Question 1(a)

[2.5 points] Find the regular expression for strings over the alphabet {a, b} which start with 'a', end with 'ab', and are of even length. For example, **aaab**, **ababab**, **abbbab**, **aaaaab**, etc.

Question 1(b)

[2.5 points] You are about to open an account in Fakebook, which is of course a social media website. You have completed the sign-up form by entering the necessary information. But the form is failing to be submitted due to password restrictions. It says that the password has to have a length of at least 5. Also, it may consist of any alphanumeric character (alphabets can be of both uppercase and lowercase), but it must have at least one uppercase alphabet. For example, ab34Hp, 12Kxy99, j7P8G5mn, etc. Write a regular expression for such passwords.

Question 2

[5 points] Draw a DFA that accepts ternary numbers divisible by 4 where Σ = {0,1,2}. Some inputs that will be accepted are 22,121,202, etc.

Turnery numbers are numbers written in base-3. For example, the first few ternary numbers are 0, 1, 2, 10, 11, 12, 20, 21, 22, 100, 101, 102, 110, 111, 112, etc. 10 in ternary is 3 in decimal, 11 in ternary is 4 in decimal, and so on.

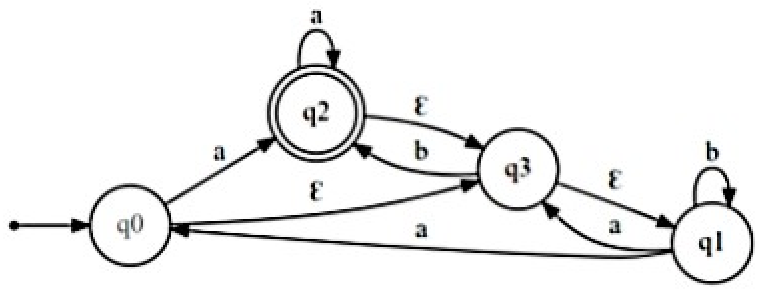
Question 3

[5 points] Convert the following RE to an NFA: 0*111* + (0+1*1)*

N.B.: You won't have to follow any specific method or show it step by step. Here, '+' stands for the 'OR' operator.

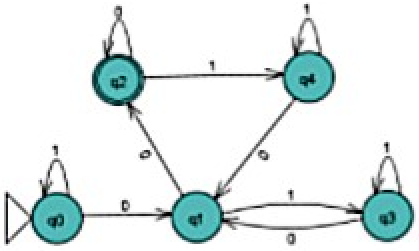
Question 4

[5 points] Convert the following Ɛ-NFA into a DFA (Compute Ɛ-closures and use the subset construction method. Do not try to answer by guessing).



Question 5

[3 + 2 = 5 points]Minimize the following DFA using Hopcroft's algorithm. **Show** the step-by-step partitioning and **draw** the minimized DFA. Here, q_0 is the starting state.



Question 6

[5 points] The transition table of a DFA is given in Fig. 1.

State/Σ	0	1
→A	B	C
B	B	D
C	B	E
D	C	F
E	F	F
F*	F	F

Here, **A** is the Start State and **F** is the Final State. Convert this DFA to its equivalent Regular Expression using the State Elimination method in the following order:

Eliminate State **C** first, then **E**, then **D**, and then **B**.

Do not add any new Start or new Final State.
Write the equivalent regular expression. Show every step as clearly as possible.

Ans. to Q. No - 5

$$\Pi_1 = \underbrace{(q_0, q_1, q_3, q_4)}_{G_1} \underbrace{(q_2)}_{G_2}$$ [Seperated the final and non-final states]

Now>

$q_0 \xrightarrow{0} G_1$

$q_1 \xrightarrow{0} \boxed{G_2}$

$q_3 \xrightarrow{0} G_1$

$q_4 \xrightarrow{0} G_1$

$q_0 \xrightarrow{1} G_1$

$q_1 \xrightarrow{1} G_1$

$q_3 \xrightarrow{1} G_1$

$q_4 \xrightarrow{1} G_1$

So, we will be seperating q_1.

