Assignment
Module-II

Shivang Shrivastava
Roll no. 60

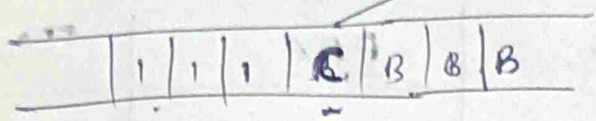
Q.1
Solve.

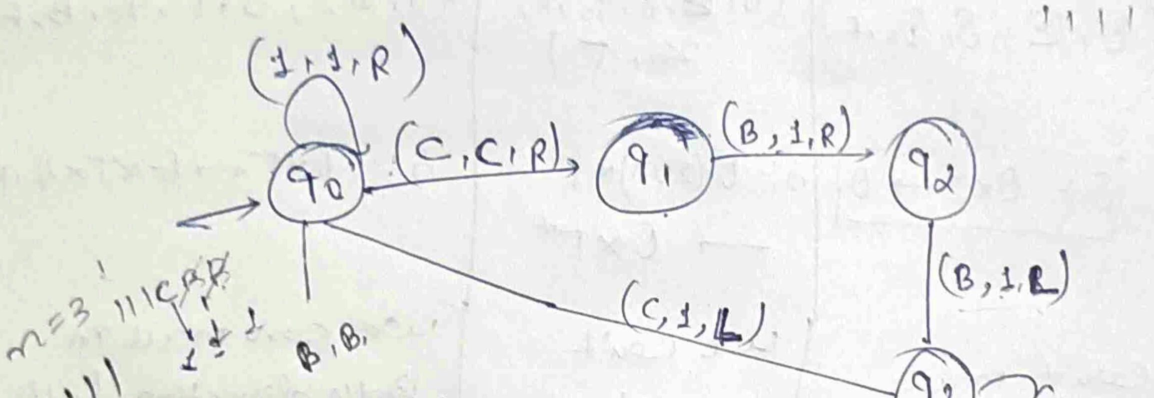
	DFA	PDA	TM
(Tuple) ~~Transition function~~ :-	$(Q, \Sigma, \delta, q_0, F)$	$(Q, \Sigma, \delta, q_0, F, Z_0, \Gamma)$	$\{Q, \Sigma, \delta, \Gamma, q_0, B, F\}$
→ Transition function:	$\boxed{\delta :- Q \times \Sigma \rightarrow Q}$	$\delta :- Q \times (\Sigma \cup \lambda) \times \Gamma \rightarrow Q \times \Gamma^*$	$\delta :- Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
⇒	we can't move Both direction.	we can't move both direction.	we can move in Both direction left as well as Right.
→	we can't write in tape.	we ~~can~~ use stack for writing purpose.	we can write in the tape.
→	It is not possible to solve complex problem.	May be solve or not solve.	→ ~~Ag~~ Any human computable problem can be solve by using TM.
Application :-	→ used in text editor. → Analysis of compiler.	Tower of hanoi problem use.	Implement In Artificial Intelligence.

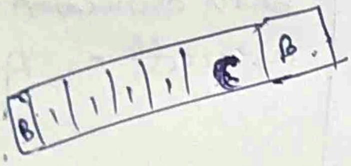
Q.2 Design a TM for $f(n) = 2n$

$n = 3$

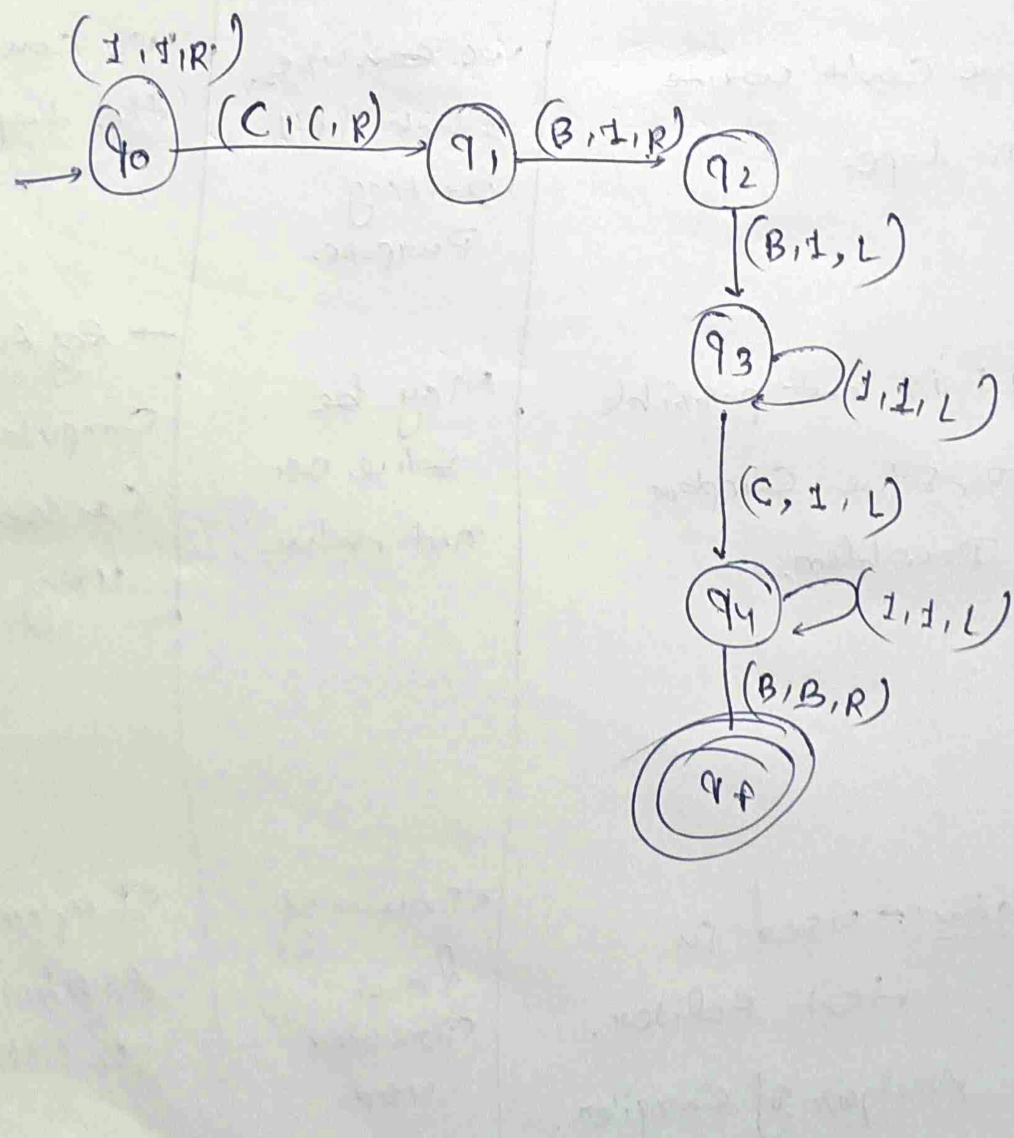
C is a seperator



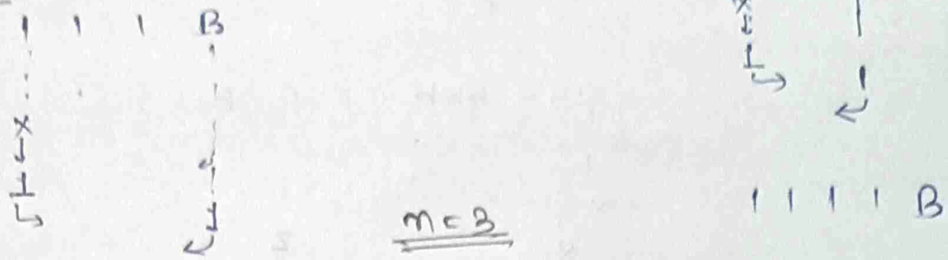




(4)

