

①

NAME IKRAMULLAH
ID# 15072
DEPARTMENT BS-CS 2nd

ANSWER#1

- ① vertex
- ② Multiple / Parallel edge
- ③ Adjacent edges
- ④ Simple path
- ⑤ cycle
- ⑥ Source Node
- ⑦ Sink
- ⑧ Isolated or Null graph
- ⑨ Regular graph.
- ⑩ Labeled graph.

ANSWER#2

① $D - \gamma * (F/G)$

pre-fix

Post-fix

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

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

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

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

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

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

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

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

ii) $T/w^R + S * M - \gamma^k$

②

Pre-fix

$$\underline{T/W^R + S * M - Y^k}$$

$$+ \underline{T/W^R} \quad \underline{S * M - Y^k}$$

$$+ / \underline{T W^R} - \underline{S * M} \cdot \underline{Y^k}$$

$$+ / T^W R - * S M^Y k$$

post-fix

$$\underline{T/W^R + S * M - Y^k}$$

$$\underline{T/W^R} \quad \underline{S * M - Y^k} \quad +$$

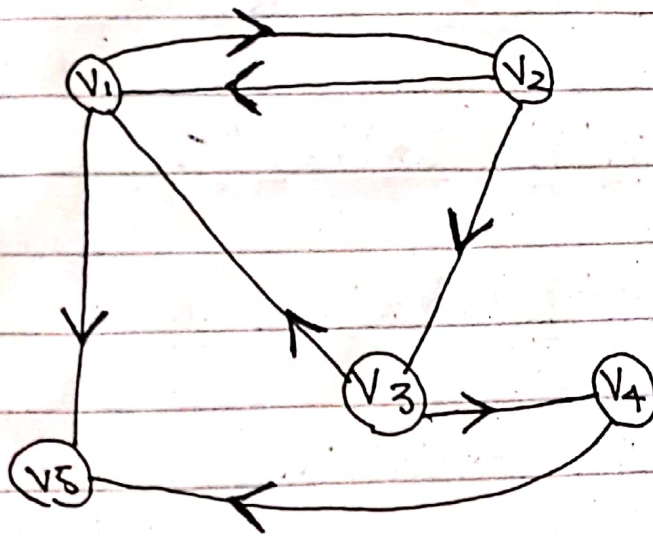
$$\underline{T/W^R} \quad S$$

$$\underline{T/W^R} / \underline{S * M} \underline{Y^k} - +$$

$$T W R^A / S M * Y k^A - +$$

3

ANSWER#4



No. of nodes = $m = 5$
order of $A = M \times M$
 $= 5 \times 5$
 $= 25$

	v_1	v_2	v_3	v_4	v_5	out degree
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
In-degree	2	1	1	1	2	7