$$\therefore \Pi_2 = \underbrace{(q_0, q_3, q_4)}_{G_1} \underbrace{(q_1)}_{G_2} \underbrace{(q_2)}_{G_3}$$

$q_0 \xrightarrow{0} G_2$

$q_3 \xrightarrow{0} G_2$

$q_4 \xrightarrow{0} G_2$

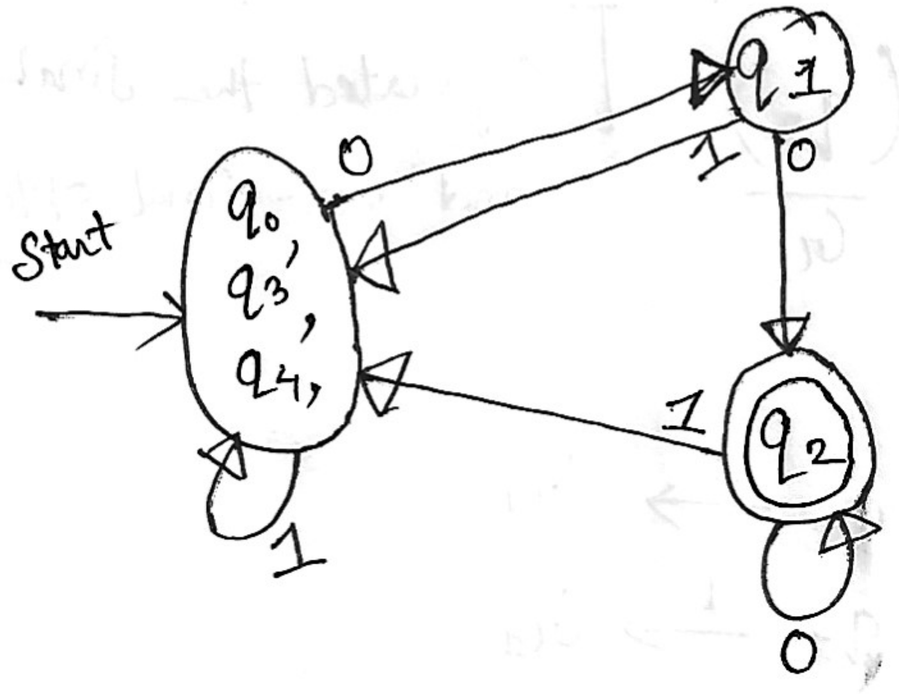