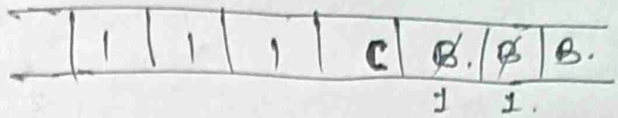


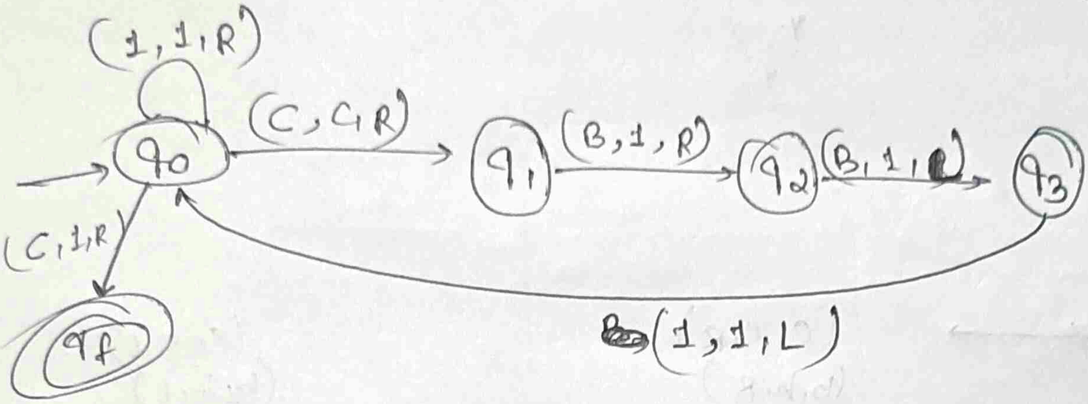
Shivang (60)



$f(n) = 2n$.

$n = 3$

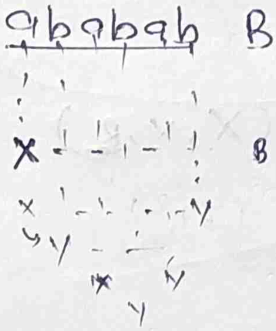




Q.3. Design a TM for $(ab)^+$

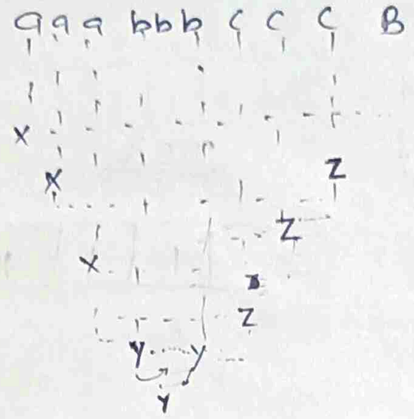
Soln.

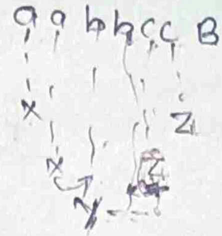
ababab B

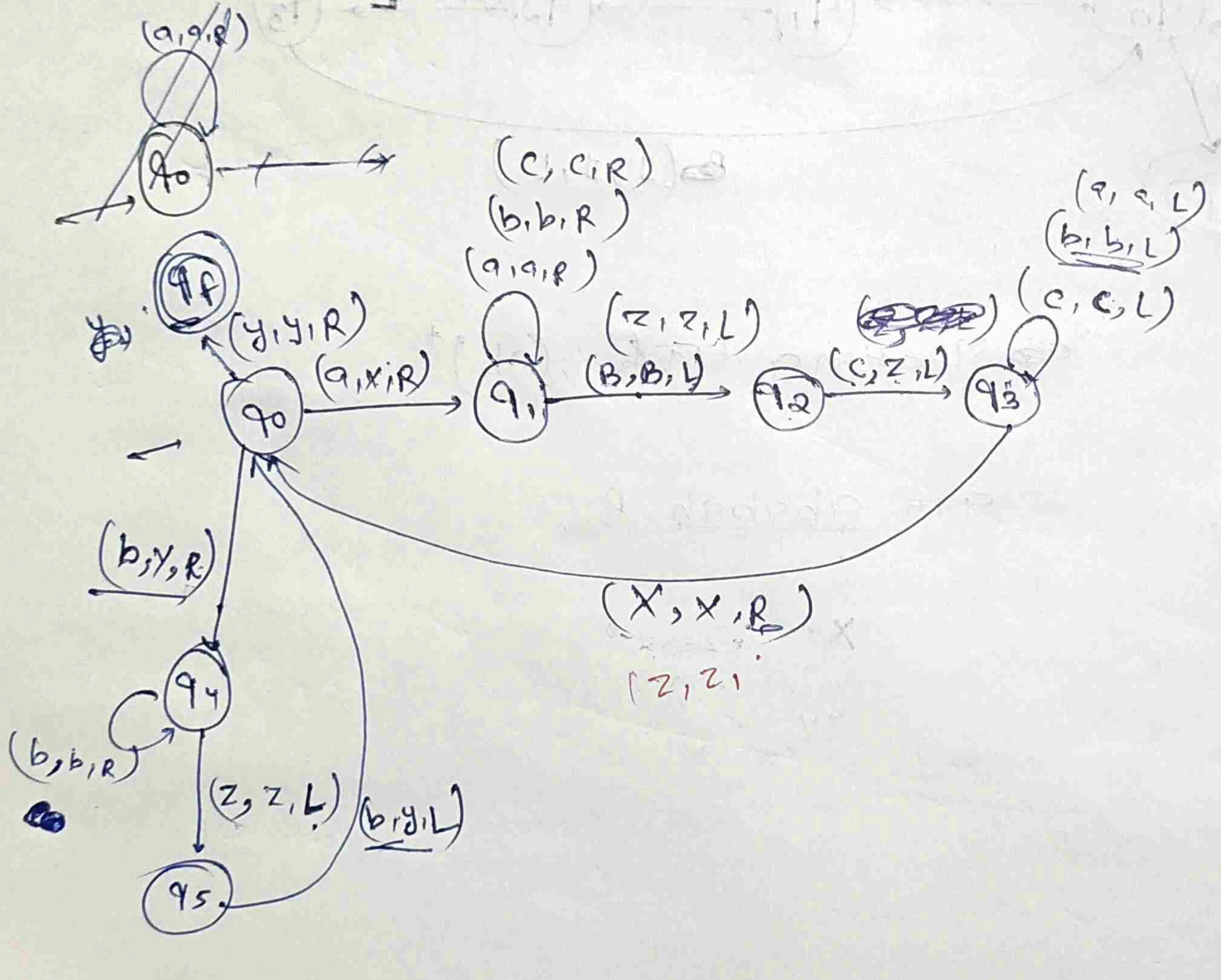


Shivang (60)

