Consider a derivation of $w$. Each application of a rule of the form $A \to BC$ increases the length of the string by 1. So we have $n - 1$ steps here. Besides that, we need exactly $n$ applications of terminal rules $A \to a$ to convert the variables into terminals. Therefore, exactly $2n - 1$ steps are required.

Assume $G$ generates a string $w$ using a derivation with at least $2^b$ steps. Let $n$ be the length of $w$. By the results of Problem 2.26, $n \geq 2^{b+1} > 2^{b-1}$.

Consider a parse tree of $w$. The right-hand side of each rule contains at most two variables, so each node of the parse tree has at most two children. Additionally, the length of $w$ is at least $2^b$, so the parse tree of $w$ must have height at least $b + 1$ to generate a string of length at least $2^b$. Hence, the tree contains a path with at least $b + 1$ variables, and therefore some variable is repeated on that path. Using a surgery on tree argument identical to the one used in the proof of the CFL pumping lemma, we can now divide $w$ into pieces $uxvxyz$ where $uv^i x y^i z \in G$ for all $i \geq 0$. Therefore, $L(G)$ is infinite.
Problem 5
(a) \(L = \{0^i 1^j | i, j \geq 0 \text{ and } i \neq j\}\)

Idea: \(L = \{0^j 1^i | i, j \geq 0 \land (i < j \lor i > j)\}\)

\(S(q_0, 0, \varepsilon)\) includes \((q_0, 0)\) \(\Rightarrow\) (up to remember a prefix \(0^n\) of all \(0^n\) in the stack)

\(S(q_0, \varepsilon, \varepsilon)\) includes \((q_1, \varepsilon)\) nondeterministically after \(q_1\) to guess \(n\) and \(0^n\) prefix \(0^n\)

\(S(q_1, 1, 0)\) includes \((q_2, \varepsilon)\) \(q_1\): first to chek off

\(S(q_1, \varepsilon, 0)\) includes \((q_2, \varepsilon)\) \(S(q_1, 1, \varepsilon)\) includes \((q_3, \varepsilon)\)

\(S(q_2, \varepsilon, 0)\) includes \((q_2, \varepsilon)\) \(S(q_3, 1, \varepsilon)\) includes \((q_3, \varepsilon)\)

Empty the stack (not necessary)

\(\text{(All other combinations, } S\text{-value} = \emptyset)\)

Set \(\emptyset\) accepting states: \(\{q_2, q_3\}\)
Problem 5 (b)

L = \{ x \in \{a, b\}^* \mid x^5 \neq z \}

Typical string x \in L:

\[
\begin{align*}
\text{There exist a pair (say, first pair) of corresponding symbols} & \quad c \text{ and } c' \text{ with } c \neq c' \\
\text{once such a pair is detected} & \quad \text{the middle substring can be any string in } \{a, b\}^* \\
\text{So (q, start, } & \quad \text{includes (q, Zo)} \quad \text{place bottom circle marker Zo} \\
& \quad \text{for matching prefix (u)} \\
& \quad \text{with me suffix in reversed (ur))} \\
& \quad \text{mark q to remember the prefix "u";}
\end{align*}
\]

\[
\begin{align*}
& \quad \text{I remark prefix} \\
& \quad \text{calls for prefix (q, A)} \\
& \quad \text{remember prefix} \\
& \quad \text{calls for prefix (q, B)} \\
& \quad \text{for all c \in \{a, b\}:} \\
& \quad \text{so (q, remark, c, Z) includes (q, c, middle, \varepsilon)} \\
& \quad \text{non-deterministically guess the encounters symbol c} \\
& \quad \text{remember c in state q, middle, and enter} \\
& \quad \text{the loop q, middle to non-deterministically consume} \\
& \quad \text{the "middle substring".}
\end{align*}
\]
Loop: \( c \text{, middle} \) nondeterministically consume one middle symbol.

\[ \forall d \in \{a, b, c\}, \]

\[ S(c, \text{middle}, d, \varepsilon) \text{ includes } (c, \text{middle}, \varepsilon) \]

\[ \forall c' \in \{a, b, c\} \text{ with } c' \neq c : \]

\[ S(c, \text{middle}, c', \varepsilon) \text{ includes } (c, \text{checkoff}, \varepsilon) \]

When encountering a symbol \( c' \neq c \) (remembered in \( c, \text{middle} \)) nondeterministically enter a check-off state (for the prefix "u" remembered in the stack).

Loop: \( c, \text{checkoff} \) to check off the suffix against the prefix remembered in the stack:

\[ S(c, \text{checkoff}, a, A) \text{ includes } (c, \text{checkoff}, \varepsilon) \]

\[ S(c, \text{checkoff}, b, B) \text{ includes } (c, \text{checkoff}, \varepsilon) \]

\[ S(c, \text{checkoff}, \varepsilon, Zo) \text{ includes } (c, \text{accept}, \varepsilon) \]

Nondeterministically guess the end of input - when one bottom-stack marker \( Zo \) is exposed - ready to accept.

Check:
- For non-trivial \( x \in L \), PDA accepts \( x \)
- For trivial \( x \in L \), PDA accepts \( x \)
- For \( x \notin L \), PDA does not accept \( x \)
Problem 6. Use Pumping Lemma on both parts to show their non-context-freeness.

(a) Suppose that \( L \) were CF.