$q_0 \xrightarrow{1} G_1$

$q_3 \xrightarrow{1} G_1$

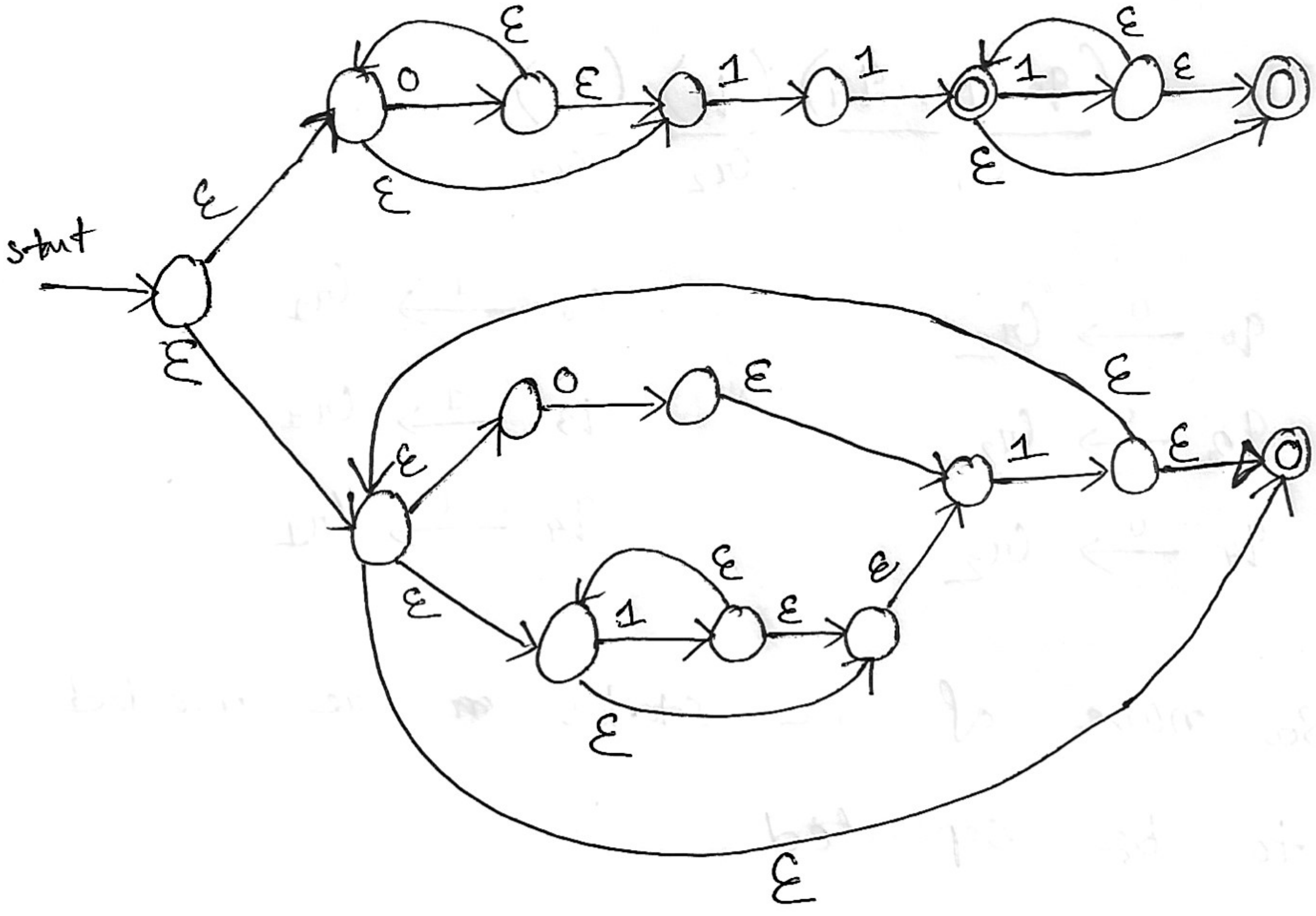
$q_4 \xrightarrow{1} G_1$

So, none of the states are needed to be seperated.

∴ final DFA (Minimized)

