The Pumping Lemma:
If L is a regular language then there is a number p, such that any input string longer than p can be split into sections x, y, and z, such that:

1. For each \( i \geq 0 \), \( xy^iz \in L \)
2. \(|y| > 0\)
3. \(|xy| \leq p\)

Problem 1
Prove that the following language is regular:

\[ L_1 = \{ w \mid w \text{ is of the form } 0^31^n \text{ where } n \geq 0 \} \]

Prove: \( L_1 \) is regular

Proof Idea: We will construct a DFA to show that this language is regular

Proof (by construction):

Commentary from Megan:
Let’s set a pumping length of 4. Are there any strings in \( L_1 \) that are longer than 4?

Duhh.

- 00011
- 00011111
- 001111111

Since we set a pumping length of 4 on a 4-state machine, once we hit the 5th character of an input string, we’re guaranteed to have hit one of the states twice. Which means that if \( L_1 \) is regular, there must be at least one non-zero substring, \( y \) (2), in the first 4 characters (3) that we could either omit, or repeat an infinite number of times (1) and the resulting string would still be in the language.

Well that’s easy, how about that first 1? So:
\[ x = 000 \]
\[ y = 1 \]
\[ z = 111111 \]

If we omit \( y \) we'll still have three 0's followed by an infinite number of 1's. We can also repeat \( y \) as many times as we like and still end up with three 0's followed by an infinite number of 1's.

Will increasing the input string sizes change anything? Will increasing the pumping length change anything?

Nope. \( L_1 \) is regular AF.
Problem 2

Prove that the following language is not regular:

\[ L_2 = \{ w \mid w \text{ is a palindrome } \} \]

Prove: \( L_2 \) is not regular

Proof Idea: We will assume that the language is regular and show that it violates the pumping lemma, leading to the conclusion that it cannot be regular.

Proof (by Contradiction):

Assume \( L_2 \) is regular. Let \( p \) be the pumping length given by the pumping lemma. Choose \( s \) to be the string \( a^p b a^p \). Because \( s \) is a member of \( L_2 \) and \( s \) has length more than \( p \), the pumping lemma guarantees that \( s \) can be split into three pieces, \( s = xyz \), where for any \( i \geq 0 \), the string \( xy^i z \) is in \( L_2 \).

There is only one case, where \( y \) is entirely contained in the first \( p \) \( a \)'s. In this case, the string \( xy^2 z \) will not have an equal number of \( a \)'s on both sides of the \( b \), leading to a contradiction.

Therefore, \( L_2 \) is not regular.

Commentary from Megan:

Let’s say we have an FSA that recognizes palindromes. How many states does it have? Maybe it also has 4 states!

Fine. So let’s set a pumping length of 4. Are there any strings in \( L_2 \) that are longer than 4?

Duh again.

- 01010
- abcba
- 00100

Let’s take that last one 00100. Since we set a pumping length of 4 on a 4-state machine, once we hit the 5th character of this input string, we’re guaranteed to have hit one of the states twice. Which means that if \( L_2 \) is regular, there must be at least one non-zero substring, \( y \) (2), in the first 4 characters (3) that we could either omit, or repeat an infinite number of times (1) and the resulting string would still be in the language.

Easy again. We’ll pick the first 1 again. So:

\[
\begin{align*}
x &= 00 \\
y &= 1
\end{align*}
\]
\[ z = 00 \]

We can pump that 1 zero or more times and we’ll still have a palindrome. So far so good. But what if we pick longer strings:

- 010101010101010
- abcdefghgfedcba
- 000000010000000

If we take that last string again, we have 000000010000000 being recognized in 4 states. Now we can only assign \( x \) and \( y \) within that first set of zeroes:

\[
\begin{align*}
  x &= 000 \\
  y &= 0 \\
  z &= 0001000000
\end{align*}
\]

This time, if we omit or repeat our \( y \), we’ll be changing the number of 0’s on one side of our palindrome. The result will be a string that our FSA will accept, but that is not a palindrome.

Uh oh.

Well what if we had chosen a longer pumping length?

No matter what pumping length, \( p \), you choose, you will always be able to repeat this contradiction with a palindrome of length \( 2p \). Thus \( L_2 \) is not regular.
Problem 3

Prove that the following language is not regular:

$L_3 = \{ 1^k w \mid w \in \{0, 1\}^* \text{ and } w \text{ contains at most } k \text{ 1's, for } k \geq 1 \}$

Prove: $L_3$ is not regular

Proof Idea: We will assume that the language is regular and show that it violates the pumping lemma, leading to the conclusion that it cannot be regular.

Proof (by Contradiction):

Assume $L_3$ is regular. Let $p$ be the pumping length given by the pumping lemma. Choose $s$ to be the string $1^p01^p$. Because $s$ is a member of $L_3$ and $s$ has length more than $p$, the pumping lemma guarantees that $s$ can be split into three pieces, $s = xyz$, where for any $i \geq 0$, the string $xy^iz$ is in $L_3$.

There is only one case, where $y$ is entirely contained in the first $p$ 1’s. In this case, the string $xz$ (where $i = 0$) will not have an equal number of 1’s on both sides of the 0, leading to a contradiction.

Therefore, $L_3$ is not regular.