

①

Name : HAMEED KHAN

Department : BS (CS)

Semester : 4th

ID #1 : 15066

Subject : Design and Analysis of Algorithms

Submitted To : Muhammad Adil

Dated : Spring/24/2020



②

~~... ..~~  
~~... ..~~  
~~... ..~~

Q 2

Ans (i)

$$D - Y * (F / G)$$

Conversion:

Pre-fix Notation:

$$\underline{D} - \underline{Y} * (\underline{F} / \underline{G})$$

$$= - \underline{D} \underline{Y} * (\underline{F} / \underline{G})$$

$$= - \underline{D} * \underline{Y} (\underline{F} / \underline{G})$$

$$= - \underline{D} * \underline{Y} ( / \underline{F} \underline{G} ) \quad \text{Ans}$$

Post-fix Notation:

$$\underline{D} - \underline{Y} * (\underline{F} / \underline{G})$$

$$= \underline{D} \underline{Y} * (\underline{F} / \underline{G}) -$$

$$= \underline{D} \underline{Y} (\underline{F} / \underline{G}) * -$$

$$= \underline{D} \underline{Y} ( \underline{F} \underline{G} / ) * - \quad \text{Ans}$$



3

(ii)

$$T/W \wedge R + S * M - Y \wedge K$$

Conversion:-

Pre-fix Notation:-

$$\underline{T/W \wedge R + S * M - Y \wedge K}$$

$$= + \underline{T} / \underline{W \wedge R} \underline{S * M - Y \wedge K}$$

$$= + / \underline{T} \underline{W \wedge R} - \underline{S * M} \underline{Y \wedge K}$$

$$= + / T \wedge R - * S M \wedge K \quad \text{Ans}$$

Post fix Notation:-

$$\underline{T/W \wedge R} + \underline{S * M - Y \wedge K}$$

$$= \underline{T} / \underline{W \wedge R} \underline{S * M - Y \wedge K} +$$

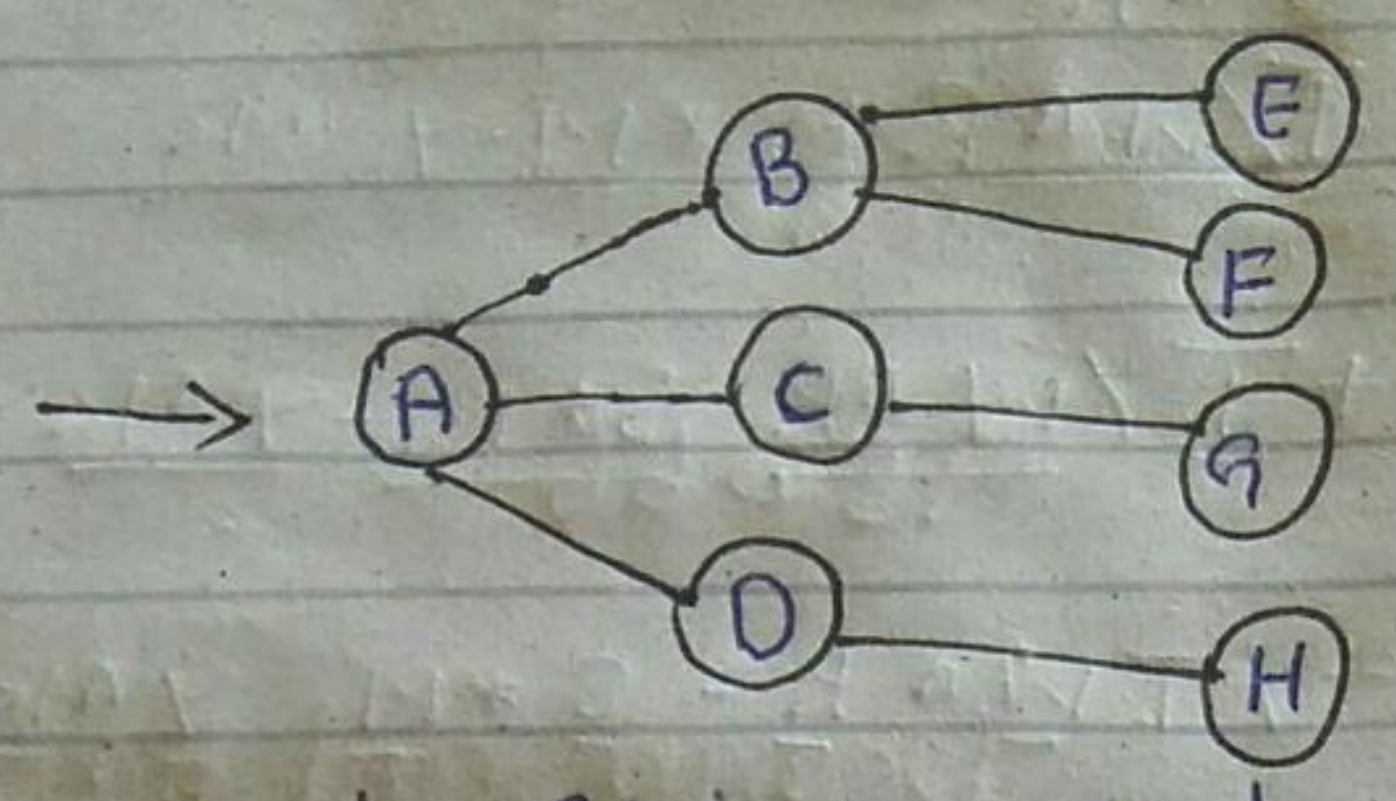
$$= \underline{T} \underline{W \wedge R} / \underline{S * M} \underline{Y \wedge K} - +$$

$$= T \wedge R / S M * Y K \wedge - + \quad \text{Ans}$$



(13)

