5 1. Write a Context Free Grammar for the language L, where
   \[ L = \{ a^i b^j c^k \mid k = (i - j), \text{ if } i \geq j, \text{ else } k = 0 \} \]. Hint: Splitting into two cases makes your job easier.

   \[
   \begin{align*}
   S & \rightarrow A | C \\
   A & \rightarrow aAc | B \\
   B & \rightarrow aBb | \lambda \\
   C & \rightarrow aCb | cb \quad ? \quad j > i
   \end{align*}
   \]

3 2. Let \( L_1, L_2 \) be Non-Regular CFLs; \( R_1, R_2 \) be Regular; Answer is about \( S \) and there should be just one cell per row that has an \( X \).

<table>
<thead>
<tr>
<th>Definition of ( S ) / Characterization of ( S )</th>
<th>Always Regular</th>
<th>At worst CFL</th>
<th>Might not be CFL</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S = L_1 \cap L_2 )</td>
<td></td>
<td>( X )</td>
<td></td>
</tr>
<tr>
<td>( S = R_1 - L_1 )</td>
<td>( X )</td>
<td>( X )</td>
<td></td>
</tr>
<tr>
<td>( S = R_1 - R_2 )</td>
<td></td>
<td>( X )</td>
<td></td>
</tr>
<tr>
<td>( S \supseteq R_1 )</td>
<td>( X )</td>
<td>( X )</td>
<td></td>
</tr>
<tr>
<td>( S \subseteq R_1 )</td>
<td></td>
<td>( X )</td>
<td></td>
</tr>
<tr>
<td>( S = L_1 \cap R_1 )</td>
<td></td>
<td>( X )</td>
<td></td>
</tr>
</tbody>
</table>

2 3. Which of the following are correct definitions of an ambiguous grammar? In each case, \( w \) is a terminal string. Write \( T(\text{true}) \) or \( F(\text{false}) \) in the underlined area following each statement.

   - There are two distinct parse trees for some string \( w \) derived by the grammar \( F \)
   - There are two distinct derivations of some string \( w \) derived by the grammar \( T \)
   - There are two distinct rightmost derivations of some string \( w \) derived by the grammar \( T \)
   - There are two distinct leftmost derivations of some string \( w \) derived by the grammar \( T \)

4 4. Show that Context-Free Languages are closed under Non-Empty-Left-Right Quotient with Regular Languages. Non-Empty-Left-Right Quotient of a CFL \( L \) and a Regular Language \( R \), both of which are over the alphabet \( \Sigma \), is denoted \( \text{NELRQ}(L, R) \), and defined as

   \[
   \text{NELRQ}(L, R) = \{ y \mid xyz \in L; x, z \in R; \text{ and } y \neq \lambda \}.
   \]

   That is, we select a non-empty substring \( y \) of \( xyz \) in \( L \), provided \( x \) and \( z \) are both in the Regular Language \( R \).

   You may assume substitution \( f(a) = \{ a, a' \} \), and homomorphisms \( g(a) = a' \) and \( h(a) = a, h(a') = \lambda \). Here \( a \in \Sigma \) and \( a' \) is a distinct new character associated with each \( a \in \Sigma \). No justification is required.

   \[
   \text{NELRQ}(L, R) = \left( f(L) \cap g(R) \right) \Sigma^{+} g(R)
   \]
10 5. Consider some language $L$. For each of (a) and (b), and for each of the three possible complexities of $L$, indicate whether this is possible (Y or N) and present evidence. Recall that

$max(A) = \{ w \mid w \in A \text{ and for no } x \neq \lambda \text{ does } wx \in A \}$

If you answer Y, you must provide an example language $A$ and the resulting $L$. In the case of part (b) you must also present a homomorphism $\sigma$. If you answer N, state some known closure property that reflects a bound on the complexity of $L$. Note: I did the first of each of the three parts for you.

a.) $L = max(A)$ where $A$ is context-free, not regular.

Can $L$ be Regular? Circle Y or N.

If yes, show $A$ and argue $max(A)$ is Regular; if no, why not?

YES. Let $A = \{ a^i b^j \mid i, j > 0 \text{ and } j > i \}$

$L = max(A) = \emptyset$, a regular set, as every string in $A$ can be extended with more $b$'s.

Can $L$ be a non-regular CFL? Circle Y or N.

If yes, show $A$ and argue $max(A)$ is a CFL; if no, why not?

$$A = \{ a^n b^n \mid n > 0 \}$$

$$L = max(A) = A = \{ a^n b^n \mid n > 0 \}$$

Can $L$ be more complex that a CFL? Circle Y or N.

If yes, show $A$ and argue $max(A)$ is not a CFL; if no, why not?

$$A = \{ a^i b^i c^k \mid k \leq i \text{ or } k \leq j \}$$

$$L = max(A) = \{ a^i b^i c^k \mid k = \max(i, j) \}$$

$L$ is known (proven) to be a CSL, non-CFL language.

b.) Let $\sigma$ be a homomorphism from $\Sigma$ into regular languages, such that, for each $a \in \Sigma$, $\sigma(a) = w_a$, for some string $w_a$. Let $A$ be a context free, non-regular language and let $L = \sigma(A)$.