(5) $a^m b^n c^m \mid m>0$



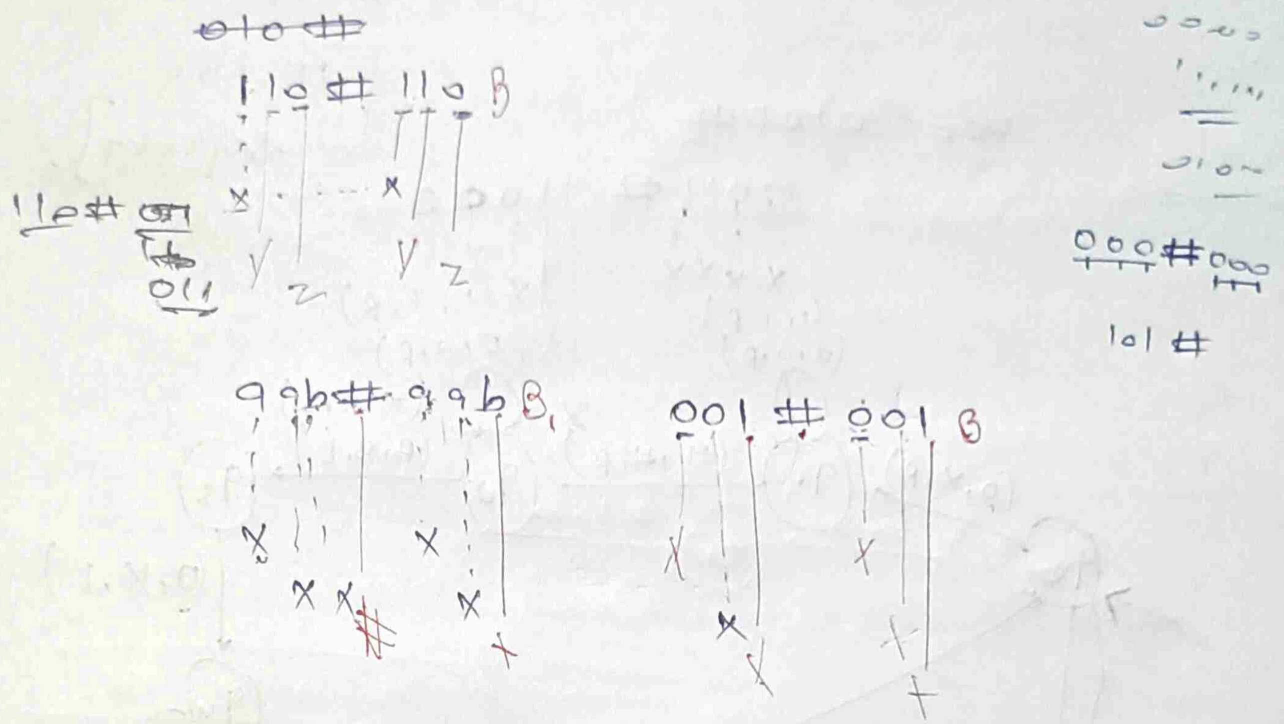


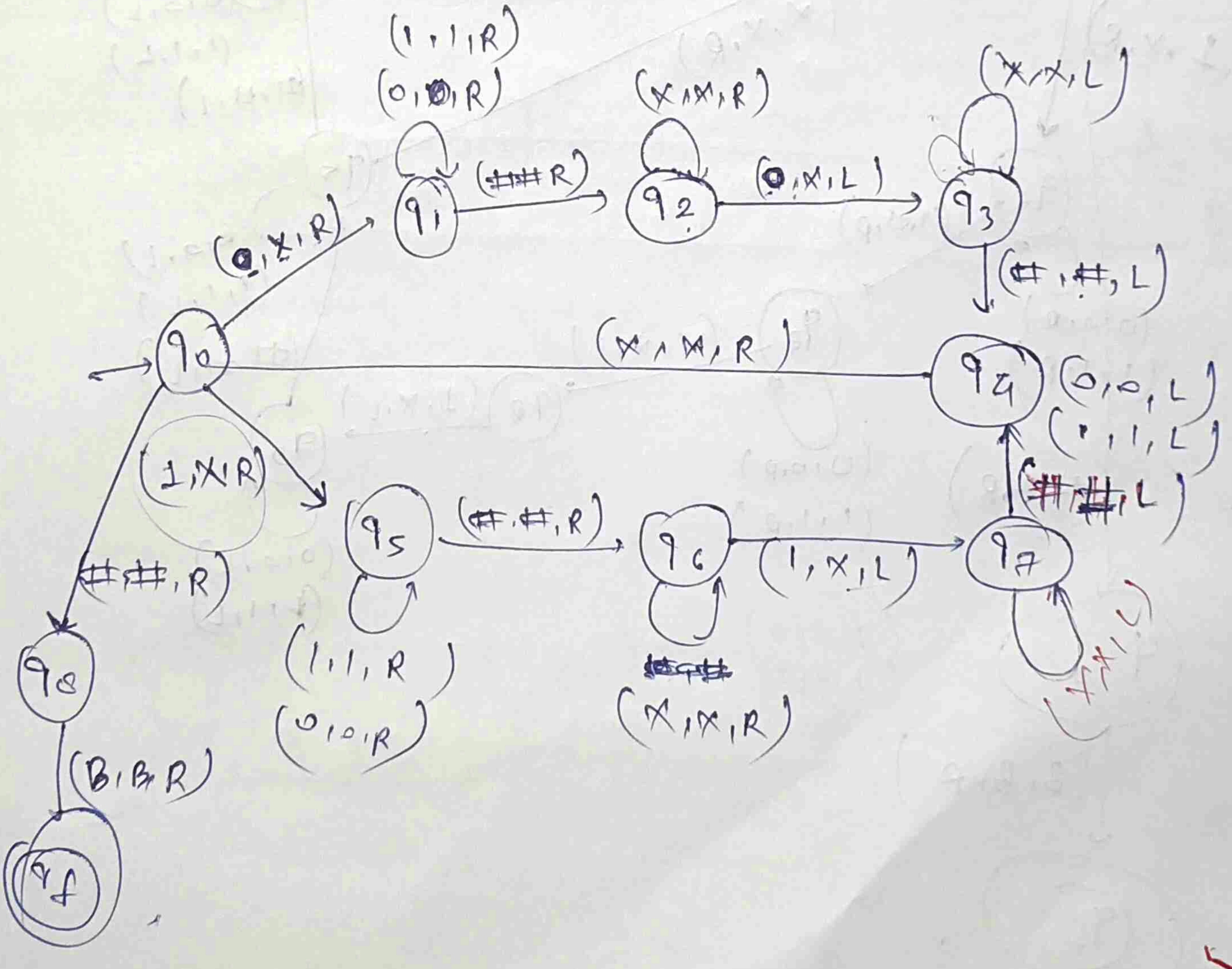


⑥ Design TM for $L = \{ w\#w \mid w \in (0,1)^* \}$.

$L = \{ \#,\ 0\#0,\ 1110\#1110$

$000\#000$





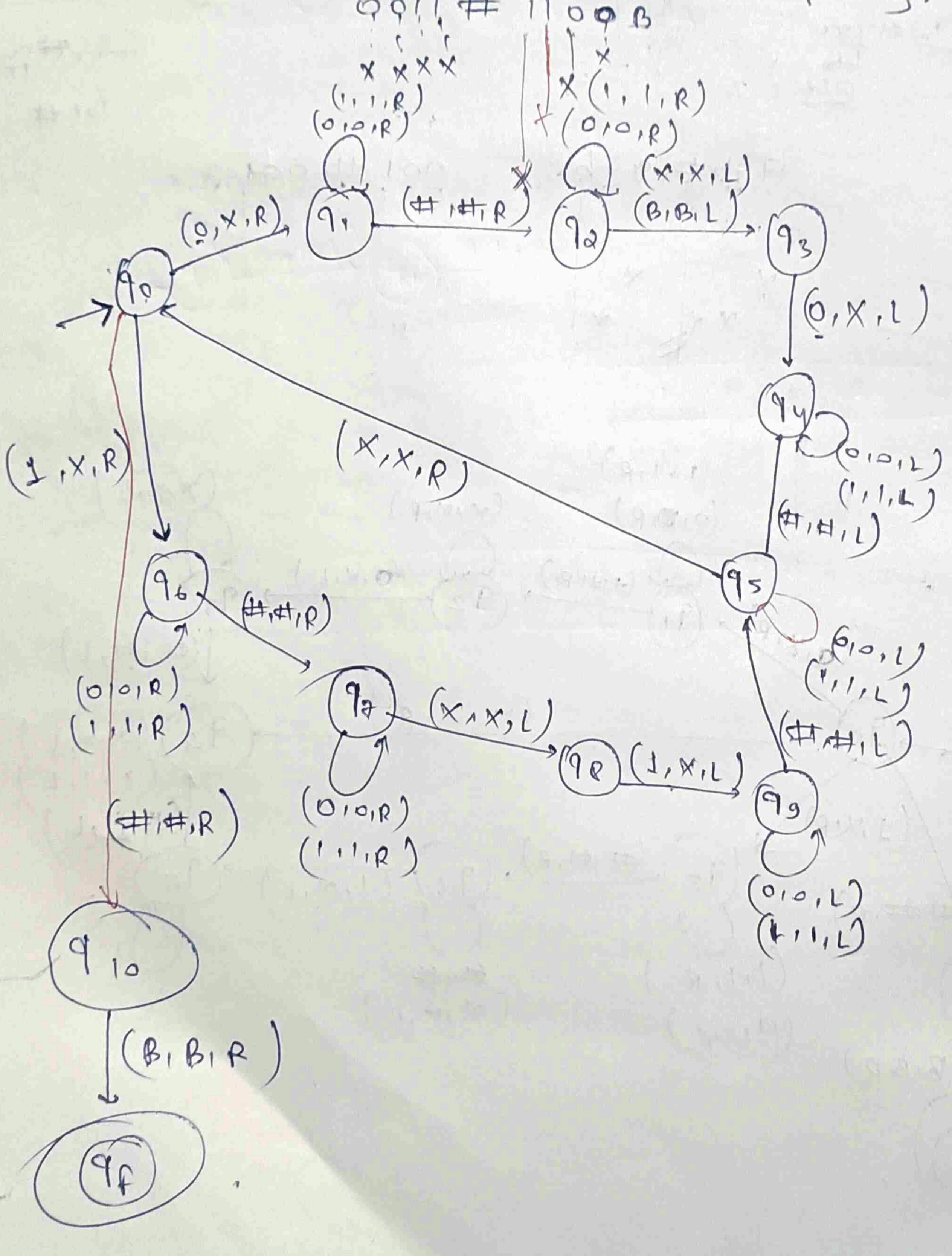
Shrvang 60

(Q.7). Design a TM for

$L = \{ ww^R \mid w \in (0,1)^* \}$.