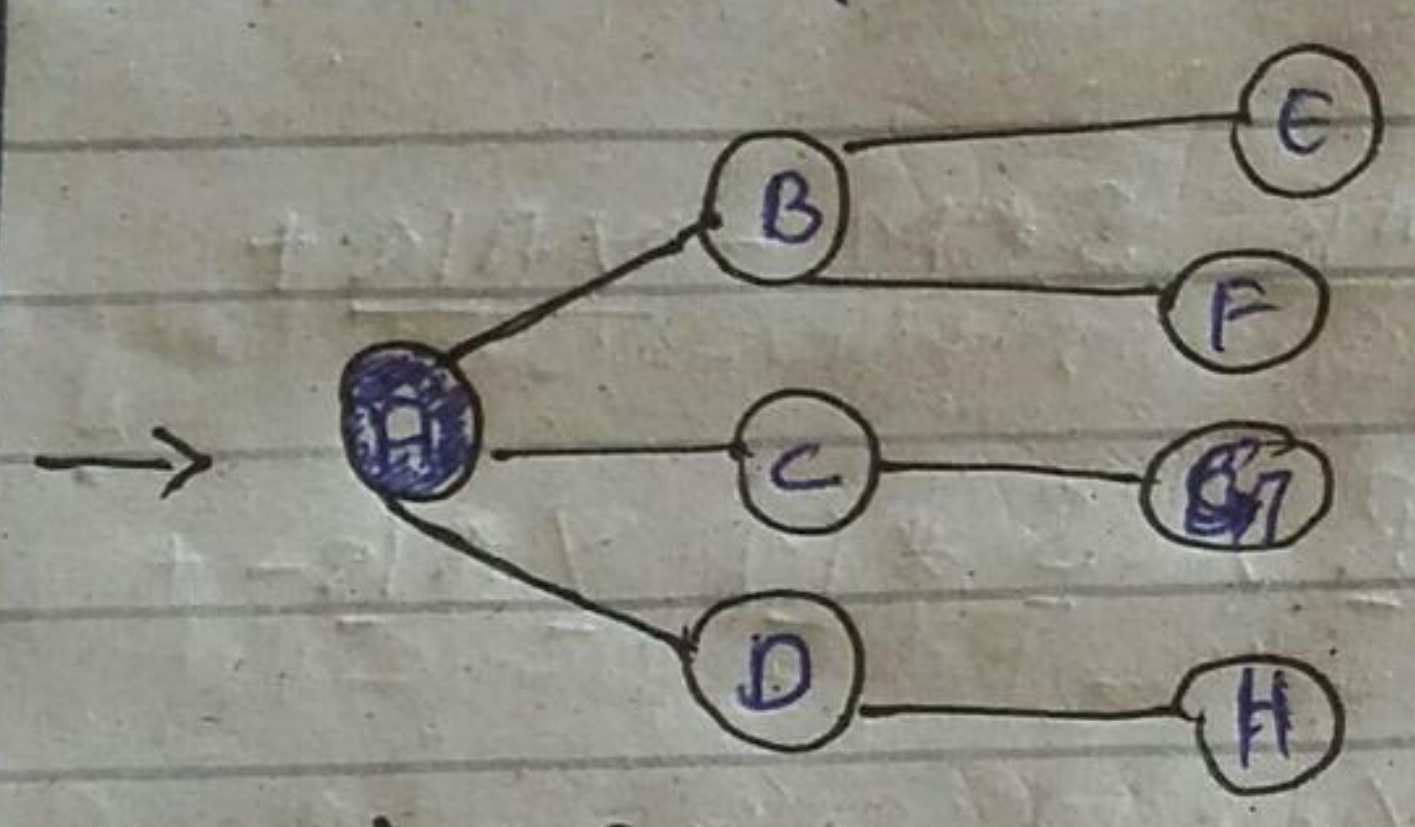
Ans Breadth-First Technique:



① (\*) Root 'A' is correct working Node (CWN).

~~...~~

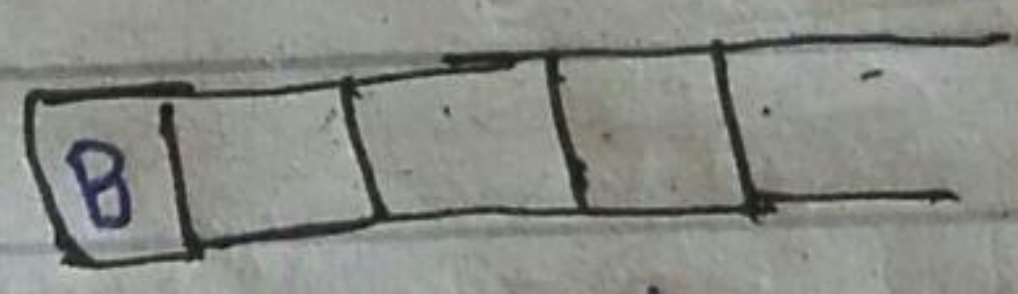
- \* Mark 'A' visited
- \* Add 'A' to the output sequence