Can $L$ be Regular? Circle Y or N.

If yes, show $A$ and $\sigma$, and argue $\sigma(A)$ is Regular; if no, why not?

YES. Let $A = \{ a^n b^n \mid n > 0 \}$

Define $\sigma(a) = \lambda$ and $\sigma(b) = \lambda$.

$L = \sigma(A) = \{ \lambda \}$, a regular set.

Can $L$ be a non-regular CFL? Circle Y or N.

If yes, show $A$ and $\sigma$, and argue $\sigma(A)$ is a CFL; if no, why not?

Let $A = \{ a^n b^n \mid n > 0 \}$

Define $r(a) = a$ and $r(b) = b$

$L = r(A) = A = \{ a^n b^n \mid n > 0 \}$ which is a CFL

Can $L$ be more complex that a CFL? Circle Y or N.

If yes, show $A$ and $\sigma$, and argue $\sigma(A)$ is not a CFL; if no, why not?

CFLs are known to be closed under substitution and homomorphism.
6. Use the Pumping Lemma for CFLs to show that the following language \( L \) is not Context Free.
\[
L = \{ a^n b^{\text{sum}(1..n)} \mid n > 0 \}
\]
Here \( \text{sum}(1..n) = \sum_1^n i \).
Be explicit as to why each case you analyze fails to be an instance of \( L \) and, of course, make sure your cases cover all possible circumstances. I have done the first two steps for you.

Assume \( L \) is Context Free

provides a whole number \( N > 0 \) that is the value associated with \( L \) based on the Pumping Lemma

Choose \( z = a^N b^{(N+1)N/2} \in L \)

\[
z = u^n w x y, \quad |nwxy| \leq N, \quad (nx) > 0 \quad \text{and} \quad
\]

\[
v \geq 0 \quad u^n w x y \in L
\]

\[\]

**CASE 1:** \( nwxy \) contains only \( b \)'s.

Set \( l = 2 \) then PL says
\[
a^N b^{(N+1)N/2 + (nx)} \in L
\]

But \( (nx) > 0 \) and so we have too many \( b \)'s, thus \( u^n w x y \notin L \)

**NOTE:** Can also use \( i = 0 \)

**CASE 2:** \( nwxy \) contains at least one \( a \)

Then \( nwxy \) contains at most \( N-1 \) \( b \)'s. Set \( i = 0 \)

In best case, we have \( N+1 \) \( a \)'s

and \( (N+1)N/2 + (N-1) b \).

But \( N+1 \) \( a \)'s requires \( (N+1)N/2 + (N+1) b \)

\( \text{AND} \quad (N+1)N/2 + (N-1) < (N+1)N/2 + (N+1) \)

So \( u^n w x y \notin L \)

As cases 1 and 2 cover all possibilities

we have a contraction proving \( u^n w x y \notin L \)

In all cases, hence \( L \) is not a CFL.
10 7. Present the CKY recognition matrix for the string **aababb** assuming the Chomsky Normal Form grammar, \( G = ( \{ S, T, U, V, W, A, B \}, \{ a, b \}, R, S \) ), specified by the rules \( R \):

```
S \to \ A T | B U
T \to \ b | B S | A V
U \to \ a | A S | B W
V \to \ T T
W \to \ U U
A \to \ a
B \to \ b
```

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>a</th>
<th>b</th>
<th>a</th>
<th>b</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UA</td>
<td>-UA</td>
<td>TB</td>
<td>UA</td>
<td>TB</td>
<td>TB</td>
</tr>
<tr>
<td>2</td>
<td>W</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>V</td>
</tr>
<tr>
<td>3</td>
<td>U</td>
<td>U</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>W</td>
<td>S</td>
<td>S</td>
<td></td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>U</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A little help

<table>
<thead>
<tr>
<th>Non-Terminal</th>
<th>First Symbol in Rules</th>
<th>Second Symbol in Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>None</td>
<td>T \to B S ; U \to A S</td>
</tr>
<tr>
<td>T</td>
<td>V \to T T</td>
<td>S \to A T ; V \to T T</td>
</tr>
<tr>
<td>U</td>
<td>W \to U U</td>
<td>S \to B U ; W \to U U</td>
</tr>
<tr>
<td>V</td>
<td>None</td>
<td>T \to A V</td>
</tr>
<tr>
<td>W</td>
<td>None</td>
<td>U \to B W</td>
</tr>
<tr>
<td>A</td>
<td>S \to A T ; T \to A V ; U \to A S</td>
<td>None</td>
</tr>
<tr>
<td>B</td>
<td>S \to B U ; T \to B S ; U \to B W</td>
<td>None</td>
</tr>
</tbody>
</table>

Is **aababb** in \( L(G) \)? Yes

What is the order of execution of this approach to determine if some \( w, |w| = N \), is in \( L \)? N^3