w = ~~0101##~~

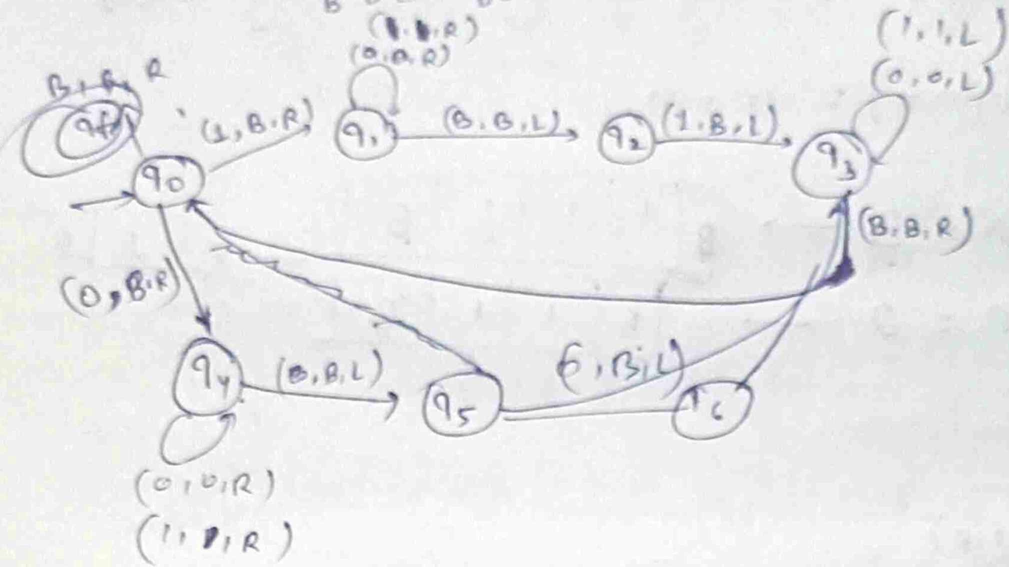
[# is a seperator].



II Approach

w = 110 011 B

B BB BB



$2^3\ 2^2\ 2^1\ 2^0$

0 1 0 1

Q. 8 Design a TM for calculate 2's complement of an input binary number.

5 → 0101

→ 1010

— 1

1011 B

B	0	1	0	1	B

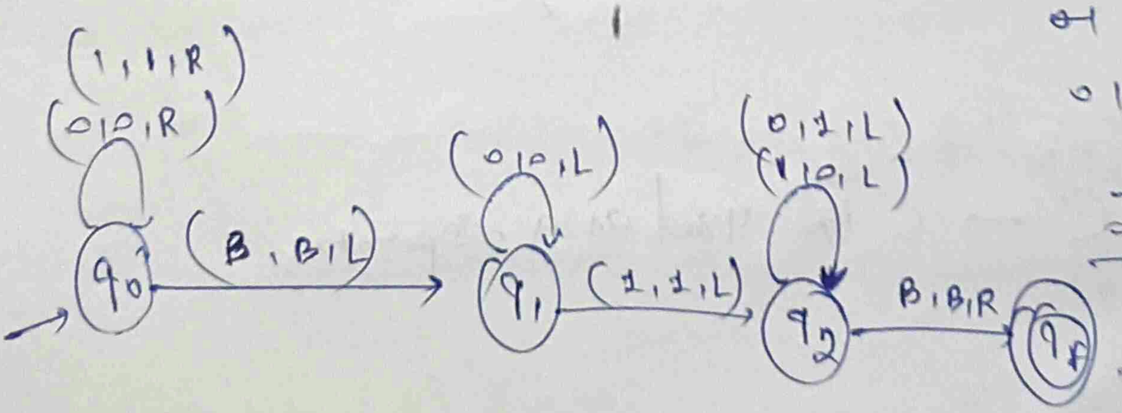
B	0	1	0	1	B
	1	0	1	1	

B 0 1 0 1 B

01000001

1

01000010



Shivang 60

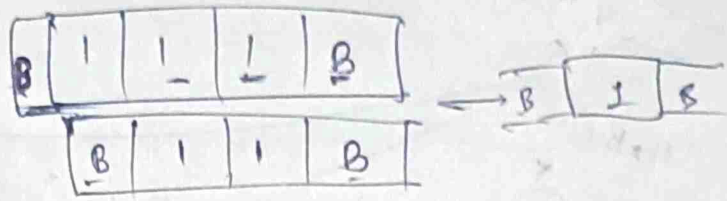
⑨. Design a Substraction of two unary numbers

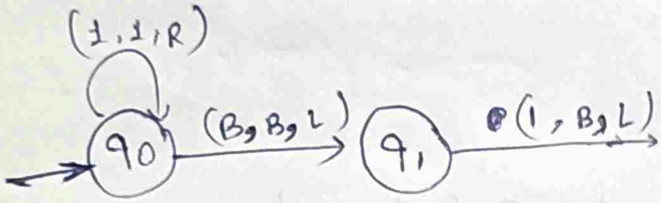
i.e $f(m,n) = m - n$.

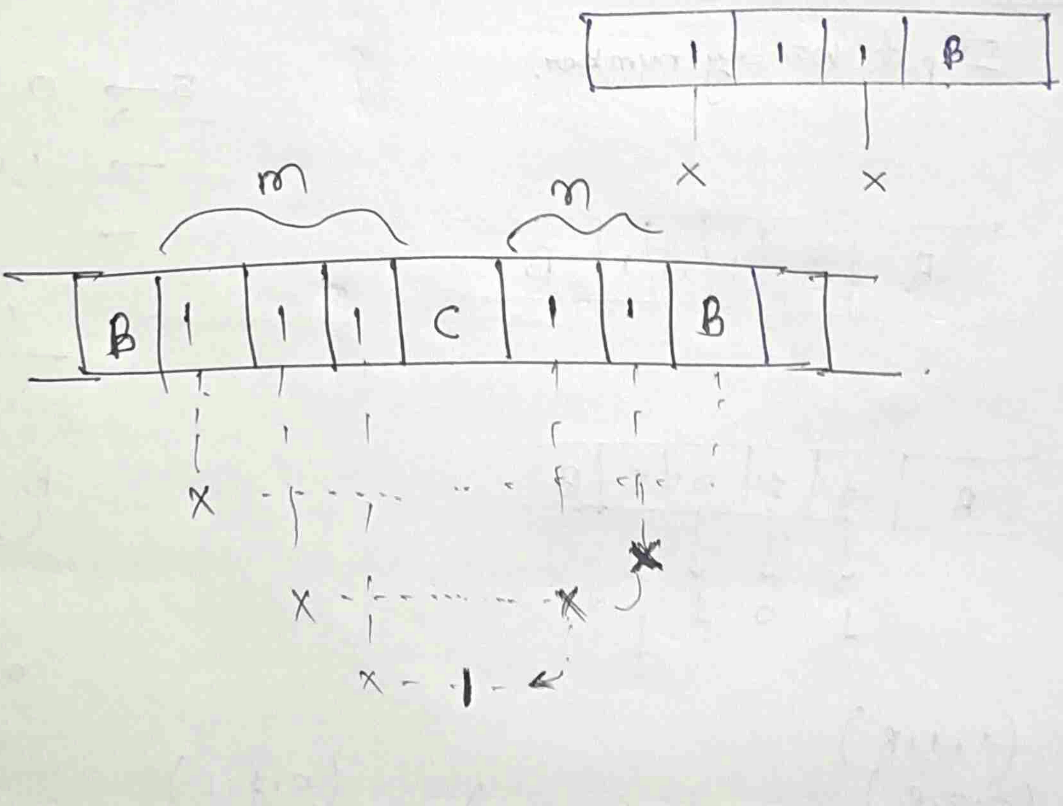
Solu:

$m = 3 \rightarrow$

$n = 2 \rightarrow$

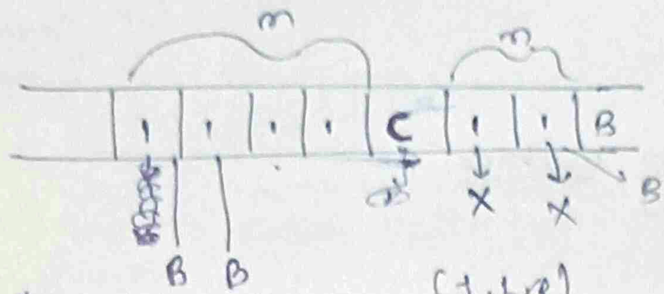


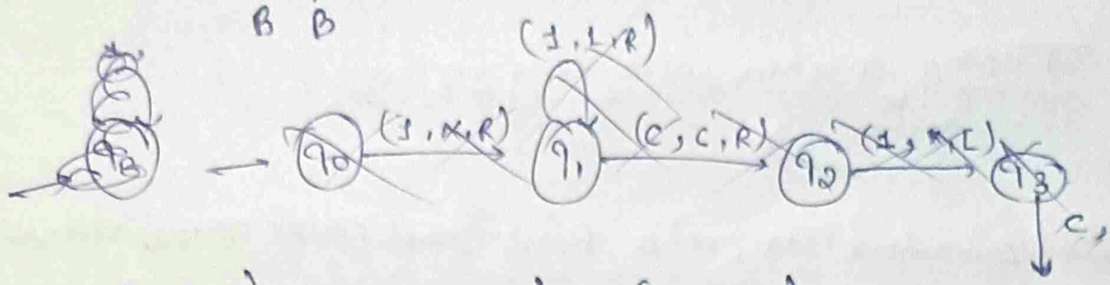


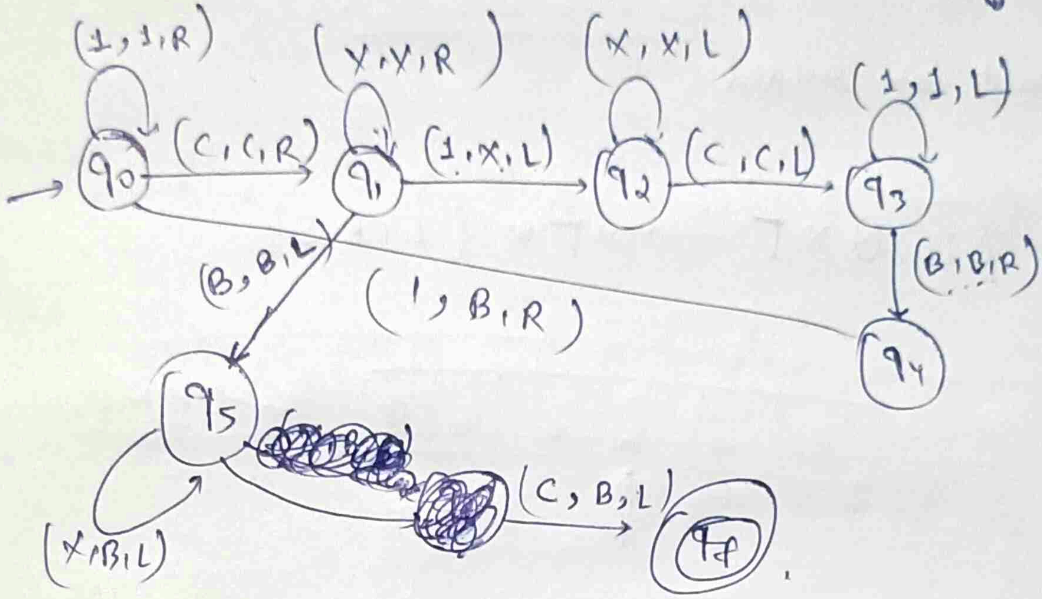


C → C is used as a seperator.

ab## B



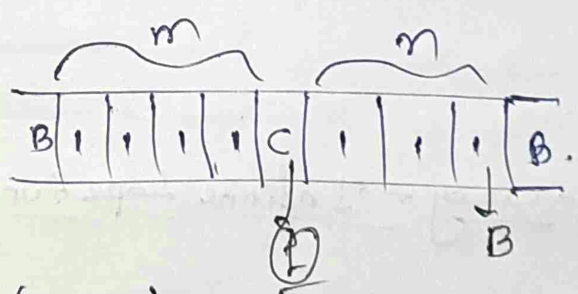


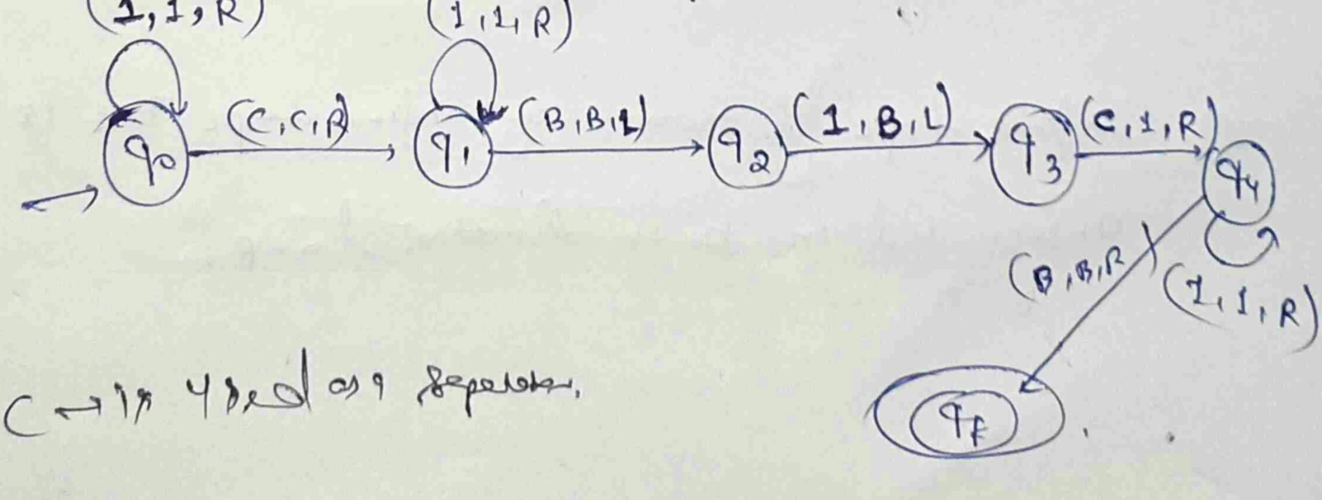


(10). Design a TM for $f(x,y) = x+y$ $\forall$ $x,y>0$.

$m = 4$

$n = 3$.





C → is used as a separator.

Q.12.

Variants of Turing-machine

(i). Turing machine with stay option.

→ In standard TM, the R/W head must move either left or right.

→. $$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R, S\}$$

ex.–

B.	a	b	a	b	B

↑ — Initial

B	a	a	a	b	B

↑

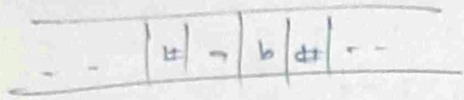
final $\{b, a, S\}$

(2). Two-way-Infinite tape turing-machine

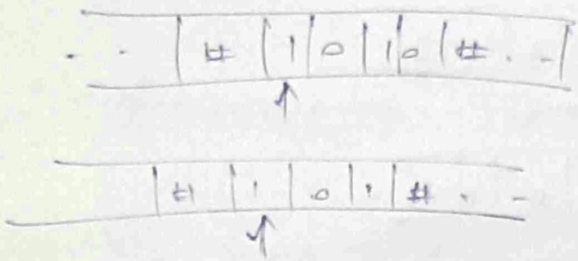
→ Infinite tape of two-way infinite tape TM is unbounded in both direction, L or R.

Shravang Co

#



the Behaviour when on the tape.