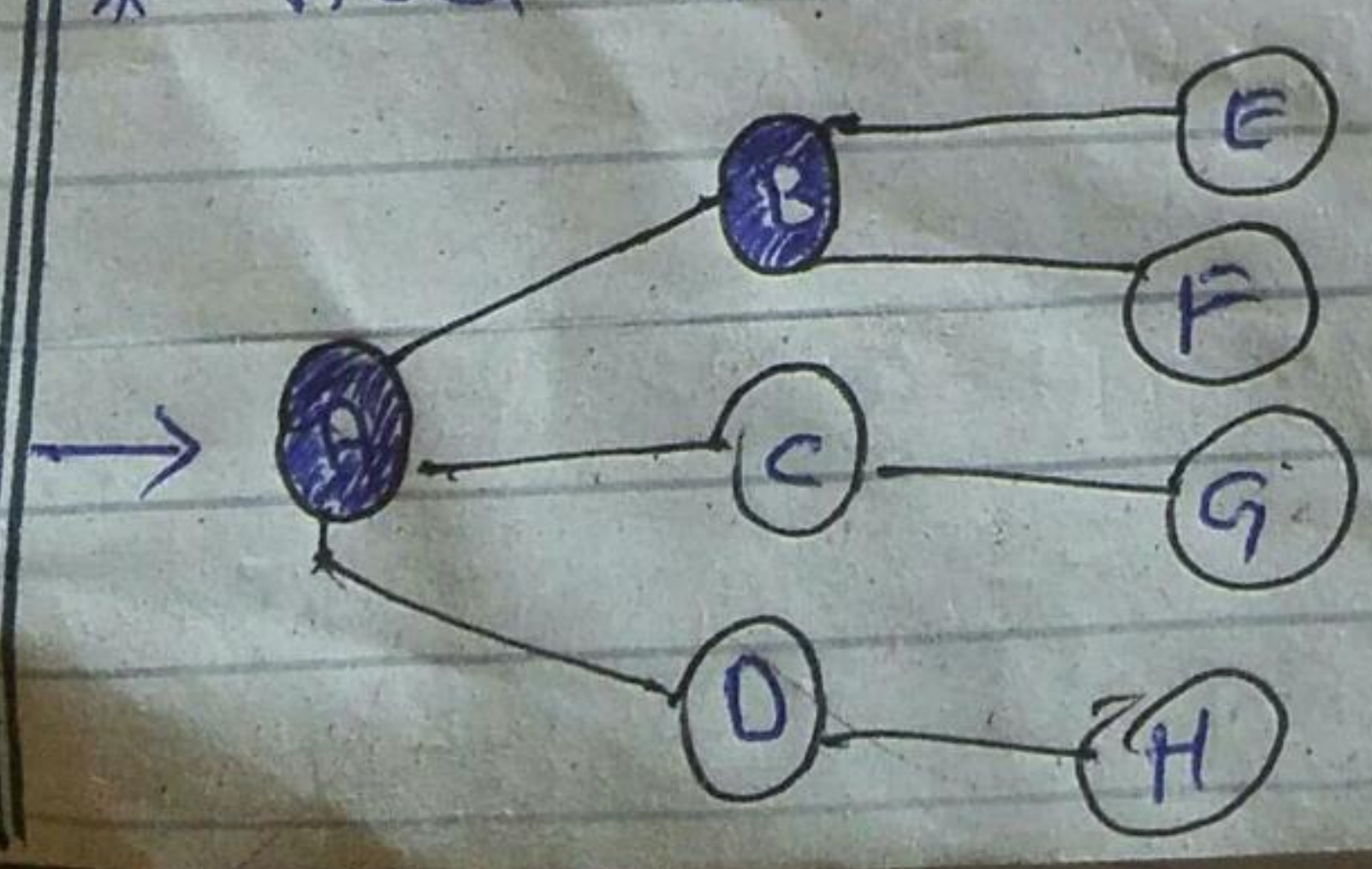
Output Sequence

②

- A.
- \* A is adjacent to B, C and D.
- \* Select 'B' and push it into



- \* Mark 'B' visited.
- \* Add 'B' to the output sequence





# Output Sequence

A, B

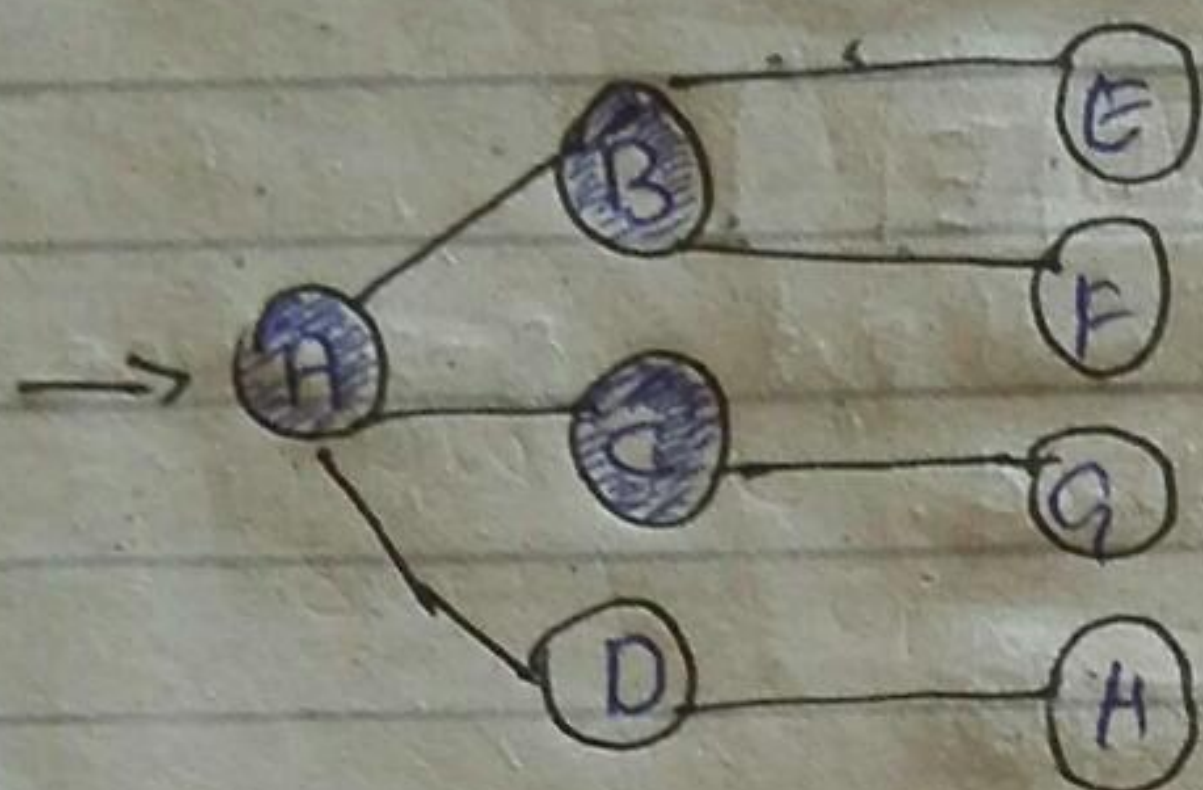
(\* Accessing 'C' from CWN is 'A')

(\* Push 'C' into  $\text{Q}$



(\* Mark 'C' visited)

(\* Add 'C' to the output Sequence.



# Output Sequence:

A, B, C

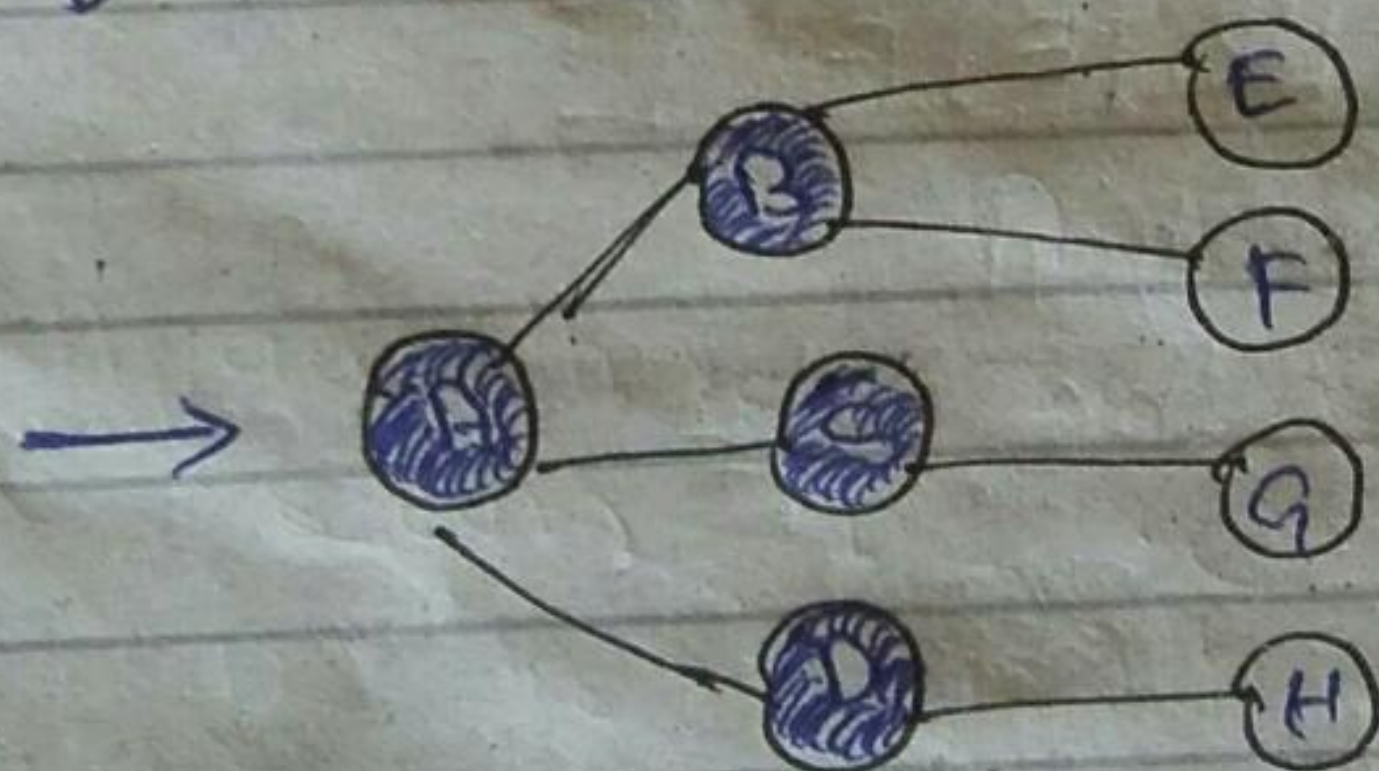
\* From CWN i.e 'A' the adjacent node 'D' is Selected.

