Problem 4

Note that the underlying graph is "simple," i.e., no "self-looping" edges.
Let \( G \) be an arbitrary (simple) graph with \( n \) vertices, \( n \geq 2 \).

Note that the sequence of possible degrees is:
\[
0, 1, 2, \ldots, n-1
\]
\( n \) possibilities.

We will eliminate (below) one of the \( n \) possibilities; hence by the pigeonhole principle, there exist at least two vertices with the same degree.

If \( G \) has no isolated vertex — vertices of degree 0 — then the sequence of possible degrees is: 1, 2, \ldots, n-1, and we are done.

Now, we consider the complementation of above case, i.e., there exists a vertex \( v \) with \( \deg(v) = 0 \).
Then, what can we say about the possible degrees of all other vertices?

We see that, for every vertex \( u \) of \( G \):
\[
\deg(u) \leq n-1
\]
(no self-looping, and no adjacency with \( v \)).
Then, the sequence of possible degrees is: 0, 1, \ldots, n-2, and we are done too.
Problem 5.

For parts (a) and (b), use mathematical induction. For example:

Example: (a) \( n \geq 0 \) \( |x_n| = |y_n| \)

Induction on \( n \).

Basis \( n = 0 \), prove \( P(0) \), i.e., \( |x_0| = |y_0| \).

Since \( x_0 = 0 \) and \( y_0 = 1 \), \( |x_0| = 0 \) and \( |y_0| = 1 \).

Inductive step: prove \( \forall n \geq 0 \) \( P(n) \Rightarrow P(n+1) \).

Let \( n \geq 0 \) be arbitrary.

Assume \( P(n) \), i.e., \( |x_n| = |y_n| \) — induction hypothesis.

Now, to prove \( P(n+1) \), i.e., \( |x_{n+1}| = |y_{n+1}| \), we consider:

\[ x_{n+1} = x_n \cdot y_n \]

by the inductive definition of \( \{x_n\} \) \( n \geq 0 \)

and \( |x_{n+1}| = 1 \cdot |x_n \cdot y_n| \)

\[ = |x_n| \cdot |y_n| \]

by a property of the absolute value function.

Similarly, we can see that

\[ |y_{n+1}| = |y_n \cdot x_n| \]

\[ = |y_n| \cdot |x_n| \]

\[ = |x_n| \cdot |y_n| \]

\[ = |x_n| \cdot |y_n| \]

\[ = |x_n \cdot y_n| \]

\[ = |x_{n+1}| \]

as desired.

This completes the induction step.

(by induction, we have shown that \( \forall n \geq 0 \), \( P(n) \) is true.)

(c) A string \( x \) is a palindrome iff \( x^r = x \).

We prove that \( \forall n \geq 0 \) \( \{x_n \text{ and } y_n \text{ are palindromes}\} \)

by mathematical induction (strong form) on \( n \).

Basis: \( n = 0 \), since \( P(0) \), i.e., \( x_{20}^r = x_{20} \) and \( y_{20}^r = y_{20} \).

Note: \( x_0^r = 0^r = 0 = x_0 \),

\( y_0^r = 1^r = 1 = y_0 \).
**Induction Step:** Prove that
\[ \forall n \geq 0 \quad \left( P(0) \land P(1) \land \ldots \land P(n) \right) \Rightarrow P(n+1) \]

Let \( n \geq 0 \) be arbitrary.

Assume \( P(0) \land P(1) \land \ldots \land P(n) \).

We prove \( P(n+1) \), i.e.,

\[ x_{2(n+1)} = x_{2n+1} \]
\[ y_{2(n+1)} = y_{2n+1} \]

**Consider** \( x_{2(n+1)} = x_{2n+2} \)

\[ = x_{2n+1} \cdot y_{2n+1} \quad \text{by the inductive hypothesis} \]

\[ = (x_n \cdot y_n) \cdot (y_n \cdot x_n) \quad \text{by the associative law} \]

Now, \( x_{2(n+1)} = (x_n \cdot y_n \cdot y_n \cdot x_n) \)

\[ = x_n \cdot y_n \cdot y_n \cdot x_n \quad \text{by the inductive hypothesis} \]

\[ = x_n \cdot y_n \cdot y_n \cdot x_n \quad \text{by the inductive hypothesis} \]

As desired.

The complete the induction step.

By induction, we have shown that \( \forall n \geq 0 \quad P(n) \) is true.
(d) From the inductive definition, \( \exists x_n \) with \( \sum_{n=0}^{\infty} y_n = 4 \), we observe that for all \( n \geq 2 \), \( y_n \) begins with 10 and ends with 10, if \( y_n \) begins with 10 and ends with 10.

Then, use an induction (on \( n \)) to show that neither \( x_n \) nor \( y_n \) contains 000 or 111 as substrings.

Problem 6.

(a) \( \{ w = a \} \) is in the language. \( \{ w = a \} \) is not in the language.

(b) \( \{ w = \varepsilon \} \) is in the language.

(c) \( \{ w = a \} \) is not in the language.

(d) \( \{ w = \varepsilon \} \) is not in the language.

(e) Can you find any string that is not in the language?
Problem 7. We disprove the given claim.

Suppose on contrary that we have two languages $L_1$ and $L_2$ such that $L_1 \neq \varepsilon^*$ and $L_2 \neq \varepsilon^*$, such that $L = L_1 L_2$.