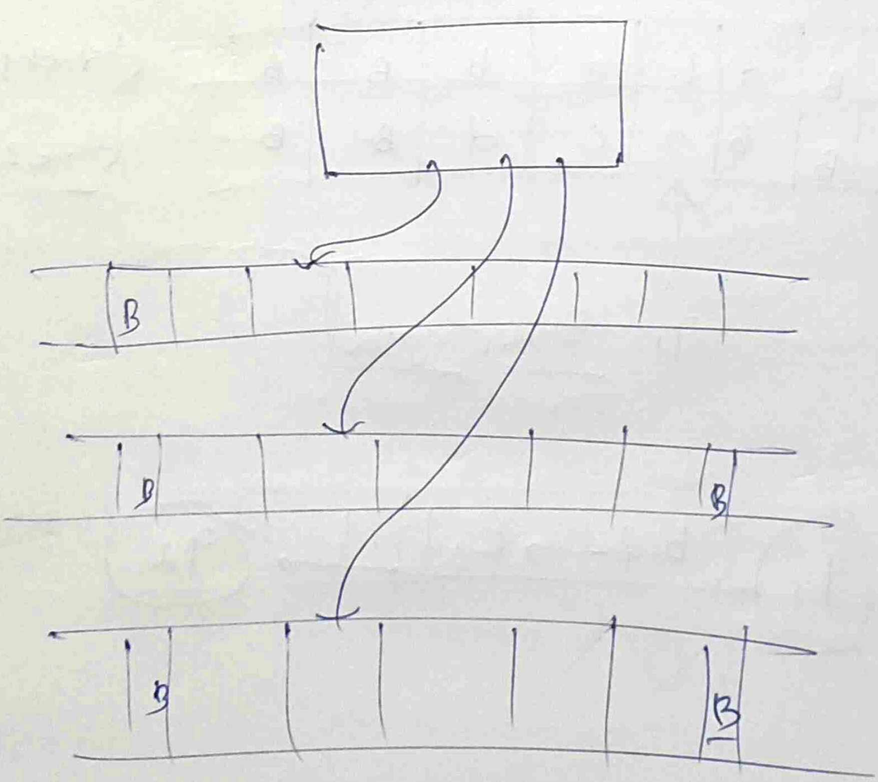


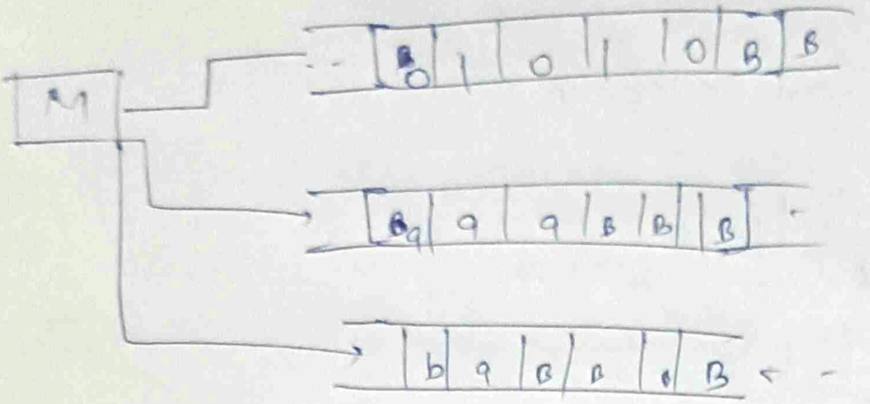
Multi Tape Turing Machine

→ TM with k tapes and k heads.

→ Each tape has its own head for Read and write.

$$\delta : Q \times \Gamma^k \longrightarrow Q \times \Gamma^k \times \{R, W\}^k.$$

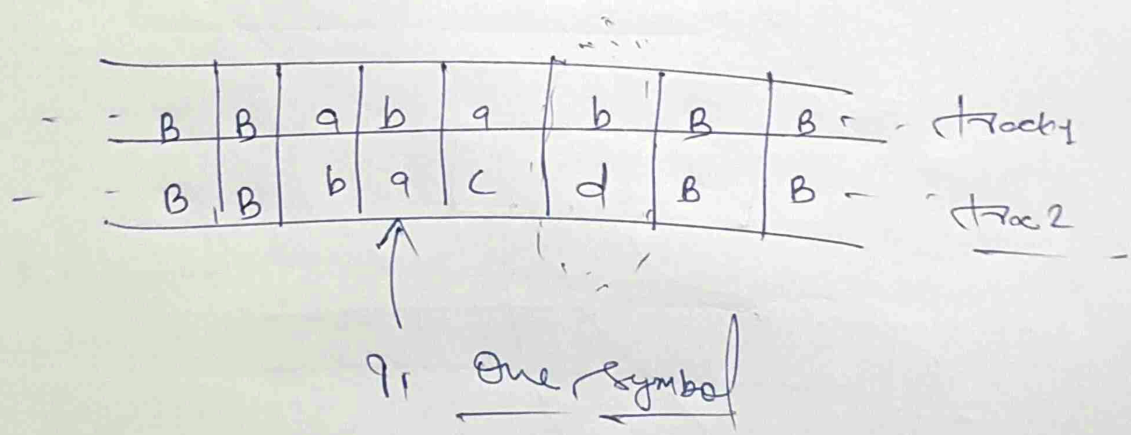




Represent 3 tapes in one.

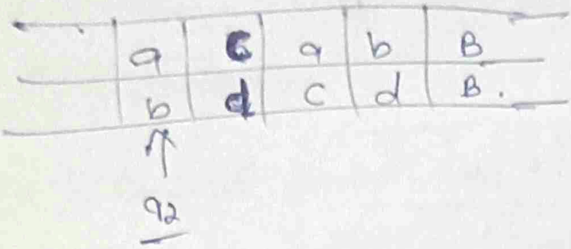
#	0	1	0	1	0	#	a	a	a	#	b	a	#	B.

(4). Multi-Track Turing Machine



Example.

$$q_1 \xrightarrow{(b,a)\,\to\,(c,d),\,L} q_2$$



$$Q \times \Sigma \to 2^{Q}$$

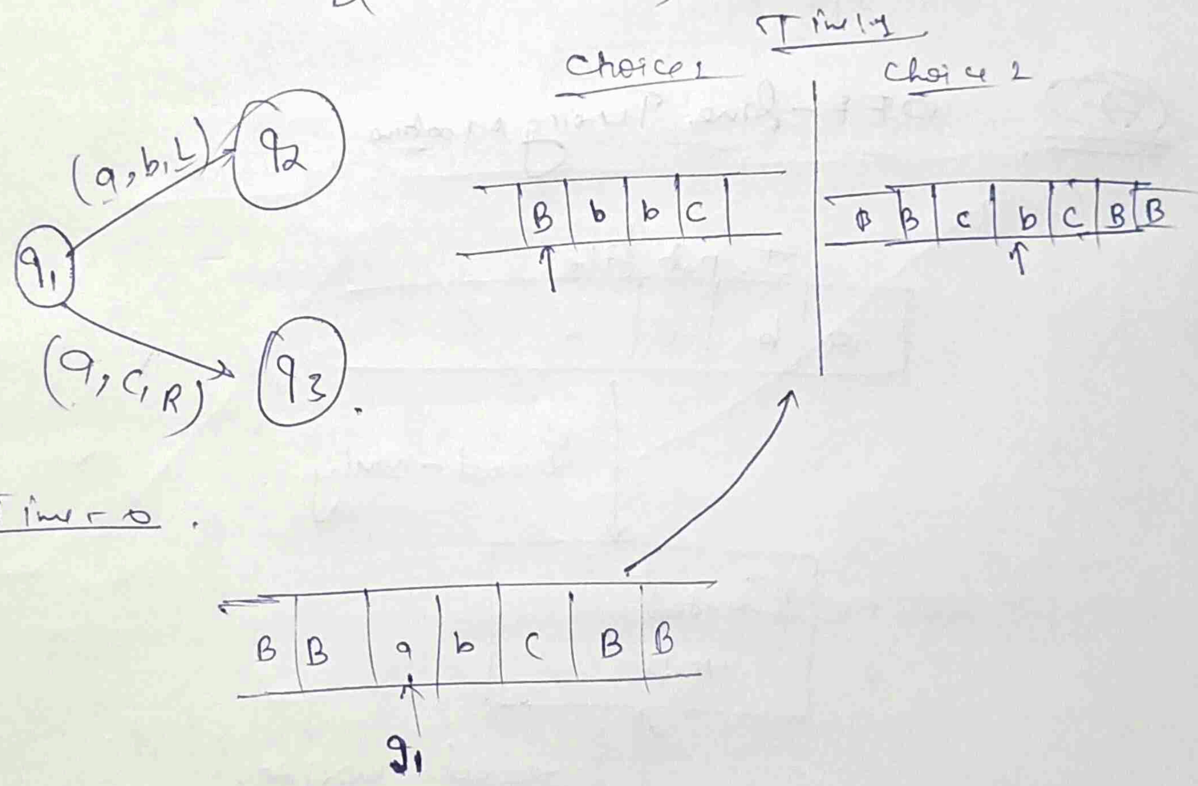
$$Q \times \Gamma \to 2^{Q \times \Gamma \times (R \times L)}$$

5. Nondeterministic Turing Machine

→ At any point in a Computition the machine may proceed according to several composition.