(\* 'D' is pushed into the  $\text{Q}$



\* 'D' is marked visited.

\* 'D' is added to the output Sequence



# Output Sequence:

A, B, C, D

\* Now as there are no more nodes adjacent to CWN i.e 'A'

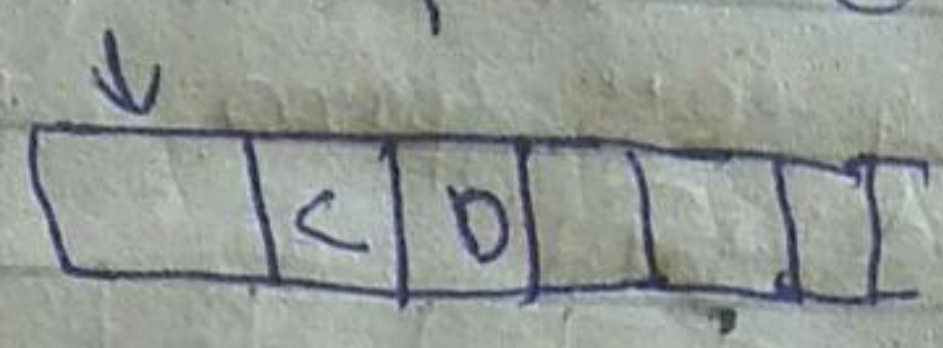
So update CWN.



(5)

(\*) Select 'B' as CWN.

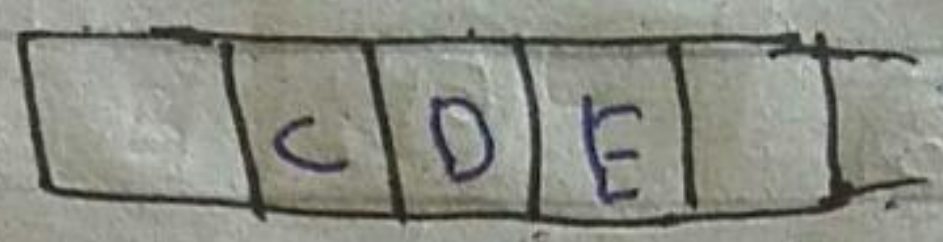
(\*) pop it from  $\sigma$



(\*)

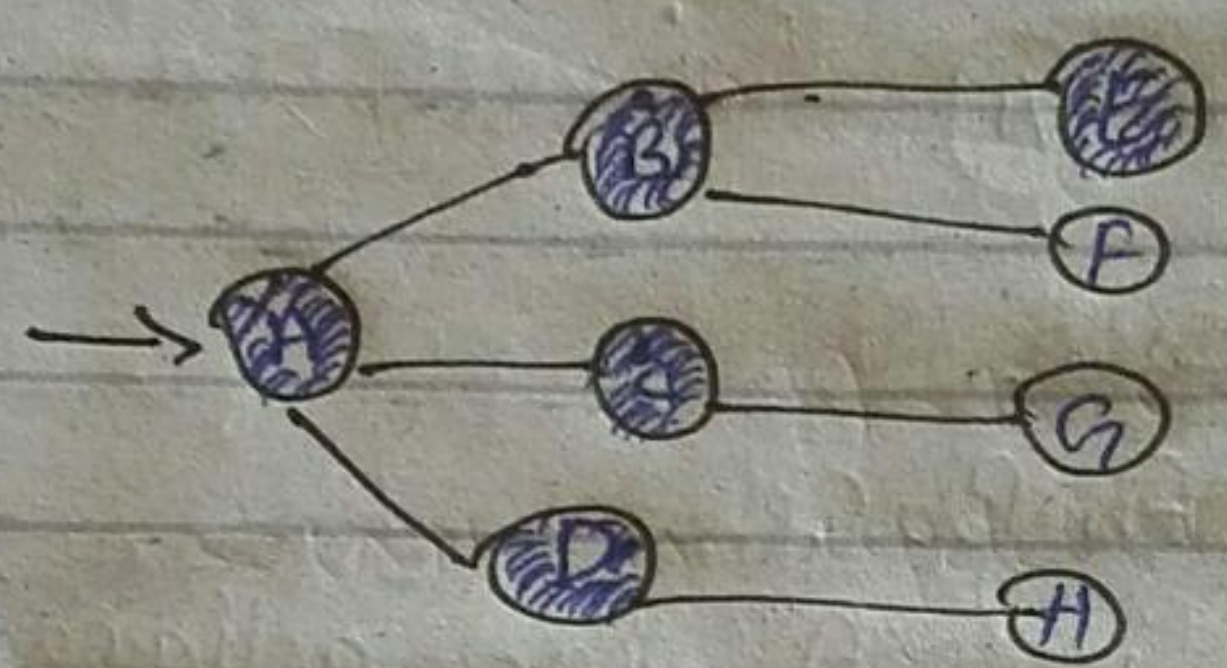
(5) \* B is adjacent to E and F

\* select 'E' and push it into  $\sigma$



\* Add 'E' to the output sequence

\* Mark 'E' visited.