In-degree

4

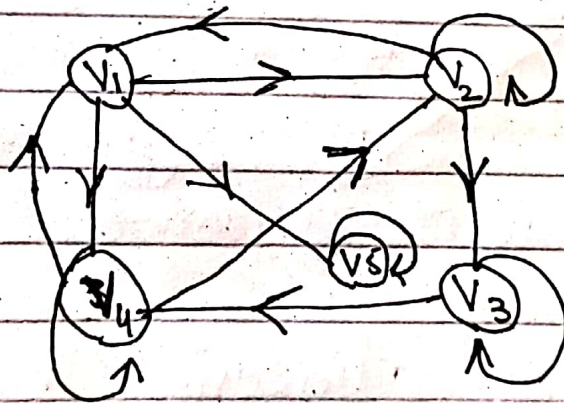
ANSWER# 5

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

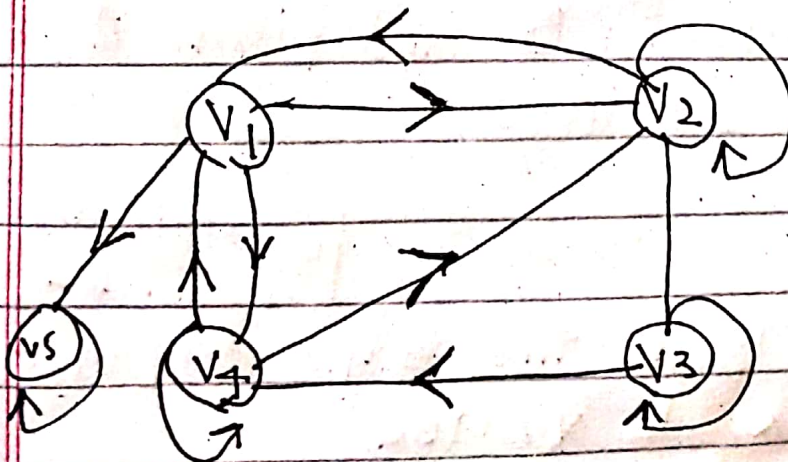
order of $A = m \times m$
 $= 5 \times 5$
 $= 25$