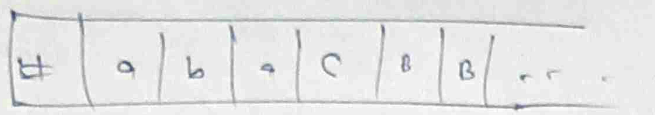
→ $Q \times \Gamma \to 2^{(Q \times T \times \{R, L\})}$

Ex

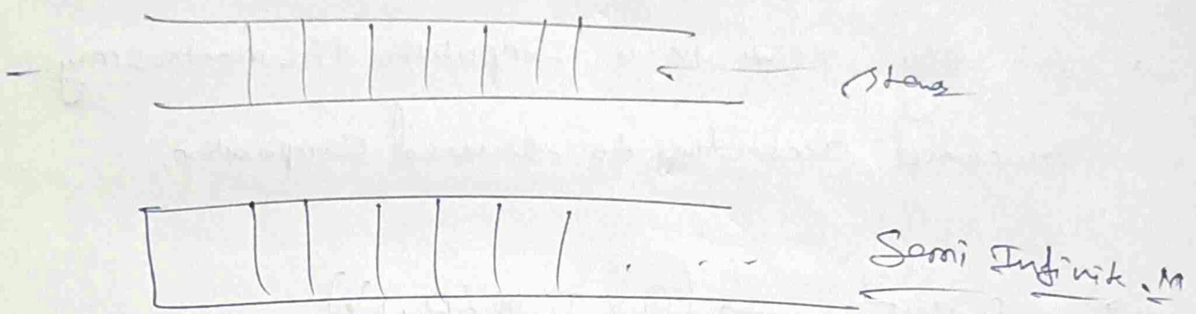


Remark :- Non-deterministic machine have same power as Deterministic turing machine.

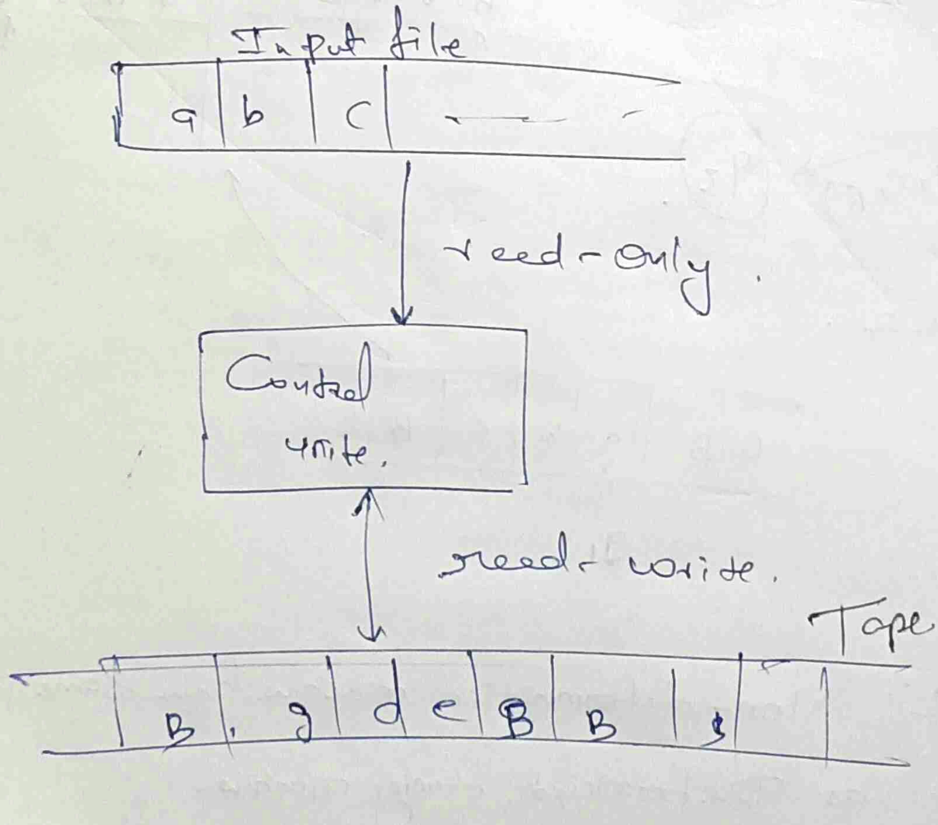
(6) Semi. Infinite tape turing



Standard T.M simulates semi-infinite tape machines.



(7) OFF-line Turing Machine



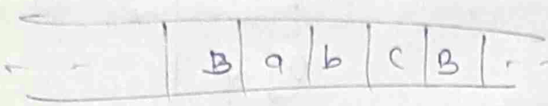
Off-line machines simulate Standard TM. | Shivang Goel |

Off-line machine :-

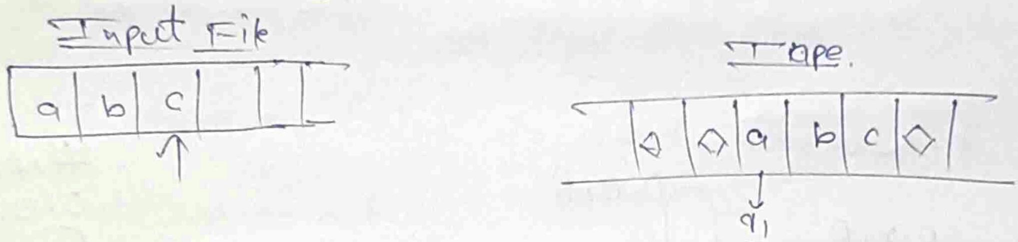
1. Copy input file to tape.

2. Continue Computation as in Standard Turing machine.

~~Star~~ Standard machine.



Off-line machine



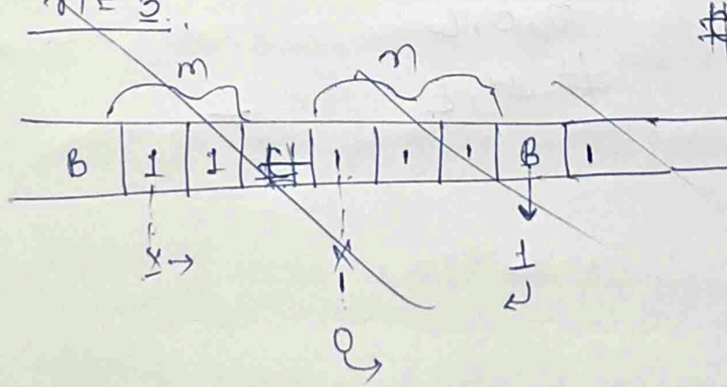
(1.) Copy Input file to tape.

(2). Do Composition as in turing machine.

Q.13 Design a TM for $f(m,n) = m \times n$.