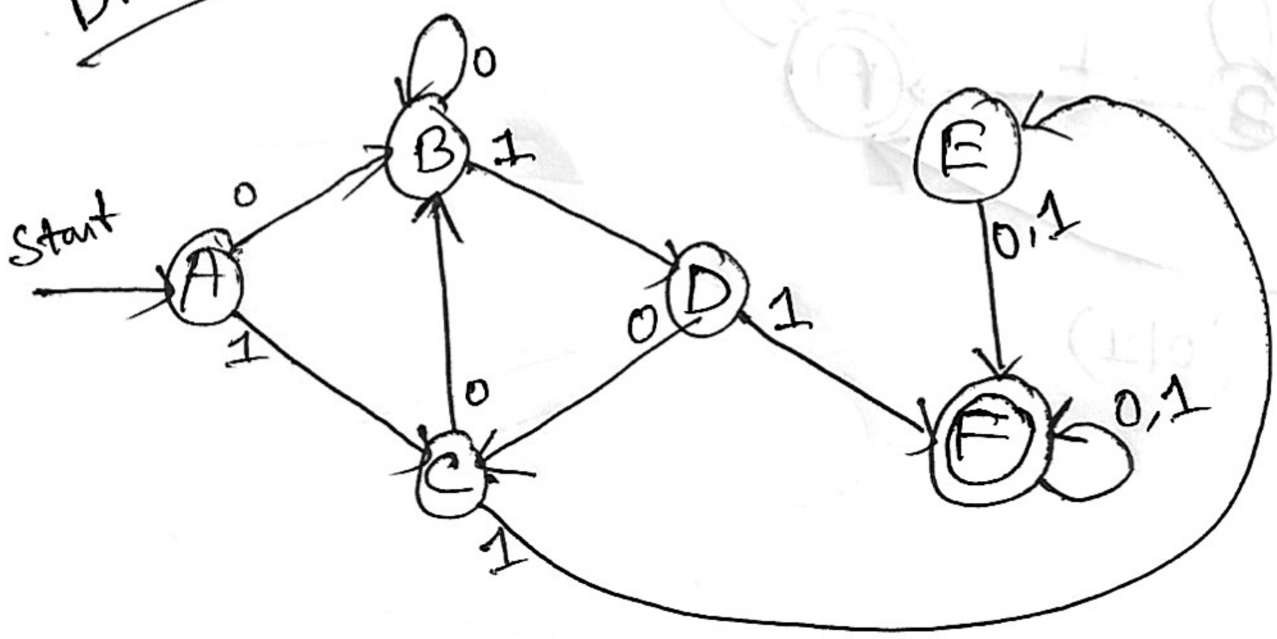


Ans. to Q. No – 3



Ans. to Q.No – 6

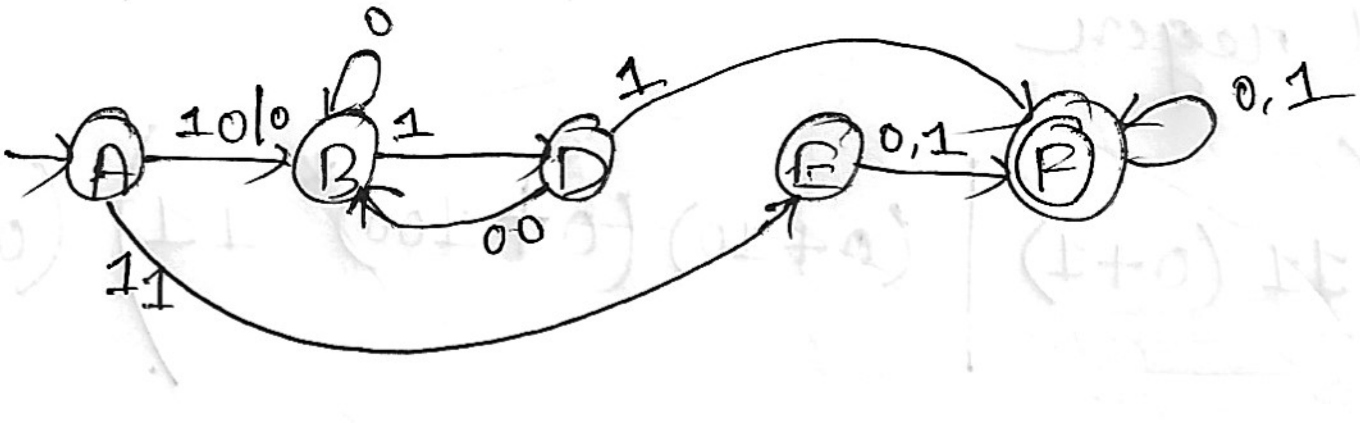
DFA:



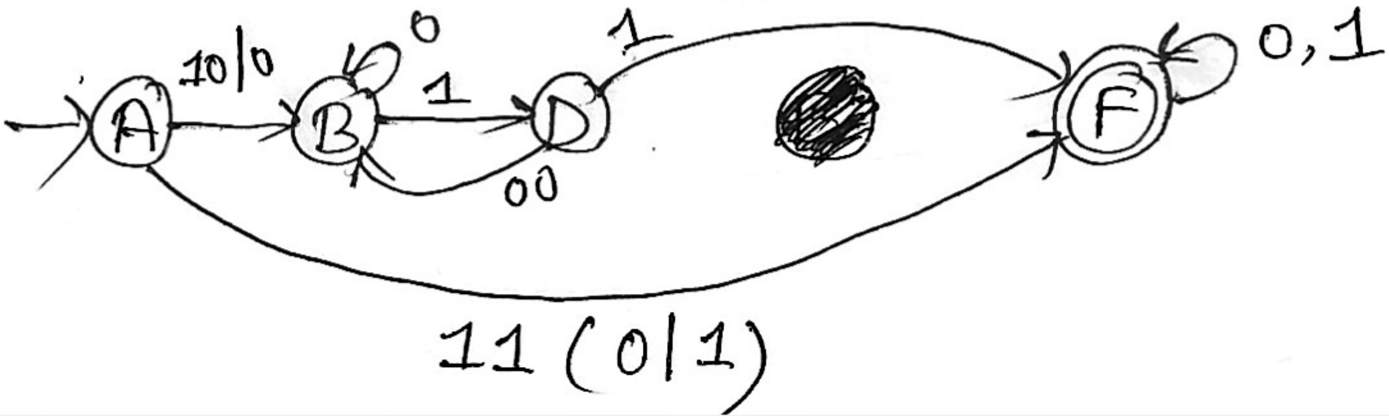
Now, After eliminating C, we get ~~Regex~~ :

$A \xrightarrow{10} B$ $\quad$ $D \xrightarrow{00} B$

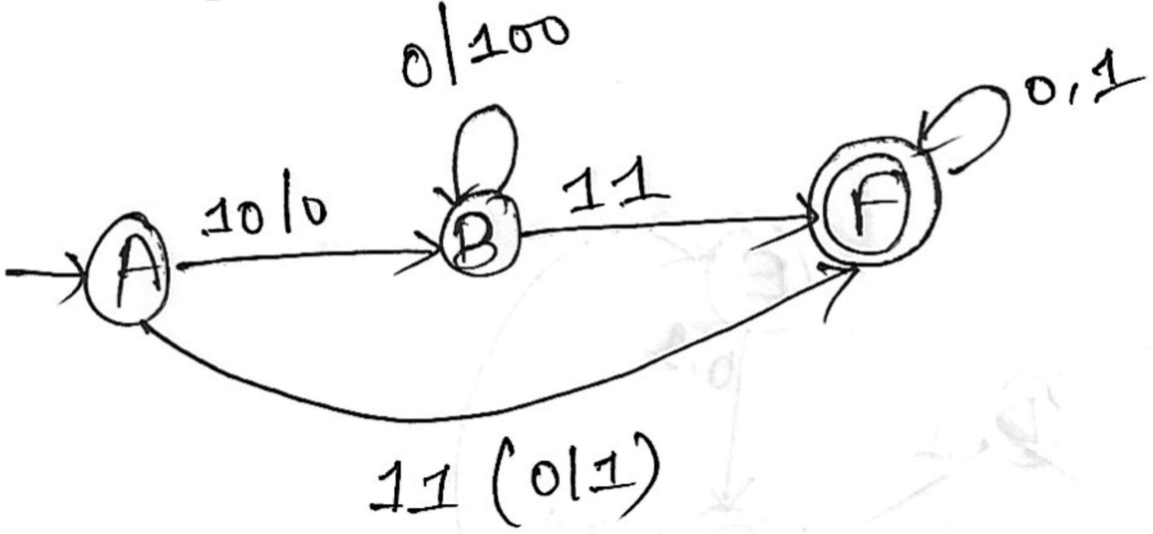
$A \xrightarrow{11} E$ $\quad$ $D \xrightarrow{01} E$