Let \( n \geq 1 \) be the Pumping Lemma constant.

Consider the string \( z = 0^n 1^n 0^n \in L \) with \( |z| = 3n \geq n \).

Consider all possible \( u, v, w, x, y \in \Sigma^* \) such that
\[
\begin{align*}
&z = uvwx y \\
&v \neq \epsilon \\
&|vwx| \leq n \\
&w x \geq 1
\end{align*}
\]

Instead of using a "brute-force" case-analysis
(sliding \( u, v, w, x, y \) from left to right
within \( z \) subject to (1) above)
we consider major cases (each is merging cases from
the brute-force approach):

1. Case when either \( v \) or \( x \) contains at least one 1:

   The means that the substring
   \( uvwx \) can only contain "0s"
   (from one of the two sides
   (prefix \( 0^n \); suffix \( 0^n \))

   Then, we let \( i = 0 \), \( uv^iwx^iy = uvwy \).
   Observe that, in \( uvwy \), the number of decreases
   but one side of \( 0s \) remain the same,
   so \( uv^0wx^0y \notin L \).
2. Case when \( v \) and \( x \) must contain all Os

Then, we let \( i = 2 \), and observe that, \( \forall u \in 0^* \) such that \( u \cdot w \in L \).
Thus, for each case above, \( \exists x \in 0^* \) such that \( uuwwx \in L \).
By Pumping Lemma, \( L \) is not CF.

(b) \( L = \{ w w^r w \mid w \in \{0,1,3\}^* \} \).

We have learned earlier that the language \( \{ w w \mid w \in \{0,1,3\}^* \} \) is not CF, so the language \( \{ w w^r \mid w \in \{0,1,3\}^* \} \) is CF.
So, the "last copy of \( w \)" in the membership of \( L \) is crucial.

Suppose that \( L \) were CF.
Let \( n \geq 1 \) be the Pumping Lemma constant.
Consider the string \( z = 0^n 1^n 0^n 1^n 0^n 1^n \) \( \in L \)
with \( 12l = 6n + 3 \geq n \).
Consider all possible $u, v, w, x, y \in \{0, 1\}^*$ such that:

\[
\begin{align*}
& z = uvwxyn, \\
& |uvwx| \leq n, \\
& |vwx| \geq 1
\end{align*}
\]  

Instead of following a "brute-force" case-analysis, we consider the following major cases.

First, as $z = \circ^n \circ^m \circ^n \circ^n \circ^n$

and $|uvwx| \leq n,$

no sub-string $uvwx$ can contain at most one 1.

1. Case when exactly one of $v$ or $x$ contains this 1:

Then, let $i = 2,$ and observe that, in $u^2vwx^2y,$ the number of 1s is not a multiple of 3, hence, $u^2vwx^2y$ is not a multiple of $3.$ Hence, $u^2vwx^2y$ is not a multiple of $u^2vwx^2y$ for any $w \in \{0, 1\}^*.$

i.e., $u^2vwx^2y \notin L.$

2. Case when $v$ and $x$ must consist of entirely 0s:

Now, we consider that $i = 2,$ with the string $u^2vwx^2y.$

Since each segment of Os is of length at least $n,$ pumping $v$ and $x$ can change the lengths (number of Os) of at most two adjacent segments of Os. Also, since $|vwx| \geq 1,$ this pumping must change the length of at least one segment of Os.
But how do we argue that $uv^2wx^y \in L$?

Observe that the positions of the 1s ($uv^2wx^y$) dictate how to view the string $uv^2wx^y$ in the desired form $\alpha\alpha\alpha$ for some $\alpha \leq 0.13^*$. Consider the following subcases:

2.1. Case when both $v$ and $x$ are embedded in $1^\text{st}$ segment of $0$s: $\overline{0^n1^n0^n1^n0^n1^n}$

Then, $\overline{u}uv^2wx^y$, since $|ux| > 1$.

Length of $1^\text{st}$ segment of $0$s in $uv^2wx^y$\n\[ > \frac{1}{2} \cdot \text{length of } 2^\text{nd} \text{ segment of } 0\text{s in } uv^2wx^y\]

So $uv^2wx^y \in L$.

Similar argument for subcases when both $v$ and $x$ are embedded in $2^\text{nd}$ segment of $0$s, in $3^\text{rd}$ segment of $0$s, or in $4^\text{th}$ segment of $0$s.

2.2. Case when $v$ and $x$ are embedded in $1^\text{st}$ or $2^\text{nd}$ segments of $0$s, respectively:

$\overline{u}uv^2wx^y = \overline{0^n1^n0^n1^n0^n}$

Then, by comparing the $1^\text{st}$ segment and $2^\text{nd}$ segment values.
2.3 Case when \( v \) is \( \alpha \setminus x \) are embedded in 2nd or 3rd segments of \( 0 \)s:
Then, by considering the 1st or 4th segments,
as \( |v_x| \geq 1 \), we have \( u v^2 w x^2 y \notin L \).

2.4 Case when \( v \) is \( \alpha \setminus x \) are embedded in 3rd or 4th segments of \( 0 \)s:
Then, by considering the 1st or 2nd segments,
as \( |v_x| \geq 1 \), we have \( u v^2 w x^2 y \notin L \).

For each case above, we obtain that
\[ u v^i w x^2 y \notin L \text{ for some } i \geq 0. \]

By Pumping Lemma, \( L \) is not CF.