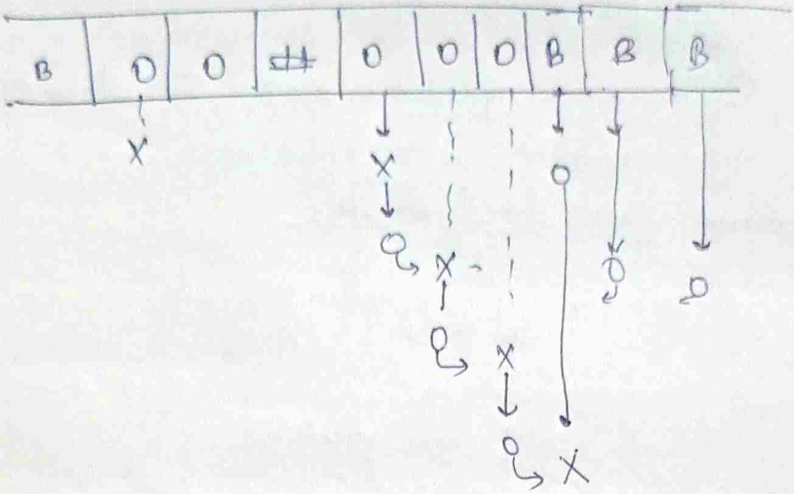
Solve:- $m = 2$
$n = 3$.

$\# \rightarrow$ ~~Sepd~~ Seperator.

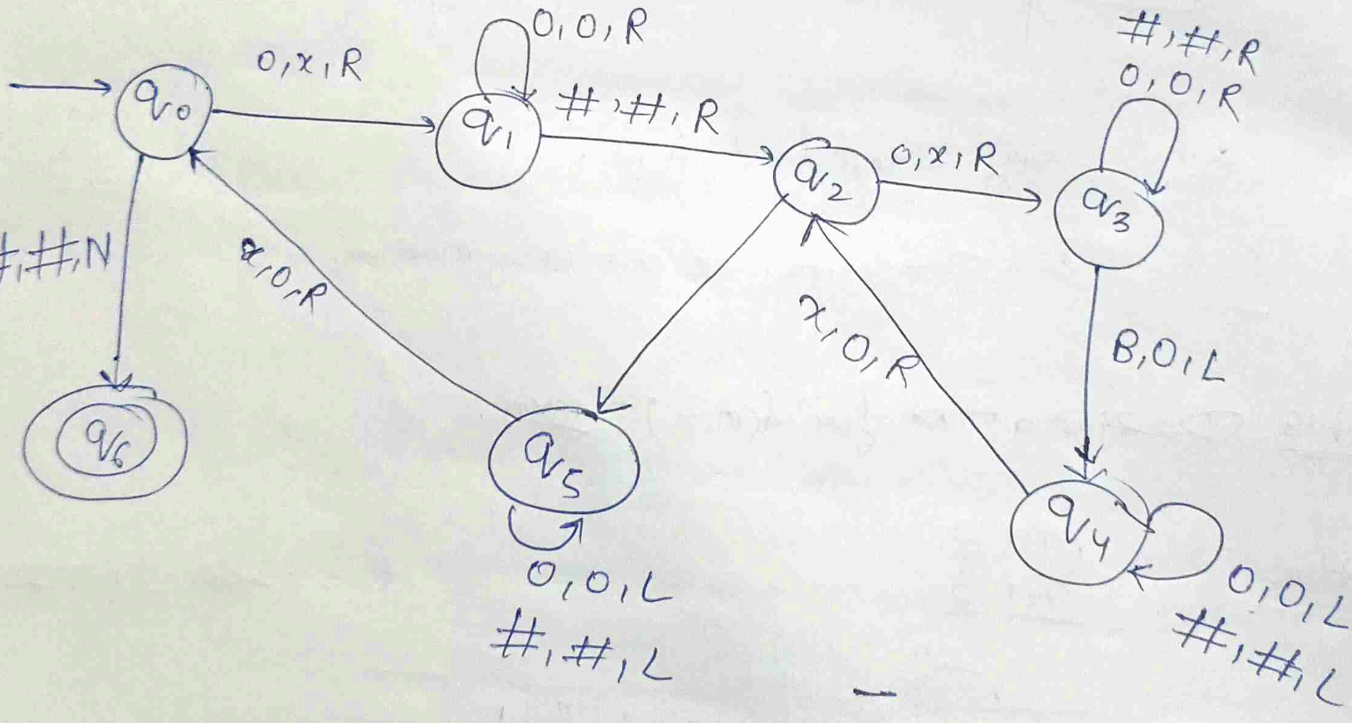


m = 2

n = 3



B
0
0
#
0
0
0
B
B
B



q_0
q_1
q_2
q_3
q_4
q_5
q_6
0, x, R
0, 0, R
#, #, R
0, x, R
#, #, R
0, 0, R
B, 0, L
0, 0, L
#, #, L
x, 0, R
0, 0, L
#, #, L
x, 0, R
#, #, N

Shivang 60

Q.4. Discuss the following

- Halting problem of TM.

- Halting means Termination.
- unsolvable problem.

Given a ~~prob~~ program and an input to the program, determine if the program will eventually stop when it is given that input.

- Given a TM M and input string w.

- Is there an "Algo" to decide whether M halts on input w or not?

- It is not recursive so it is undecidable.

- Halting means that program on certain input will accept it & halt or reject it & halt and never go into an infinite loop.

- So, can we have Algo that will tell that the given program will halt or not?

- In terms of TM, will it terminate when run on some machine with some particular given input string.

- There is no Algo / procedure that will determine whether an arbitary TM, will halt on a input string.

Church thesis.

- Every algorithmically Computable function is TM- Computable.

- A function is Computable if it can be solved by a Turing machine.

Thesis not Theorm!- Because we cannot prove this. with a ~~___~~ Counter example we could disprove it (by this not done yet).

→ ~~By~~ Any Algorithmic procedure that can be solved by ~~___~~ human or team of human that can be carried out by TM.

Shivang Go

Recursive and recursive enumerable

Recursive:- A language is recursive, if some Turing machine accepts it and halts on any input string.

$L \longrightarrow$ recursive language

$M \longrightarrow$ TM.

For string ω.

if $\omega \in L$ then M halts at final state.

if $\omega \in L$ the M halts at non-final state.

Recursive enumerable

language is recursively enumerable if some T.M accepts it.

Only acceptable case is in RE.

$L \longrightarrow$ recursive E.

$M \longrightarrow$ T.M that accept it.

if $\omega \in L$ the M halts at final state.

if $\omega \in L$ then M halts in a non-final or loop forever.

Different Compleaxity classes

Complexity and Computablity

① **class P.**

If problem is solue by TM in polynomial time (fesible time).

Ex: → Postman problem.

→ Searching shortest path.

→ searching maching in a graph.

→ Complexity parameter (i) Time

(ii) Space.

② **class NP.**

Language L is in class NP. If there is a non-deterministic TM (M) and a polynomial time-Complexity (t_n) such that:-

$\boxed{L = L(M)}$ and when (M) is given an input of length (n).

ex: (i) Travelling salesman prob.

(ii) lineer programming

(iii) Graph isomerphism.

Shivang Goel

Np Complete

let L be a language in NP, we can say L is in NP Complete. If the following statement true :-

(i) L is in NP.

(ii) For every language L' in NP there is polynomial time reduction of L' to L.

(iii) Np-hard problem

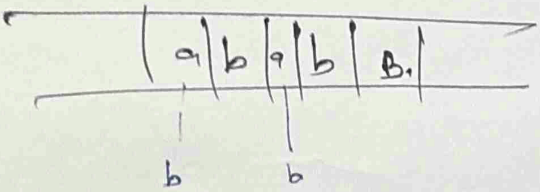
Some problem L are so hard that although we can prove condition.

(ii) of NP Complete, But we cannot prove Condition (1). that is L is in NP.

This problem is called Np-hard problem.

Ex. 2^{1000}.

Q.15

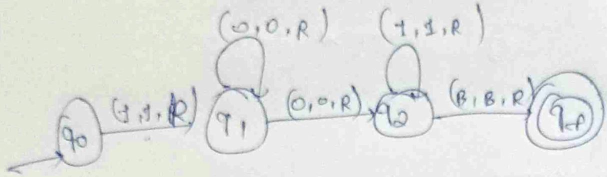


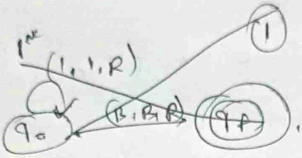
Not accepted string "abba".

Shivangi

(16)

Soln. $10^* + 01^*$





$$\begin{array}{r} 10 \\ +\ 01 \\ \hline 11 \end{array} \qquad \begin{array}{r} 1 \\ 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 100 \\ 011 \\ \hline 111 \end{array} \qquad \begin{array}{r} 1000 \\ 0111 \\ \hline 1111 \end{array}$$

1000 0111

100 011

111 ~~qf~~