Q. No. 1  \[(2+8)*4 = 40\text{ points}\]
Clearly mention, whether following languages are regular or non-regular?
If regular, then draw a fully specified \textit{DFA}. If non-regular, then prove using \textit{pumping lemma}.

a) \[L_1 = \{ww \mid w \in \{a, b\}\}.\]
This language is non-regular.
\textbf{Proof:}
We use pumping lemma to prove that \(L_1\) is not regular. The proof is by contradiction. Assume to the contrary that \(L_1\) is regular. Let \(m\) be the pumping length given by pumping lemma. We choose \(w\), such that \(|w| \geq m\). \(w\) can be partitioned into three pieces:
\[w = xyz, \text{ such that } |xy| \leq m, |y| \geq 1\]
If \(L\) is regular and \(xyz \in L_1\) then \(xy^kz \in L_1\) where \(k \geq 0\).
We choose following \(w\):
\[w = vv \in L_1, v\text{ is a string over } \{a, b\} \text{ of length } m.\]
So, \(|w| = 2m \geq m\)
\(w\) is partitioned as follows:

- \(x\)
- \(y\)
- \(z\)

Here, length of \(y\) is greater that or equal to 1 and length of \(xy\) is less or equal to \(m\).
According to claim of pumping lemma for regular languages, \(xy^0z \in L_1\).
As \(y\) contains only more than one letters. So, by pumping it for \(k=0\), number of letters in first \(v\) will decrease at least by 1. Now, number of letters in first \(v\) will be \(m-1\) or less, whereas number of letters in second \(v\) remain \(m\). Such string can’t be part of \(L_1\). So, contradiction!!!
Hence, our assumption was false, and \(L_1\) is non-regular.

b) \[L_2 = \{a^{2^n} \mid n \geq 0\}\]
This language is non-regular.
\textbf{Proof:}
We use pumping lemma to prove that \(L_2\) is not regular. The proof is by contradiction. Assume to the contrary that \(L_2\) is regular. Let \(m\) be the pumping length given by pumping lemma. We choose \(w\), such that \(|w| \geq m\). \(w\) can be partitioned into three pieces:
\[w = xyz, \text{ such that } |xy| \leq m, |y| \geq 1\]
If \(L\) is regular and \(xyz \in L_2\) then \(xy^kz \in L_2\) where \(k \geq 0\).
We choose following \(w\):
\( w = a^{2m} \in L \)

So, \(|w| = 2^m > m\)

\( w \) is partitioned as follows:

\[
\begin{array}{c}
\underbrace{\text{aaa} \ldots \text{aaa}}_{m} \\
\underbrace{\text{aaa} \ldots \text{aaa}}_{2^m-m} \\
\end{array}
\]

Here, length of \( y \) is greater than or equal to 1 and length of \( xy \) is less or equal to \( m \).

According to claim of pumping lemma for regular languages, \( xy^2z \in L \).

As \( y \) contains only a’s. So, by pumping it for \( k=2 \), number of a’s in \( xy^2z \) will increase and become \( 2^m+p \), where \( p \) is length of \( y \) and \( 1 \leq p \leq m \). So,

\[
2^m+p \leq 2^m+m < 2^m+2^m
= 2.2^m = 2^{m+1}
\]

So, \( 2^m+p < 2^{m+1} \)

So, \( 2^m+p \) can’t be power of 2 and hence string of this length can’t be part of this language. Thus, contradiction!!!

Hence, our assumption was false, and \( L_2 \) is non-regular.

c) \( L_3 = \) The complement of \( \{0^n1^n \mid n \geq 0\} \)

This language is not regular.

**Proof:**

\( \{0^n1^n \mid n \geq 0\} \) is a non-regular language. Prove it after referring textbook or lecture slides.

Complement of an non-regular language can’t be regular.

Suppose that complement of a non-regular language is regular. But we know that complement of regular language is regular. So, complement of complement will also be regular. Hence, our assumption is incorrect, and complement of non-regular language is non-regular. So, \( L_3 \) is non-regular.
d) \( L_4 = \{ w \mid w \in \{0, 1\}^* \text{ and when } w \text{ is interpreted in reverse as a binary integer, is a multiple of 5} \} \)

This language is regular.
Following is the DFA.
Q. No. 2  [10 points]
Here is a “proof”, using the pumping lemma, that the language L of all strings of 0’s and 1’s of length 100 is not regular. Since the result being “proved” is false (all finite languages are regular), the proof cannot be correct. What is the flaw in the proof? After removing flaw, can you prove using pumping lemma, that L is regular? If yes, then please give modified correct proof?

“Assume for the sake of contradiction that L is regular. By the pumping lemma, if we choose an element of L, say w = 0100, there are strings x, y and z, with |y| > 0, so that every string of the form xy^kz (where k ≥ 0) is in L. Since there are infinitely many different strings of this form, this contradicts the fact that L is finite. Therefore, L is not regular.”

Flaws:
- No consideration of an arbitrary integer m, which is provided by pumping lemma
- Invalid selection of w
- Pumping lemma can’t be applied to finite languages

Q. No. 3  [10 points]
Find two regular languages L1 and L2 over {a,b}, where L1 and L2 are not equals and neither is a subset of the other, and \((L1 \cup L2)^* = L1^* \cup L2^*\).

L1 = {\text{lambda}}
L2 = {a}

Or there may be many more………..

Q. No. 4  [20+10+10=40 points]
a) Suppose that p and q are distinguishable states of a given DFA A with n states. As a function of n, what is the tightest upper bound on how long the shortest string that distinguishes p from q can be?

Tightest upper bound is \(n-2\).
It is length of the distinguishing string when first two states are p and q and nth state is final state. In all other possibilities, this length is less or equal to n-2.

b) Draw the table of distinguishabilities for the following automaton? Also construct the minimum-state equivalent DFA? (Arrow represents start state and * represents final state)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B E</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>C F</td>
<td></td>
</tr>
<tr>
<td>*C</td>
<td>D H</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>E H</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>F I</td>
<td></td>
</tr>
<tr>
<td>*F</td>
<td>G B</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>H B</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>I C</td>
<td></td>
</tr>
<tr>
<td>*I</td>
<td>A E</td>
<td></td>
</tr>
</tbody>
</table>
### Table of Distinguishabilities

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

Equivalence classes:

\[
\{\{A,D,G\}, \{B,E,H\}, \{C,F,I\}\}
\]

Minimized DFA:

![Minimized DFA Diagram]