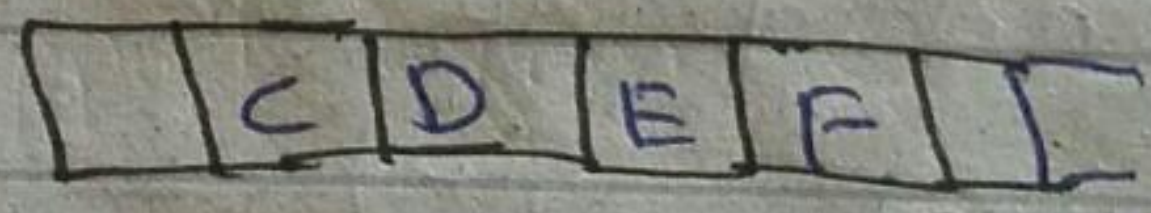


output sequence

A, B, C, D, E

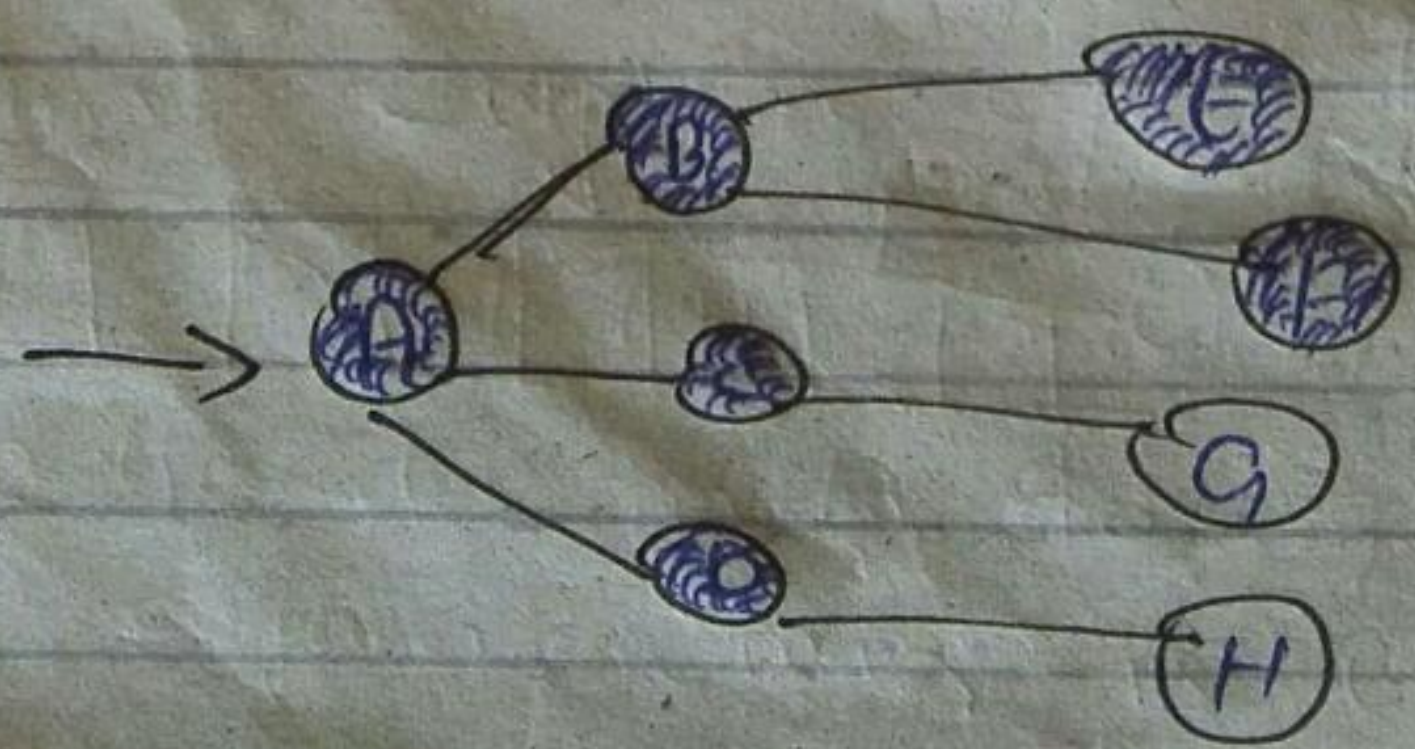
(6) (\*) from CWN i.e. 'B' access 'F'

\* push 'F' into  $\sigma$



\* Mark 'F' visited

\* add 'F' to the output sequence



output sequence



7

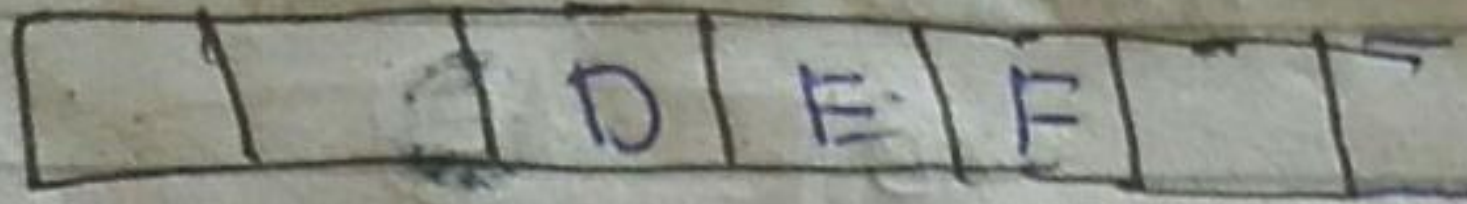
A, B, C, D, E, F

\* As there are no more nodes adjacent to CWN i.e. 'B' so update

CWN

\* Select 'C' as CWN (New)

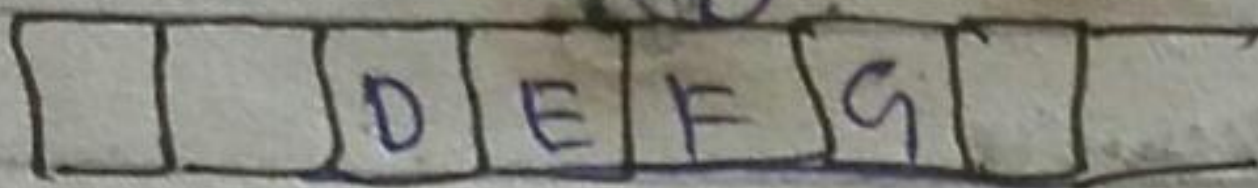
\* 'C' is popped from Q



7

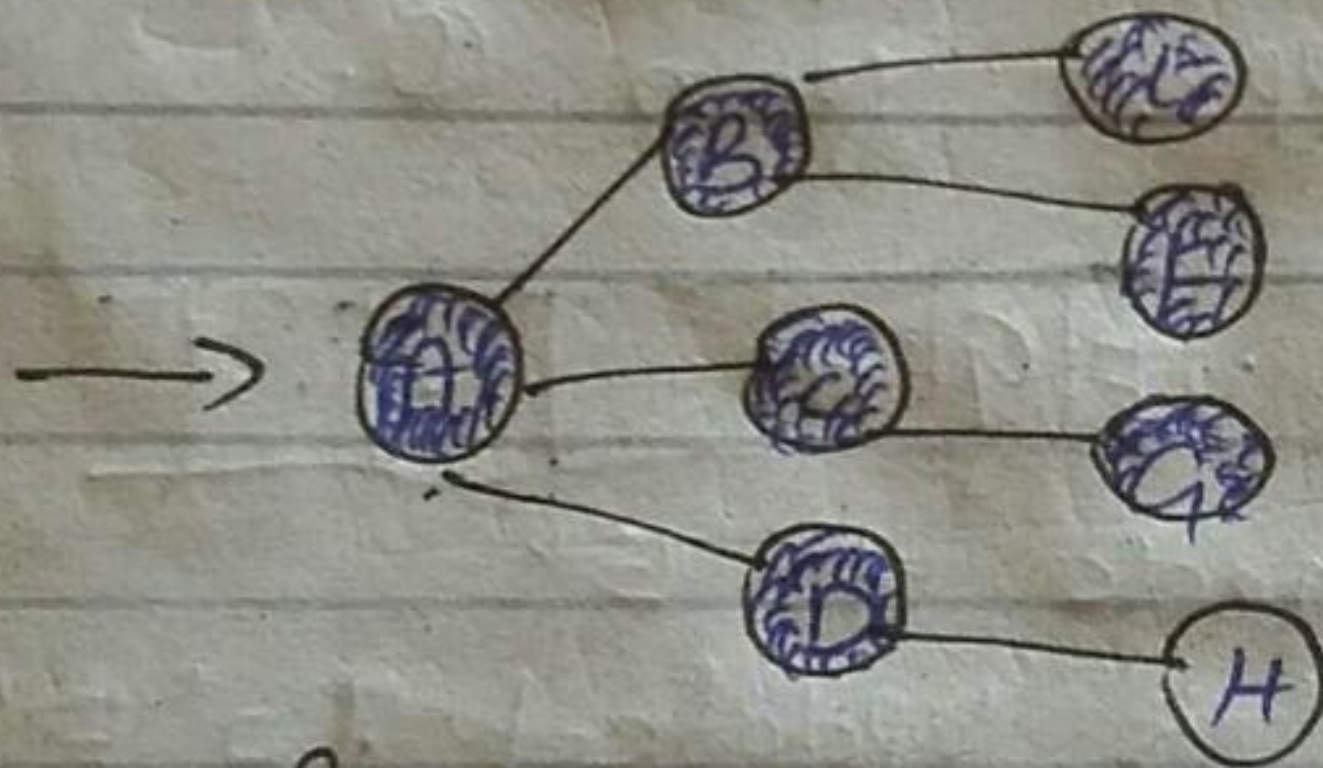
\* Now 'C' is adjacent to 'G'

\* Select 'G' and push it into the



\* 'G' is marked visited

\* 'G' is added to output sequence.