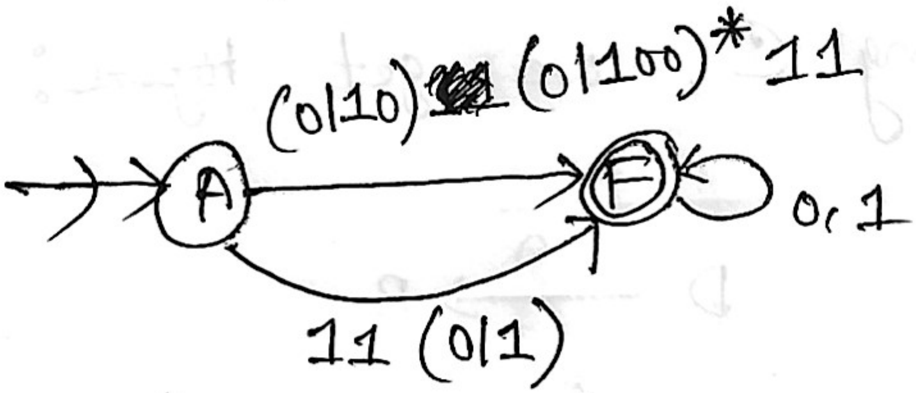
Eliminating E we get,



Eliminating D, we get,



Eliminating B we get-



∴ Final regex

$$= \left(11\,(0+1) \;\middle|\; (0+10)\,(0+100)^{*}\,11 \right) (0|1)^{*}$$

Ans. to Q.No – 4

Start state = $\{q_0, q_1, q_3\} = 1$

$\text{Move}_{DFA}(1, a) = \{q_0, q_1, q_2, q_3\} = 2$

" $(1, b) = \{q_1, q_2, q_3\} = 3$

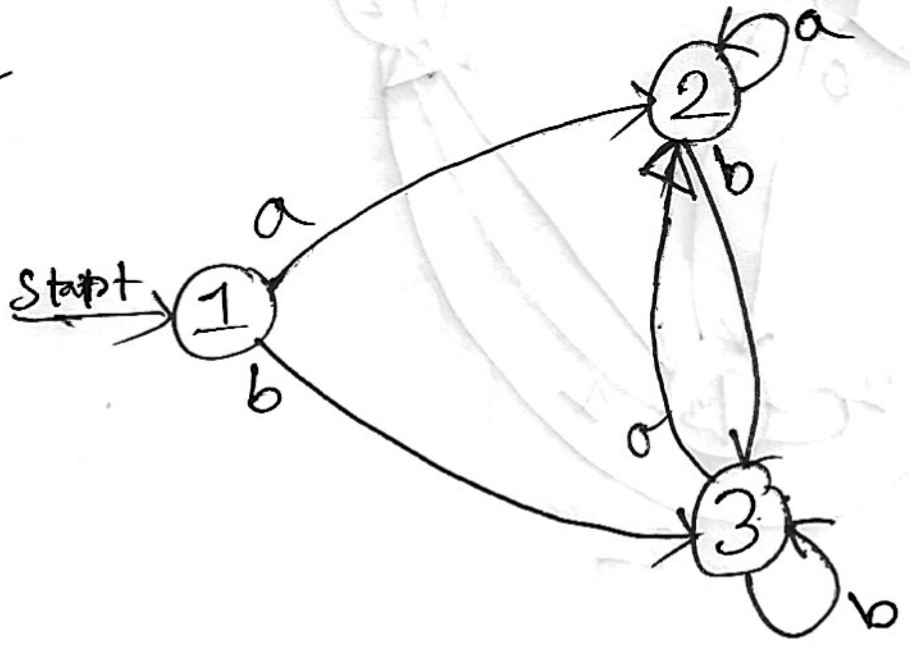
" $(2, a) = \{q_0, q_1, q_2, q_3\} = 2$

" $(2, b) = \{q_1, q_2, q_3\} = 3$

