In this exercise, we want to show that the subgraph isomorphism problem is NP-complete. The subgraph isomorphism problem takes two undirected graphs G_1 and G_2 as input and asks whether G_1 is isomorphic to a subgraph of G_2.

Theorem 1. *The subgraph isomorphism problem is NP-complete.*

Proof. A problem is NP-complete if it is in NP and every problem in NP is reducible to it in polynomial time. First, we show that the subgraph isomorphism problem is in NP, i.e. we can verify a proposed solution in polynomial time. Given graphs G_1 and G_2 and a mapping $f : V_{G_1} \to V_{G_2}$, we must determine the following in polynomial time:

1. Whether f is an injection.
2. Whether for all edges $(u, v) \in E_{G_1}$, $(f(u), f(v)) \in E_{G_2}$.

The function f is an injection if and only if for all $u, v \in V_{G_1}$, $f(u) = f(v)$ implies $u = v$. A simple algorithm to verify this is as follows:

Algorithm 1: Verify Injection

Input: Graphs G_1 and G_2, mapping $f : V_{G_1} \to V_{G_2}$
Output: Whether f is an injection

```
Initialize hash map H;
for u ∈ V_G1 do
    if H[f(u)] is defined then
        return false;
    end
    H[f(u)] = u;
end
return true;
```

The runtime of this algorithm is bounded solely by the number of vertices in G_1, i.e. $O(|V_{G_1}|)$, so it runs in polynomial time. To verify the second condition, we simply iterate over all edges in G_1 and check whether the corresponding edges exist in G_2. A simple algorithm to verify this is as follows:

Algorithm 2: Verify Edge Mapping

Input: Graphs G_1 and G_2, mapping $f : V_{G_1} \to V_{G_2}$
Output: Whether f maps edges of G_1 to edges of G_2

```
Initialize hash set S from E_G2;
for (u, v) ∈ E_G1 do
    if (f(u), f(v)) ∉ S then
        return false;
    end
end
return true;
```

The runtime of this algorithm is bounded by the number of edges in G_1, i.e. $O(|E_{G_1}|)$, so it runs in polynomial time. Therefore, the subgraph isomorphism problem is in NP.

Next, we want to show the following:

1. The NP-complete problem 3-SAT can be reduced to the subgraph isomorphism problem.
2. The reduction from 3-SAT to the subgraph isomorphism problem preserves answers.
3. The reduction from 3-SAT to the subgraph isomorphism problem can be done in polynomial time.

Given a 3-SAT formula φ with n boolean variables $x_1, x_2, \ldots, x_n$ and m clauses $C_1, C_2, \ldots, C_m$, we can construct the graphs G_1 and G_2 as follows: $\square$