What is the algorithmic strategy, e.g., greedy, divide and conquer, dynamic programming, backtracking or randomized, associated with this CKY algorithm called? Dynamic Programming
Consider the CFG $G = (\{S, A, B\}, \{a, b\}, R, S)$ where $R$ is:

- $S \rightarrow SAb | AbBa$
- $A \rightarrow aS | a$
- $B \rightarrow bS | b$

In the PDAs below, you may (and are encouraged) to use a transition diagram where transitions have arcs with labels of form $a, \alpha \rightarrow \beta$ where $a \in \Sigma \cup \{\lambda\}, \alpha, \beta \in \Gamma^*$. Note: This just means that you can use extended stack operations that push or pop arbitrary length strings.

a.) Present a pushdown automaton that parses the language $L(G)$ using a top down strategy.

**INITIAL CONTENTS OF STACK = ___**

```
\[ q_0, w, S \quad \vdash \quad [q_0, \lambda, x] \]
```

b.) Now, using the notation of IDs (Instantaneous Descriptions, $[q, x, z]$), describe how your PDA in (a) accepts strings generated by $G$.

c.) Present a pushdown automaton that parses the language $L(G)$ using a bottom up strategy. Note: I am fine with your showing strings that are on top of the stack in either reversed or non-reversed form.

**INITIAL CONTENTS OF STACK = ___**

```
\[ q_0, \#, S \quad \vdash \quad [q_0, \lambda, \lambda] \]
```

d.) Now, using the notation of IDs (Instantaneous Descriptions, $[q, x, z]$), describe how your PDA in (c) accepts strings generated by $G$.

```
\[ \Sigma q_j, w_j, \# \quad \vdash \quad \Sigma S, \lambda, \lambda \]
```
3 a.) Consider the context-free grammar $G_1 = (\{ S, A, B \}, \{ 0, 1 \}, R_1, S)$, where $R_1$ is:

$S \rightarrow AB$
$A \rightarrow 0A0 | \lambda$
$B \rightarrow 1B1 | \lambda$

Remove all $\lambda$-rules, except possibly for a start symbol, creating an equivalent grammar $G_1'$. Show all rules.

$\text{Nullable} = \{S, A, B\}$

\[
\begin{align*}
S' & \rightarrow \lambda | A | B | AB \\
S & \rightarrow A | B | AB \\
A & \rightarrow 0A0 | 100 \\
B & \rightarrow 1B1 | 11
\end{align*}
\]

3 b.) Consider the context-free grammar $G_2 = (\{ S, A, B \}, \{ 0, 1 \}, R_2, S)$, where $R_2$ is

$S \rightarrow AB | B$
$A \rightarrow 1A0 | 10$
$B \rightarrow A | AA$

Remove all unit rules, creating an equivalent grammar $G_2'$. Show all rules.

$\text{Unit}(S) = \{ S, \epsilon, A \}; \text{Unit}(A) = \{ A \}; \text{Unit}(B) = \{ B, A \}$

\[
\begin{align*}
S & \rightarrow AB | 1A0 | 10 | AA \\
A & \rightarrow 1A0 | 10 \\
B & \rightarrow AA | 1A0 | 10
\end{align*}
\]
9. c.) Consider the context-free grammar \( G_3 = ( \{ S, A, B \}, \{ 0, 1 \}, R_3, S \) ), where \( R_3 \) is
\[
S \rightarrow AB \mid BB \\
A \rightarrow 1A0 \\
B \rightarrow 0B1 \mid 01
\]
Remove all non-productive non-terminals, creating an equivalent grammar \( G_3' \). Show all rules.
\[
Productive = \{ S, B \}; \ Unproductive = \{ A \}
\]
\[
S \rightarrow BB \\
B \rightarrow 0B1 \mid 01
\]

9. d.) Consider the reduced context-free grammar \( G_4 = ( \{ S, A, B \}, \{ 0, 1 \}, R_4, S \) ), where \( R_4 \) is
\[
S \rightarrow AABB \\
A \rightarrow 1B0 \mid 10 \\
B \rightarrow 0A1 \mid 01
\]
Convert an equivalent grammar \( G_4' \). Show all rules.
\[
S \rightarrow \langle AAB \rangle B \\
A \rightarrow \langle 1B \rangle \langle 0 \rangle \mid \langle 1 \rangle \langle 0 \rangle \\
B \rightarrow \langle OA \rangle \langle 1 \rangle \mid \langle 0 \rangle \langle 1 \rangle \\
\langle AAB \rangle \rightarrow \langle AA \rangle B \\
\langle AA \rangle \rightarrow AA \\
\langle 1B \rangle \rightarrow \langle 1 \rangle B \\
\langle OA \rangle \rightarrow \langle 0 \rangle A \\
\langle 0 \rangle \rightarrow 0 \\
\langle 1 \rangle \rightarrow 1
\]
\[
\text{OR} \quad S \rightarrow \langle AA \rangle \langle BB \rangle \\
\langle AA \rangle \rightarrow AA \\
\langle BB \rangle \rightarrow BB
\]