" $(3, a) = \{q_0, q_1, q_2, q_3\} = 2$

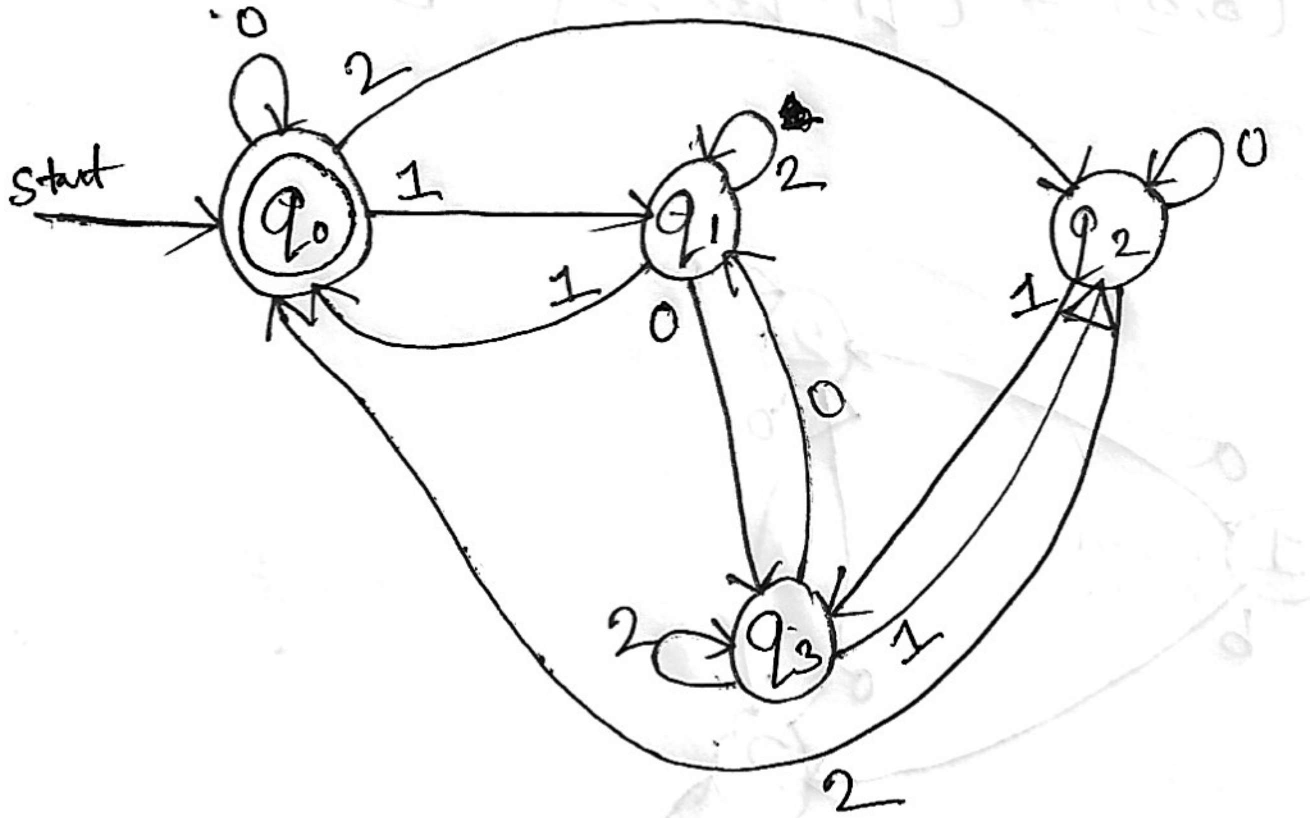
" $(3, b) = \{q_1, q_2, q_3\} = 3$

DFA:



Ans. to Q No – 2

	0	1	2
Start → q_0	q_0	q_1	q_2
q_1	q_3	q_0	q_1
q_2	q_2	q_3	q_0
q_3	q_1	q_2	q_3



Ans to Q No - 1

a

Regex = $a\left(\left(a|b\right)\left(\left(a|b\right)\left(a|b\right)\right)^{*}\right)ab$

b

Let use define a character class

Let, $(A\text{-}Za\text{-}z0\text{-}9) = C$

∴ Regex

$= (C)^{*} \, (A\text{-}Z) \cdot CCCC(C^{*})$