Since $L = \{ u \in \varepsilon^* | u = vv \text{ for some string } v \in \varepsilon^* \}$, every even-length string of $0$s must be in $L = L_1 L_2$ by supposition, and hence are arbitrarily long strings of $0$s that are in either $L_1$ or $L_2$.

Similarly, there are arbitrarily long strings of $1$s that are in either $L_1$ or $L_2$.

Where are all those strings of $0$s and strings of $1$s distributed into $L_1$ or $L_2$?

Note that it is not possible for $L_1$ to have a non-empty string of $0$s as $L_2$ to have a non-empty string of $1$s, since their concatenation $(0, 1) = L$ could not be in $L$.

Similarly, we cannot have a string of $1$s in $L_1$ and a string of $0$s in $L_2$.

Thus, the only two remaining cases are for all the strings of $0$s or all the strings of $1$s to be in $L_1$ or for all those strings to be in $L_2$.

We show that these two cases may not exist.

Case when all the strings of $0$s or all the strings of $1$s are in $L_1$. Let $y$ be a string in $L_2$. 
Now, no string \( w \leq L_2 \) must contain both symbols 0s and 1s.

Since no one substring (long strings of 0s in \( L_1 \) or \( L_1 \) contains a string \( x \cdot y \) of \( 0s \) with \( |x| \geq |y| \).

Then, the concatenation \( xy \), which is in \( L_2 (=L) \) has 1s in its second half and not in its first half. Thus, \( xy \) cannot be in \( L \), which is a contradiction.

Case when all the strings 0s and all the strings 1s are in \( L_2 \); similar contradictory argument.

Therefore, the supposition that \( L = L_1 \) or \( L_2 \leq \text{sup} \) \( L \) is false.
Problem 8:

(a) We prove that, \( \forall \) languages \( L_1, L_2, L_3, (L_1 \cup L_2)L_3 \subseteq L_1 L_3 L_2 L_3 \)

Let \( L_1, L_2, \) and \( L_3 \) be arbitrary languages.

Consider an arbitrary \( x \in (L_1 \cup L_2)L_3 \).

So \( x = yz \) for some \( y \in L_1 \cup L_2 \) and \( z \in L_3 \),

that is, \( x = yz \) for some \( (y \in L_1 \) or \( y \in L_2 \)) \( \) and \( z \in L_3 \),

that is, \( x = yz \) for some \( (y \in L_1 \) and \( z \in L_3 \)) \( \) or \( (y \in L_2 \) and \( z \in L_3 \))

\( yz \in L_1 L_3 \) \( yz \in L_2 L_3 \)

\( x \in L_1 L_3 L_2 L_3 \), as desired.

(b) We prove that, \( \forall \) languages \( L_1, L_2, L_3, (L_1 \cup L_2)L_3 \subseteq (L_1 \cup L_2)L_3 \)

Let \( L_1, L_2, \) and \( L_3 \) be arbitrary languages.

Consider an arbitrary \( x \in L_1 \cup L_2 \cup L_3 \).

That is, \( x \in L_1 L_3 \) or \( x \in L_2 L_3 \).

First, consider the case: \( x \in L_1 L_3 \).

So \( x = yz \) for some \( y \in L_1 \) and \( z \in L_3 \).

This implies that \( x = yz \) for some \( y \in L_1 \cup L_2 \) and \( z \in L_3 \),

which gives that \( x \in (L_1 \cup L_2)L_3 \).

The case \( x \in L_2 L_3 \) is similar.
(c) We show that "A language $L_1, L_2, L_3$ 
$(L_1 - L_2) L_3 \subseteq L_1 L_3 - L_2 L_3$" is false.
That is, we show that 
$\exists$ languages $L_1, L_2, L_3$ 
$(L_1 - L_2) L_3 \not\subseteq L_1 L_3 - L_2 L_3$
(by a counterexample).

Consider $L_1 = \{a, b\}$, $L_2 = \{c, d\}$ 
$L_3 = \Sigma^* \subseteq \Sigma^*$ the underlying alphabet.
Then, $(L_1 - L_2) L_3 = (\{a, b\} - \{c, d\}) \Sigma^* = \{a, b\} \Sigma^*$, 
and $L_1 L_3 - L_2 L_3 = \{a, b\} \Sigma^* - \{c, d\} \Sigma^* = \{a, b\} \Sigma^* - \Sigma^*$.

Obviously, $(L_1 - L_2) L_3 \not\subseteq L_1 L_3 - L_2 L_3$.

(d) We prove that, "A language $L_1, L_2, L_3$, $L_1 L_3 - L_2 L_3 \subseteq (L_1 - L_2) L_3$.
Let $L_1, L_2, L_3$ be arbitrary languages.
Consider an arbitrary $x \in L_1 L_3 - L_2 L_3$.

So $x = yz$ for some $y \in L_1$ and $z \in L_3$, and $x \not\in L_2 L_3$.

Now, ask: Can $y \in L_2$?

No: If $y$ were in $L_2$, 
then $x = yz$ with $y \in L_2$ and $z \in L_3$.
These give that $x \in L_2 L_3$ which contradicts our assumption.

Hence, we have $x = yz$ for some $y \in L_1$ and $y \not\in L_2$ and $z \in L_3$.
So $x = yz$ for some $y \in L_1 - L_2$ and $z \in L_3$.

I.e., $x \in (L_1 - L_2) L_3$, as desired.