Problem 1

Consider the following two languages:

\textsc{VertexCover} = \{\langle G, k \rangle \mid G \text{ is an undirected graph, } k \text{ is a number of vertices, and there exists a set } S \text{ of } k \text{ or fewer vertices of } G \text{ such that every edge of } G \text{ is connected to at least one vertex in } S.\}

\textsc{SubsetUnion} = \{\langle S, Q, k \rangle \mid S \text{ is a set of integers, } Q \text{ is a set of sets of integers, and } k \text{ is a positive integer such that there is a collection of } k \text{ elements } Q \text{ where the union of those } k \text{ elements is } S.\}\}

For example, if

\[ S = \{1, 2, 3, 5, 7, 9\}, Q = \{\{1, 2, 5\}, \{1, 3, 7\}, \{2, 3, 5, 9\}\}, k = 3, \]

then \( \langle S, Q, k \rangle \in \textsc{SubsetUnion} \) since \( \{\{1, 2, 5\}, \{1, 3, 7\}, \{2, 3, 5, 9\}\} \) contains 3 sets from \( Q \) and the union of those 3 sets is equal to \( S \).

Alternatively, if

\[ S = \{1, 2, 3, 4, 5\}, Q = \{\{1\}, \{2, 3\}, \{4, 5\}\}, k = 2, \]

then \( \langle S, Q, k \rangle \notin \textsc{SubsetUnion} \) since no two elements of \( Q \) union to form \( S \).
a. Explain in a few sentences how one can reduce the language VERTEX-COVER to the language SUBSETUNION in polynomial time.

b. Prove that your reduction works.

c. Explain why your reduction has polynomial running time.

d. Prove that if VERTEXCOVER is NP-Complete, then SUBSETUNION is also NP-Complete.

Problem 2

Rudy and Ginger are blissfully in love, but are getting tired of sneaking around in order to keep their relationship secret from their families. In a desperate attempt to spend more time together, they come up with a plan. They decide to use root vegetables cooked in each of their family’s kitchens, as well as an old Turing machine from Rudy’s JicaMate dating days, to create a life-size robotic replica of Ginger.\(^1\) Ginger’s robotic double would take her place in her everyday life, allowing the two star-crossed lovers to be together at Rudy’s parsnip kitchen in the countryside. In order for their plan to work, the robot must not only look like Ginger, but it must be able to convince the O’Nions of Ginger’s identity if it is discovered and questioned.

After several days, Rudy and Ginger think that they have successfully created the robot, which they call Ginger 2.0. For the robot to pass as Ginger, it must be as good at answering computer science questions as she is. They ask it the questions below, to which it gives answers. Check the robot’s work to confirm that it is, in fact, a convincing replica of Ginger.

For this problem, only consider languages over the alphabet \{0, 1\}.

a. Let \(L_1\) and \(L_2\) be two languages in NP.

   (i) Must \(L_1 \cup L_2\) be in NP?

   (ii) Must \(L_1 \cap L_2\) be in NP?

b. Now, suppose \(L_1\) and \(L_2\) are both NP-complete.

   (i) Must \(L_1 \cup L_2\) be NP-complete?

   (ii) Must \(L_1 \cap L_2\) be NP-complete?

\(^1\)It’s just like one of those potato-powered alarm clocks, except it’s a humanoid, socially competent AI.
Problem 3

Rudy and Ginger teach Ginger 2.0 everything it will need to take Ginger’s place in society, including general Myami and O’Nion family lore. The root vegetable cooking community actually has quite an extensive volume of literature, including such classic cookbooks as *The Great Gatsbeet*, *The Catcher in the Rhizome*, and *Twenty Thousand Leeks Under the Sea*. Ginger 2.0 needs to understand cultural references to these cookbooks, but they sometimes reference one another. This could lead to an unfortunate cycle of referencing the same things over and over again, which would not make for a convincing impression of Ginger. In their directed graph model of popular cooking culture, Rudy and Ginger want to cover all the important cookbooks, but need to remove vertices so that Ginger 2.0 won’t get stuck in any cycles.

Consider the following language:

\[
\text{COOKBOOKCOVER} = \{(G, k) \mid G \text{ is a directed graph, } k \in \mathbb{N}, \text{ and } \\
\exists V' \subseteq V \text{ such that } |V'| \leq k, \\
E' = \{(u, v) \in E \mid u \in V' \text{ or } v \in V'\}, \text{ and } \\
G' = (V - V', E - E') \text{ is acyclic}\}
\]

In this problem, we will prove that COOKBOOKCOVER is NP-Complete. To do so, recall the language VERTEXCOVER:

\[
\text{VERTEXCOVER} = \{(G, k) \mid G \text{ is an undirected graph, } k \text{ is a number of vertices, } \\
\text{ and there exists a set } S \text{ of } k \text{ or fewer vertices of } G \text{ such that } \\
\text{ every edge of } G \text{ is connected to at least one vertex in } S\}
\]

We will show NP-completeness via a reduction from VERTEXCOVER.

a. In a reduction from VERTEXCOVER to COOKBOOKCOVER, explain what the input to and output from the reduction is.

b. Describe a reduction for creating an instance \((G', k')\) of COOKBOOKCOVER from an arbitrary instance \((G, k)\) of VERTEXCOVER with the following properties:

i. If \((G, k) \in \text{VERTEXCOVER}\) then \((G', k') \in \text{COOKBOOKCOVER}\)
ii. If $\langle G', k' \rangle \in \text{CookbookCover}$ then $\langle G, k \rangle \in \text{VertexCover}$

Make sure that your reduction takes polynomial time, and justify why it satisfies the two properties above.

**Hint:** Suppose for a vertex cover $U \subseteq V$ of $G = (V, E)$, we remove from $G$ all vertices of $U$ and all edges incident to any vertex in $U$. What edges remain in the graph?

c. Explain why the reduction you provided above shows that CookbookCover is NP-hard.

d. Explain why CookbookCover is in NP and conclude that it is NP-complete.