1. Below is a definition of the graph isomorphism problem.

   Input: two graphs, $G_1 = (V_1, E_1)$, and $G_2 = (V_2, E_2)$.
   Question: Can the nodes of $G_1$ be renamed s.t. $G_1$ becomes $G_2$?

   In other words, is there a one-to-one function $f : V_1 \rightarrow V_2$ such that for any edge $(x, y) \in E_1$ if only if $(f(x), f(y)) \in E_2$. Show that the graph Isomorphism problem is in NP.

   **Solution:** To show that this is in NP, we need to show it has a poly-time verifier. A certificate in this case is a one-to-one mapping $\delta$ from nodes in $V_1$ to nodes $V_2$. For each pair of nodes $u$ and $v \in V_1$, the verifier checks if $(u, v) \in E_1$ and if $(\delta(u), \delta(v)) \in E_2$. If the two answers are consistent (both yes, or both no), then it is ok. If all pair of nodes checks ok, then the certificate is a verified solution. This procedure clearly runs in poly time ($O(|V|^2)$).

2. 8.2

   We impose an arbitrary ordering on the edges and remove the edges one by one. If the graph obtained by removing an edge $e$ from the current graph still has a Rudrata path, we remove $e$ permanently and update the current graph. If the new graph does not have a Rudrata path, we add $e$ back. Hence, we maintain the invariant that the graph we have always has a Rudrata path (if the given graph did). Since it is possible to remove all edges except a single Rudrata path, we will be left with a Rudrata path in the end. This involves solving the Rudrata path decision problem $|E|$ times, if each of which requires poly time, overall, this is a poly-time algorithm for Rudrata Path.

3. 8.4

   **Solution:** (a.) A certificate in this case is a set of nodes. It is easy to check for each pair of nodes in the set whether there indeed is an edge in the original graph connecting them. If the answer is yes for all pairs, this is a valid clique. If the set size is greater than the given number, then the certificate checks ok. The run time is quadratic in the size of the set, clearly poly-time.
   (b.) The direction of the reduction is wrong. The given reduction shows that CLIQUE is at least as hard as 3-CLIQUE but does not establish the hardness of 3-CLIQUE.
   (c.) The statement “a subset $C \subseteq V$ is a vertex cover in $G$ if and only if the complimentary set $VC$ is a clique in $G$” used in the reduction is false. $C$ is a vertex cover if and only if $VC$ is an independent set in $G$.
   (d.) The largest clique that can be formed in such a graph is size 4. Thus we can exhaustively enumerate and search through all size 4 sets, which takes $O(|V|^4)$ time.

4. 8.9

   **Solution:** First we show that Hitting set is in NP. A certificate in this case is a set $H$. The verifier simply checks for each $S_i$ if $H \cap S_i = \emptyset$. If it checks ok (none-empty intersection) for every set $S_i$, then the solution is ok. This can be done in time that is linear in the total size of all $S_i$’s and $H$.

   We can then show that we can reduce vertex cover to the hitting set problem. Given a vertex cover problem $(G, b)$, each edge $(u, v)$ is simply a set containing $u$ and $v$. Finding a hitting set of size $b$ in this setting is the same as finding a vertex cover of size $b$ in the original graph. Thus hitting set is NP-Complete.
5. 8.10(a, b, d, e)
   Hint (a: CLIQUE; b: Rudrata Path; d: CLIQUE; e: Independent Set)

   Solution:
   a. We can view this as a generalization of the CLIQUE problem. Given an input \((G, k)\) for CLIQUE, let \(H\) be a graph consisting of \(k\) vertices with every pair connected by an edge (i.e. a clique of size \(k\)). Then \(G\) contains a clique of size \(k\) if and only if \(H\) is a subgraph of \(G\).

   b. This is a generalization of RUDRATA-PATH. Given a graph \(G\) with \(n\) vertices, let \(g = n1\). Then \((G, n1)\) is an instance of LONGEST PATH. However, a simple path of length \(n1\) must contain \(n\) vertices and hence must be a Rudrata path. Also, any Rudrata path is of length \(n1\). Hence for any graph with \(n\) vertices, LONGEST-PATH\((G, n1)\) is precisely the asking for the Rudrata path.

   d. This also a generalization of CLIQUE. Given an instance \((G, k)\) let \(a = k, b = k(k1)/2\). Any subgraph of \(G\) with \(k\) vertices and containing \(k(k1)/2\) edges, must have an edge between every possible pair out of these \(k\) vertices and hence must be a clique. Similarly, a clique on \(k\) vertices must contain \(k(k1)/2\) edges.

   e. This is a generalization of INDEPENDENT SET. Given \((G, k)\), an instance for independent set, let \(a = k, b = 0\). Then \((G, a, b)\) is an instance for SPARSE SUBGRAPH. Also, any subgraph of \(G\) with \(k\) vertices and 0 edges must be an independent set of size \(k\) and vice-versa.