1. **Reductions**
   Let \( A \leq B \) for two problems \( A \) and \( B \) mean that problem \( A \) can be solved in \( \bigO \) of the time it takes to solve problem \( B \).

   (a) Show that \( \text{MULTIPLICATION} \leq \text{SQUARING} \).
   (b) Show that \( \text{SQUARING} \leq \text{MULTIPLICATION} \).
   (c) Show that \( \text{SQUARING} \leq \text{RECIPROCAL} \).
   (d) If \( A \equiv B \) means \( A \leq B \) and \( B \leq A \) which of \( \text{MULTIPLICATION}, \text{SQUARING}, \) and \( \text{RECIPROCAL} \) are equivalent?

   **HINT:** \( \frac{1}{x} - \frac{1}{y} = \frac{y-x}{xy} \). Try \( y = x + 1 \).

2. **Lucas Numbers:**
   **INPUT:** A \( K \) bit number \( X \).
   **QUESTION:** Is \( X \) a Fibonacci number?
   The Lucas numbers are defined by the recurrence
   \[ L_n = L_{n-1} + L_{n-2} \]
   with the initial conditions: \( L_0 = 2, \ L_1 = 1 \)
   Show that this problem is in \( \mathcal{P} \) by outlining (NO CODE, just explain what you’re doing) an algorithm, AND showing that your algorithm runs in polynomial time in \( K \), the number of bits.

3. **Roots:**
   Without finding the solutions, show that \( x^2 - x - 1 = 0 \) has:
   (a) NO positive integer solutions
   (b) NO rational solutions
   **HINTS:**
   i. Assume that \( x = \frac{p}{q} \) where \( p \) and \( q \) are integers with no common factors.
   ii. \( p^2 - q^2 = (p - q)(p + q) \).
   iii. Each integer is either ODD or EVEN.

4. **Average Case:**
   Do Exercise 5.5 in the NOTES on page 61.

5. **Lower Bound:**
   Exercise 6.1 in the NOTES on page 71, is about the lower bound of \( \frac{3}{2} n - 2 \) comparisons to find the largest and smallest elements in an array. Devise a divide-and-conquer algorithm for this problem and show that the number of comparisons used by your algorithm achieves this lower bound.
6. **Boolean Expression:**
Assume that you have an algorithm $\text{YS}( )$ so that when you input a Boolean expression $E(x_1, \ldots, x_n)$ into $\text{YS}( )$,
$\text{YS}( E )$ outputs $\text{YES}$ if $E$ is satisfiable, and $\text{YS}( E )$ outputs $\text{NO}$ if $E$ is not satisfiable.

(a) Show how to use $\text{YS}( )$ to construct an algorithm $\text{FIND}( D(x_1, \ldots, x_n) )$ which when given a satisfiable Boolean expression $D(x_1, \ldots, x_n)$, returns an assignment $x_1 = a_1$, $x_2 = a_2$, $\ldots$, $x_n = a_n$, so that $D(a_1, \ldots, a_n)$ is $\text{TRUE}$.

(b) Assume that $\text{YS}( D(x_1, \ldots, x_n) )$ has run time $\mathcal{O}(n^k)$ and find the run time of $\text{FIND}( D(x_1, \ldots, x_n) )$.

7. **Platonic Hamiltonian Circuits:**
Show that each of the PLATONIC solids has a Hamiltonian circuit.

8. **s-t Hamiltonian Path:**
INPUT: A graph $G$ and two specified vertices $s$ and $t$.
QUESTION: Does $G$ have a Hamiltonian Path which starts at $s$ and ends at $t$?

(a) Assume that you know that Hamiltonian Circuit is $\mathcal{NP}$–Complete, show that s-t Hamiltonian Path is $\mathcal{NP}$–Complete.

(b) Assume that you know that s-t Hamiltonian Path is $\mathcal{NP}$–Complete, show that Hamiltonian Circuit is $\mathcal{NP}$–Complete.

(c) Show that the Yes/No version of TSP (Traveling Sales Person) with all edge weights in $\{1, 2\}$ is $\mathcal{NP}$–Complete. (You should assume that Hamiltonian Circuit is $\mathcal{NP}$–Complete.)

9. **Graph Isomorphism:**
Graph isomorphism is an example of a problem which is in $\mathcal{NP}$, but is not known to be $\mathcal{NP}$-complete, nor is it known to be in $\text{co-}\mathcal{NP}$.

INPUT: Two graphs, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.
QUESTION: Can the vertices of $G_1$ be renamed so that $G_1$ becomes $G_2$? (Is there a one-to-one onto function $f : V_1 \rightarrow V_2$ so that $\forall x, y \quad (x, y) \in E_1 \text{ iff } (f(x), f(y)) \in E_2$?

Show that GRAPH ISOMORPHISM is in $\mathcal{NP}$.
10. **Canonical Number:**
A graph with $n$ vertices can be represented as an $n \times n$ binary matrix which has a 1 in position $(i, j)$ if and only if there is an edge $(v_i, v_j)$. If you “unroll” this matrix (say by rows), you will have a vector of $n^2$ bits and you can consider this to be a number in standard binary notation. So, there is a correspondence between $n$ vertex graphs and $n^2$ bit numbers. If we re-label the vertices of the graph, we don’t change the graph properties. Different re-labelings of the graph will (usually) give different numbers. Clearly among all re-labelings of the graph, there is some re-labeling which gives the smallest value for this binary number. We would like to represent a graph by the minimum number we can get by re-labeling. We’ll call this minimal number the canonical number of the graph. It’s easy to see that two graphs are isomorphic iff they have the same canonical number.

(a) The graph $v_1 - v_2 - v_3$ is isomorphic to $v_1 - v_3 - v_2$ and is also isomorphic to $v_2 - v_1 - v_3$.
Find the canonical number of $v_1 - v_2 - v_3$.

(b) Show that if finding the canonical number of a graph is easy, then GRAPH ISOMORPHISM is easy. (Here, easy means takes polynomial time.)
However, canonical number may be harder than GRAPH ISOMORPHISM. If I can tell that two graphs are NOT isomorphic, I know that their canonical numbers are different, but I don’t know what their canonical numbers are. Further, if I know that two graphs are isomorphic, I know that their canonical numbers are identical, but again I don’t know what these canonical numbers are.

(c) **Is-Canonical:**

**INPUT:** A graph $G$ and an integer $I$.
**QUESTION:** Is $I < \text{the canonical number of } G$?
**EXERCISE:** Show that IS-CANONICAL is in co-$\mathcal{NP}$.

11. **Tautology:**

**INPUT:** A Boolean Expression $E(x_1, \ldots, x_n)$.
**QUESTION:** Does $E$ evaluate to TRUE for each and every assignment of TRUE and FALSE to the variables, the $x$’s ?

Show that TAUTOLOGY is co-$\mathcal{NP}$-complete.

12. **3-SAT:**

**INPUT:** A Boolean Expression $E(x_1, \ldots, x_n)$ in Clause form with at most 3 literals per clause.
**QUESTION:** Does $E$ evaluate to TRUE for some assignment of TRUE and FALSE to the variables, the $x$’s ?

Show that if SAT is $\mathcal{NP}$-complete, then 3-SAT is $\mathcal{NP}$-complete.