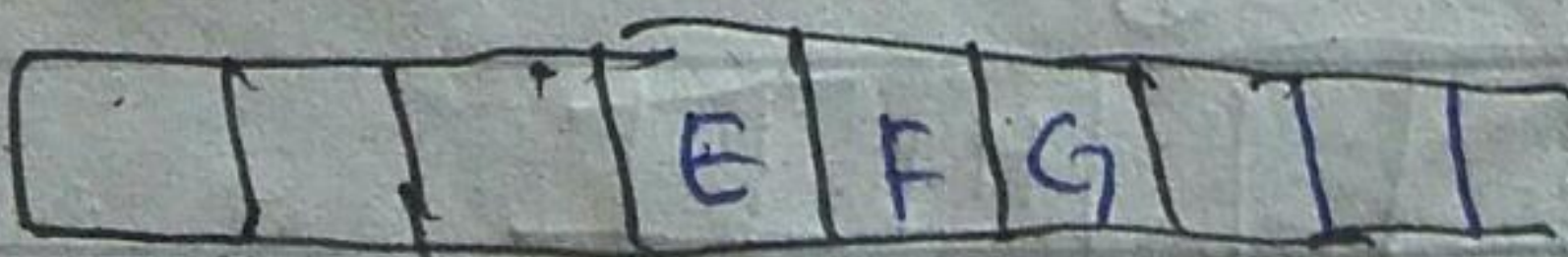
output sequence:-

A, B, C, D, E, F, G

\* Again there are no more nodes adjacent to CWN i.e. 'C' so update CWN.

\* 'D' is selected as new CWN

\* 'D' is popped from Q



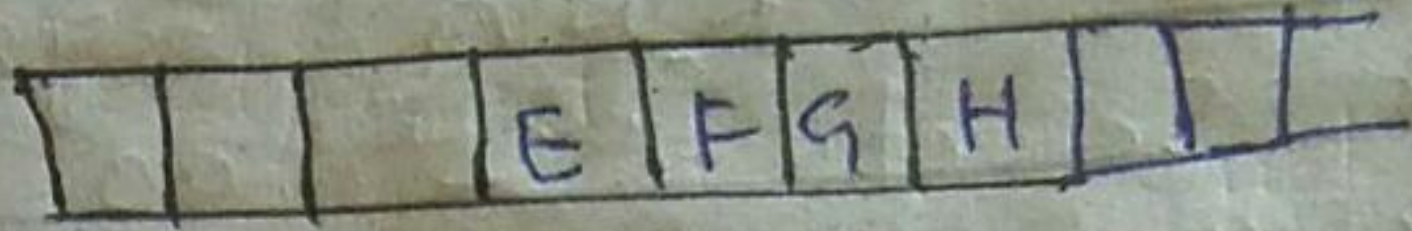


8

8

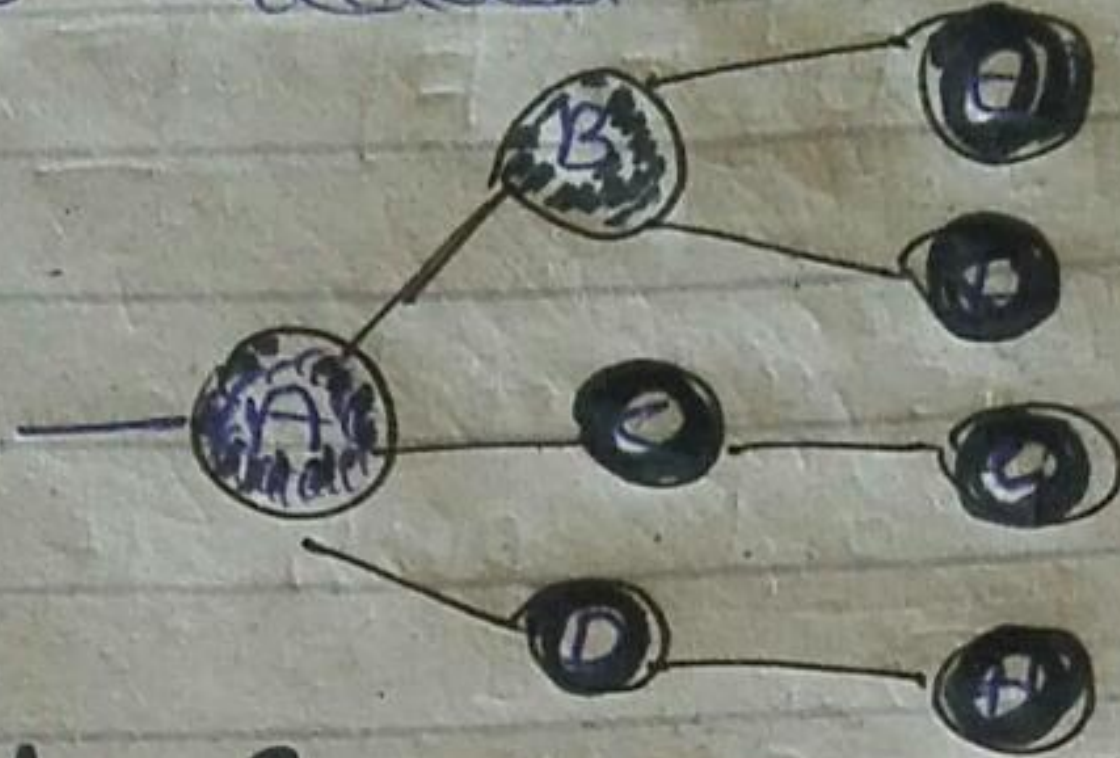
\* from CWN i.e 'D' adjacent node is H.

\* 'H' is selected and is pushed into the



\* 'H' is marked visited.

