representation.

31.A. (20 pts.)

Socially Distanced Set

Instance: A graph G, integer k
Question: Is there a distanced set in G of size k? A set $X \subseteq \mathsf{V}(\mathsf{G})$ is a distanced set if no two vertices of X are connected by an edge, or a path of length at most 4.

31.B. (20 pts.)

Edge Independent

Instance: A graph G, a set a parameter k.
Question: Is there a subset of k edges in the graph, such that no pair of edges is adjacent?

31.C. (20 pts.)

Sum to target

Instance: S: Set of positive integers. t: An integer number (target).
Question: Is there a subset $X \subseteq S$ such that $\sum_{x \in X} x - \sum_{y \in S \setminus X} y = t$?

31.D. (20 pts.)

4DM

Instance: X, Y, Z, W sets of n elements, and T a set of quadruples, such that $T \subseteq X \times Y \times Z \times W$.
Question: Is there a subset $S \subseteq T$ of n disjoint quadruples, such that every element of $X \cup Y \cup Z \cup W$ is covered exactly once by one of the quadruples of S?

31.E. (20 pts.)

SET DISJOINT COVER

Instance: $(\mathsf{U}, \mathcal{F}, k)$:
U: A set of n elements
$\mathcal{F}$: A family of m subsets of U, s.t. $\bigcup_{X \in \mathcal{F}} X = \mathsf{U}$.
k: A positive integer.
Question: Are there k pairwise-disjoint sets $S_1, \ldots, S_k \in \mathcal{F}$ that cover U?

Formally, the sets $S_1, \ldots, S_k$ ***cover*** U if $\bigcup_i S_i = \mathsf{U}$. They are ***pairwise-disjoint*** if for any $i \neq j$, we have that $S_i \cap S_j = \emptyset$.

32 (100 pts.) Clique of SATs

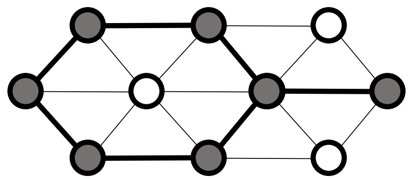
Given an undirected graph $G = (V, E)$, a partition of V into $V_1, V_2, \ldots, V_k$ is a ***clique cover*** of size k if each V_i is a clique in G. CLIQUE-COVER is the following decision problem: given G and integer k, does G have a clique cover of size at most k?

32.A. (80 pts.) Describe a polynomial-time reduction from CLIQUE-COVER to SAT.

32.B. (20 pts.) Does this prove that CLIQUE-COVER is NP-Complete? You just need to provide a yes/no answer with clear/concise explanation.

33 (100 PTS.) No Triangles club.

A subset S of vertices in an undirected graph G is ***triangle-free*** if, for every triple of vertices $u, v, w \in S$, at least one of the three edges uv, uw, vw is *absent* from G. Prove that finding the size of the largest triangle-free subset of vertices in a given undirected graph is NP-hard.



Above is a triangle-free subset of 7 vertices.
This is ***not*** the largest triangle-free subset in this graph.