# Solution Outline

To establish that the Subgraph-Isomorphism Problem qualifies as NP-complete, we must complete two essential proofs:

1. **NP Membership Proof**: Demonstrate that any valid solution to the problem can be efficiently verified. 2. **NP-Hardness Proof**: Show that a known NP-complete problem can be reduced to the Subgraph-Isomorphism Problem, indicating that this problem is at least as challenging as any NP-complete problem.

## 1. Demonstrating Membership in NP

For the Subgraph-Isomorphism Problem to be in NP, we need to verify that a proposed solution can be checked quickly. In this problem, the certificate should confirm whether a subgraph $G_1$ (the smaller graph) is embedded within $H$ (the larger graph) as an isomorphic subgraph.

- **Certificate Definition:**
  - The certificate in this case includes two main components:
    - A subset of vertices $V' \subset V_H$ that is hypothesized to correspond to the vertices of $G_1$.
    - A mapping $f : V_{G_1} \to V'$ that assigns each vertex in $G_1$ to a unique vertex in $H$, suggesting a potential isomorphic structure.
- **Steps for Solution Verification:** The verification process consists of multiple checks to ensure that $G_1$ is indeed isomorphic to a subgraph of $H$. Each check is designed to be performed in polynomial time, ensuring efficient verification. Here are the steps:
  1. **Check for Unique Vertex Mapping:**
     - First, we verify that the mapping $f$ is one-to-one, meaning each vertex in $G_1$ corresponds uniquely to a vertex in $V' \subset V_H$.
     - Specifically, this mapping should not allow any two vertices in $G_1$ to map to the same vertex in $H$. This one-to-one property ensures that each vertex in $G_1$ maintains its structural position within $H$.
     - To perform this verification, we iterate over each vertex in $V_{G_1}$ and check that each vertex is assigned a distinct counterpart in $V'$. This check requires $O(|V_{G_1}|)$ time, as each vertex in $G_1$ is mapped once to a vertex in $H$.
  2. **Edge Correspondence Verification:**
     - After confirming the vertex mapping, we move on to verifying the edges. For $G_1$ to be isomorphic to a subgraph in $H$, the edges in $G_1$ must also map correspondingly within $H$.

- We examine each edge $(u, v) \in E_{G_1}$ and check if there is a corresponding edge $(f(u), f(v)) \in E_H$ between the mapped vertices in $H$. This edge verification step ensures that all adjacency relationships in $G_1$ are preserved in $H$.
- This check involves iterating over each edge in $E_{G_1}$, verifying that it maps correctly in $H$, which can be done in $O(|E_{G_1}|)$ time.

3. **Completeness and Consistency of Mapping:**
   - The verification process also requires confirming that every vertex and edge in $G_1$ has a mapped counterpart in $H$, with no extraneous mappings that would disrupt the isomorphism.
   - By ensuring that the mapping fully encompasses all vertices and edges in $G_1$, we verify that no part of $G_1$ remains unmapped and that the subgraph structure is consistently preserved in $H$.

## 2. Establishing NP-Hardness

To show that Subgraph-Isomorphism is NP-hard, we need to find a way to solve a known NP-complete problem by reducing it to the Subgraph-Isomorphism Problem. Here, we use the **Clique Problem**, a well-known NP-complete problem, for our reduction.

- **The Clique Problem Recap:**
  - In the Clique Problem, we are given an undirected graph $G$ and an integer $k$.
  - The task is to determine whether there exists a complete subgraph (or clique) of $k$ vertices within $G$. In other words, we need to find if there exists a subset of $k$ vertices in $G$ such that every pair of vertices in this subset is connected by an edge.
- **Reduction to Subgraph-Isomorphism:**
  - To perform the reduction, we take an instance of the Clique Problem, i.e., a graph $G$ and a target integer $k$, and transform it into an instance of Subgraph-Isomorphism.
  - The goal is to determine if $G$ contains a subgraph that is isomorphic to a complete graph $H_k$, where $H_k$ represents a clique with exactly $k$ vertices.

    Here's a breakdown of the steps involved in this reduction:

    1. **Constructing $H_k$:**
       - We first construct $H_k$, which is the complete graph on $k$ vertices. In $H_k$, every vertex is connected to every other vertex, resulting in a graph with $k$ vertices and $\binom{k}{2}$ edges.

* The structure of $H_k$ ensures that any isomorphic subgraph found within $G$ would represent a clique of size $k$ within $G$.

2. **Setting Up the Subgraph-Isomorphism Query:**
   * Now that we have $H_k$, we reformulate the problem of finding a $k$-clique in $G$ as a question of subgraph isomorphism. Specifically, we ask whether there exists a subset of vertices and edges in $G$ that form a subgraph isomorphic to $H_k$.
   * This means we are looking for a subset $V' \subset V_G$ with exactly $k$ vertices and a corresponding edge set $E' \subset E_G$ such that $G[V']$, the subgraph of $G$ induced by $V'$, is identical in structure to $H_k$.
3. **Verifying Isomorphism with $H_k$:**
   * For $G$ to contain an isomorphic copy of $H_k$, every vertex in $V'$ must have edges connecting it to each of the other $k-1$ vertices in $V'$. This is equivalent to saying that $V'$ forms a clique in $G$.
   * Therefore, if an isomorphic subgraph to $H_k$ exists in $G$, then $G$ contains a clique of size $k$. Conversely, if $G$ does not contain a $k$-clique, no isomorphic subgraph of $H_k$ can be found in $G$.
4. **Time Complexity and Polynomial-Time Reduction:**
   * This reduction involves constructing $H_k$ and setting up the Subgraph-Isomorphism Problem by simply creating a query to check if $G$ has a subgraph isomorphic to $H_k$. These steps require only polynomial time relative to the size of $G$ and $k$.
   * Since the Clique Problem is NP-complete and we have shown how it can be reduced to the Subgraph-Isomorphism Problem in polynomial time, this establishes that Subgraph-Isomorphism is at least as hard as the Clique Problem. Consequently, the Subgraph-Isomorphism Problem is NP-hard.

## Conclusion

By combining the proofs for NP membership and NP-hardness, we confirm that Subgraph-Isomorphism is NP-complete, placing it in the class of computationally challenging problems that are central to complexity theory.