\* 'H' is added to output sequence

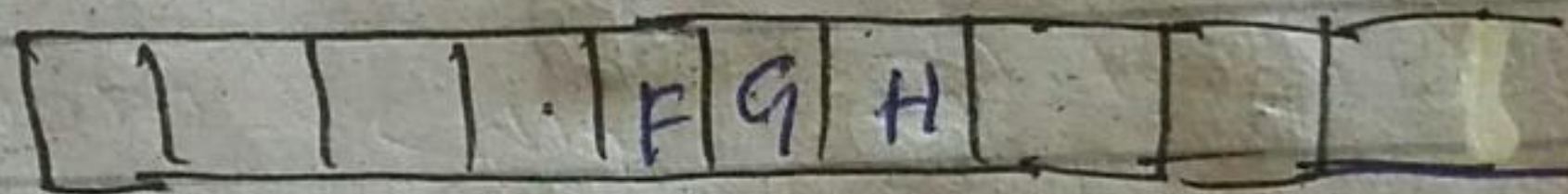


### Output Sequence

A, B, C, D, E, F, G, H

\* Now CWN is updated to 'E'

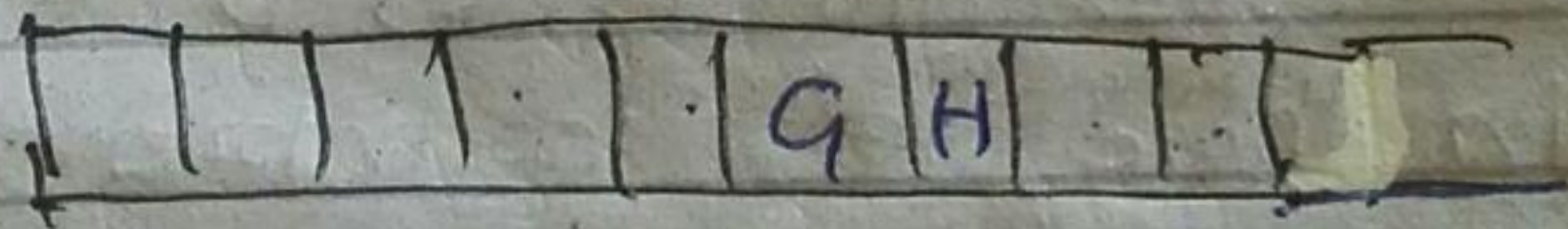
\* 'E' is popped from



\* No adjacent node to 'E'

\* Again CWN is updated to 'F'

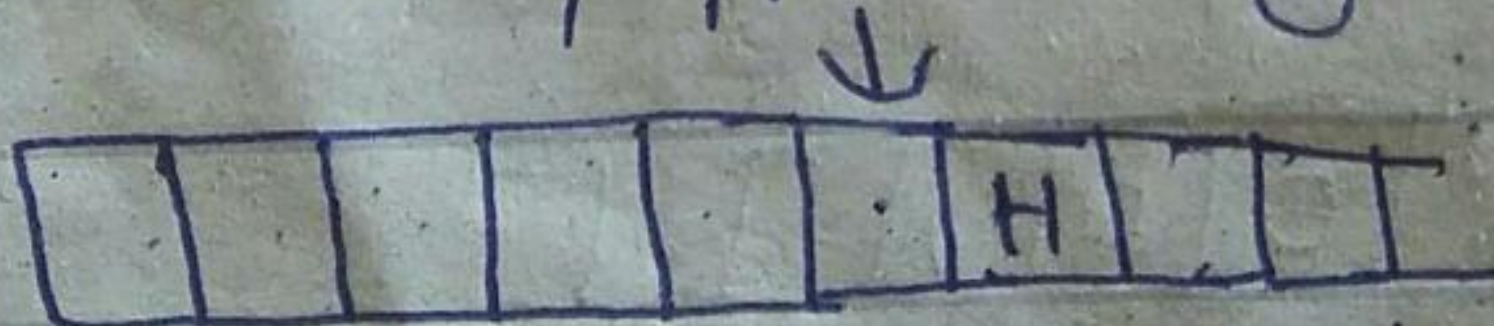
\* 'F' is popped from



\* No adjacent node to 'F'

\* Again CWN is updated to 'G'

\* 'G' is popped from



\* No adjacent node to 'G'

\* Now again CWN is updated to 'H'



Q

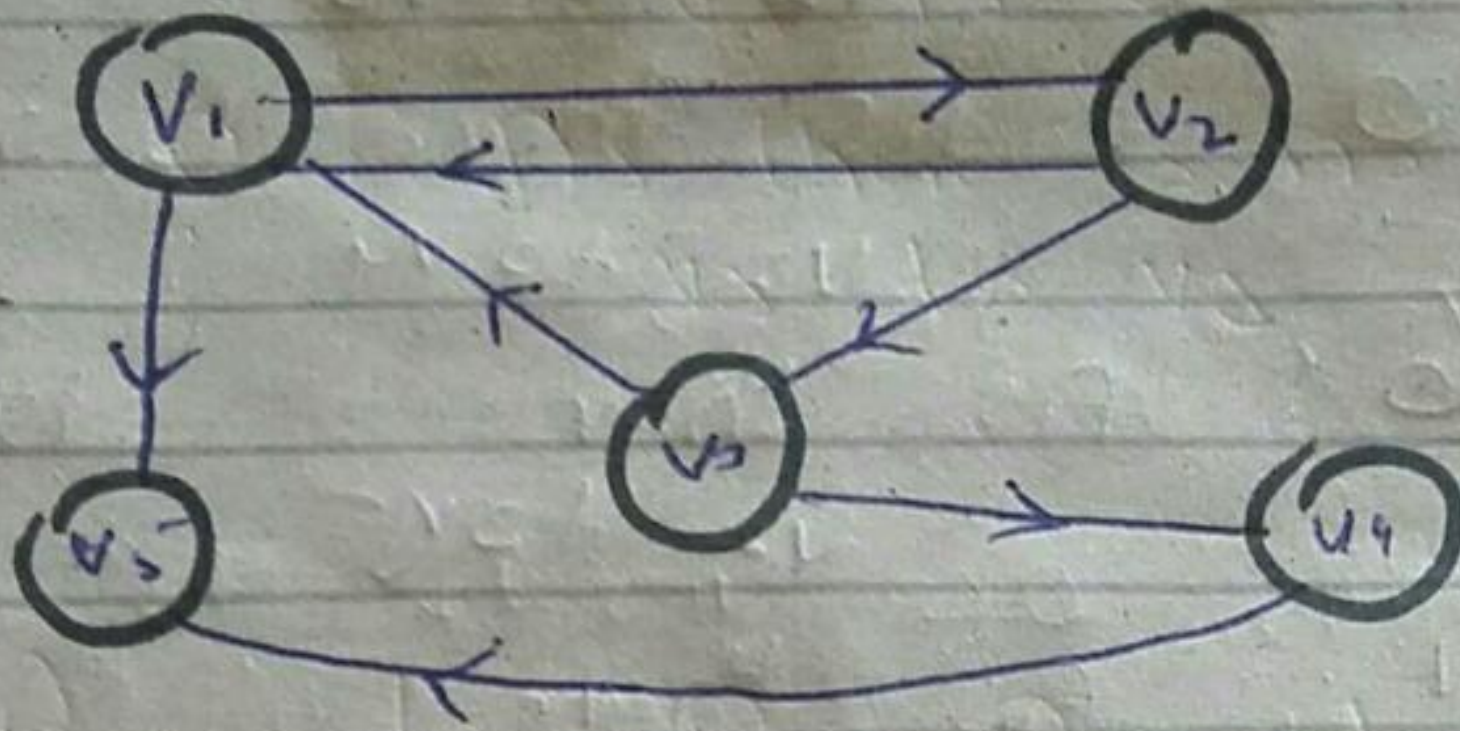
\* H is popped from Q



\* Q is now empty so Breadth first search stops

Q 4:-

Ans adjacency Matrix:



In this graph:

Number of nodes =  $n = 5$   
order of  $A = n \times n$   
 $= 5 \times 5$

$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$a_{15}$
$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$a_{25}$
$a_{31}$	$a_{32}$	$a_{33}$	$a_{34}$	$a_{35}$
$a_{41}$	$a_{42}$	$a_{43}$	$a_{44}$	$a_{45}$
$a_{51}$	$a_{52}$	$a_{53}$	$a_{54}$	$a_{55}$

Now

$a_{11} = 0$  : AS there is no edge from  $v_1$  to  $v_1$

$a_{12} = 1$  : AS there is an edge from  $v_1$  to  $v_2$

$a_{13} = 0$  : AS there is no edge from  $v_1$  to  $v_3$



$a_{14} = 0$  : As there is no edge from  $v_1$  to  $v_4$

$a_{15} = 1$  : As there is an edge from  $v_1$  to  $v_5$

$a_{21} = 1$  : As there is an edge from  $v_2$  to  $v_1$

$a_{22} = 0$  : As there is no edge from  $v_2$  to  $v_2$

$a_{23} = 1$  : As there is an edge from  $v_2$  to  $v_3$

$a_{24} = 0$  : As there is no edge from  $v_2$  to  $v_4$

$a_{25} = 0$  : As there is no edge from  $v_2$  to  $v_5$

$a_{31} = 1$  : As there is an edge from  $v_3$  to  $v_1$

$a_{32} = 0$  : As there is no edge from  $v_3$  to  $v_2$

$a_{33} = 0$  : As there is no edge from  $v_3$  to  $v_3$

$a_{34} = 1$  : As there is an edge from  $v_3$  to  $v_4$

$a_{35} = 0$  : As there is no edge from  $v_3$  to  $v_5$

$a_{41} = 0$  : As there is no edge from  $v_4$  to  $v_1$

$a_{42} = 0$  : As there is no edge from  $v_4$  to  $v_2$

$a_{43} = 0$  : As there is no edge from  $v_4$  to  $v_3$

$a_{44} = 0$  : As there is no edge from



(11)

$a_{45} = 1$  : As there is an edge from  $v_4$  to  $v_5$

$a_{51} = 0$  : As there is no edge from  $v_5$  to  $v_1$

$a_{52} = 0$  : As there is no edge from  $v_5$  to  $v_2$

$a_{53} = 0$  : As there is no edge from  $v_5$  to  $v_3$

$a_{54} = 0$  : As there is no edge from  $v_5$  to  $v_4$

$a_{55} = 0$  : As there is no edge from  $v_5$  to  $v_5$

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	Outdegree
$v_1$	0	1	0	0	1	2
$v_2$	1	0	1	0	0	2
$v_3$	1	0	0	1	0	2
$v_4$	0	0	0	0	1	1
$v_5$	0	0	0	0	0	0

Indegree

2 1 1 1 2

(7)

which is required Adjacency Matrix



(Q5)

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

As

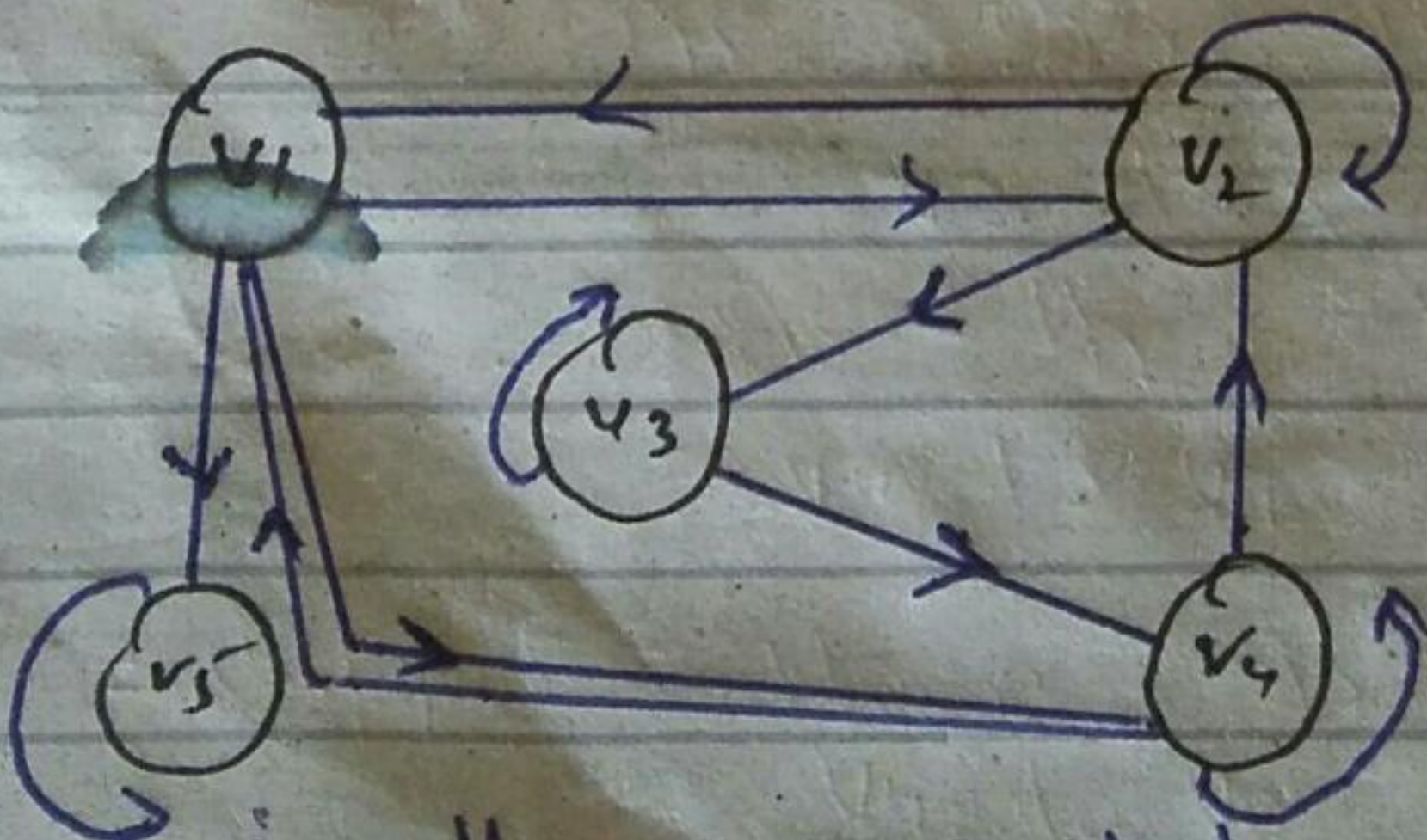
order of A = m x m  
= 5 x 5

So

Number of nodes = 5  
Let's

the nodes be v<sub>1</sub>, v<sub>2</sub>, v<sub>3</sub>, v<sub>4</sub> and v<sub>5</sub>

$$A = \begin{matrix} & \begin{matrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{matrix} \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$



which is the required

Directed Graph



(13)

(No 1)

- ① Verten
- ② multiple / parallel edge
- ③ Adjacent edges
- ④ Simple path
- ⑤ cycle
- ⑥ : source code
- ⑦ Sink
- ⑧ isolated / null graph
- ⑨ regular graph
- ⑩ labeled graph

The End