No of nodes = 5

The nodes be V_1, V_2, V_3, V_4, V_5

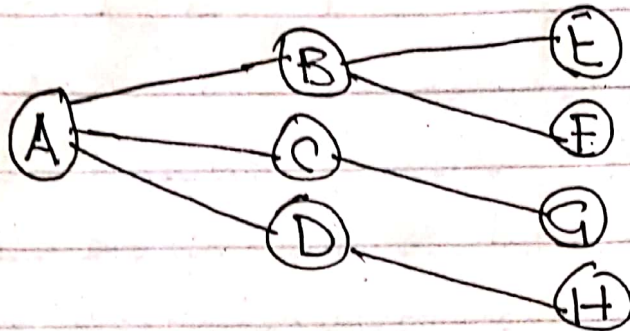


OR



5

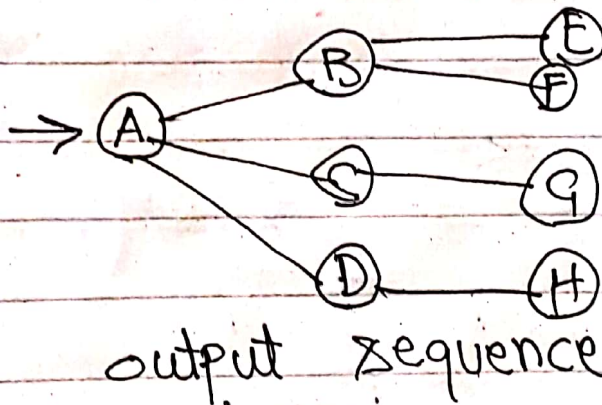
ANSWER#3



(i) Add Root A to the output sequence

* Mark A visited

* A is CN



(ii) A is adjacent to B, C and D

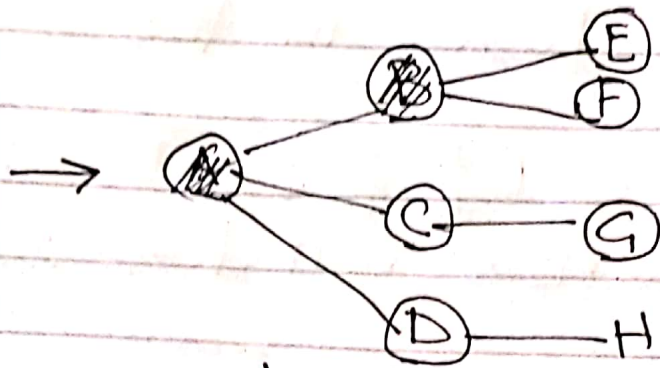
* Select B and push it into sequence



* Add B to the output sequence

* Mark B visited

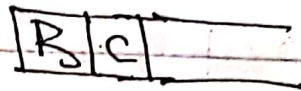
6



output sequence
A, B

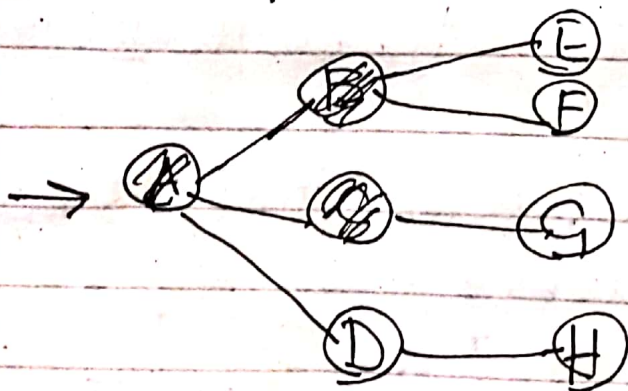
iii) From root i.e 'A' the adjacent node is 'C'

* 'C' is pushed into the queue



* 'C' is marked visited

* 'C' is added to output sequence.

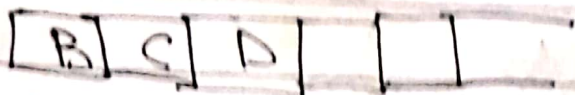


output sequence
A, B, C

iv) As 'D' is also adjacent to 'A'

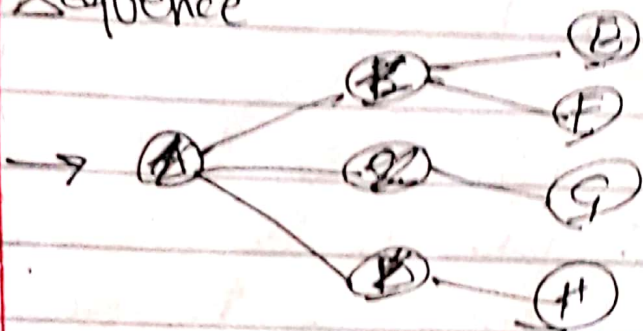
7

* D is pushed into the queue



* D is mark visited

* D is added to the output sequence

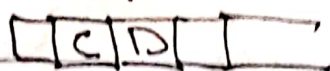


output sequence ABCD

* Now CWN is updated

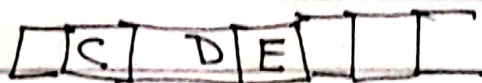
* 'B' is selected as new CWN

* 'B' is popped from queue



⊙ A B is adjacent to E and F

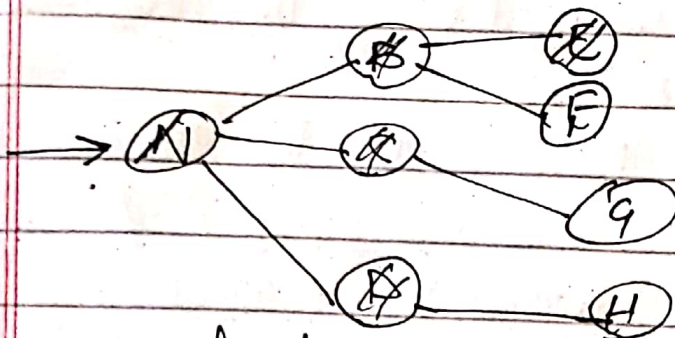
* 'E' is selected and pushed into the queue



* 'E' is marked visited

* 'E' is added to output sequence

8



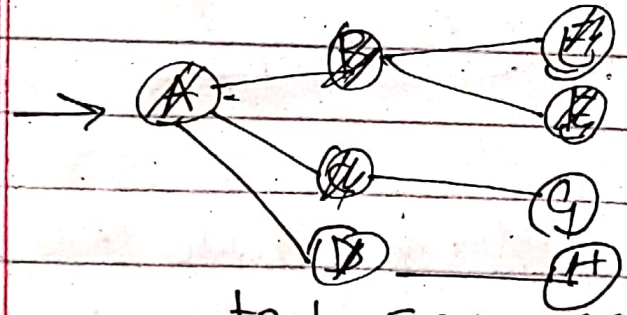
output sequence
A, B, C, D, E

vi From curN i.e B' the adjacent node 'F' is selected

* F is pushed into the queue



* 'F' is mark visited
sequence



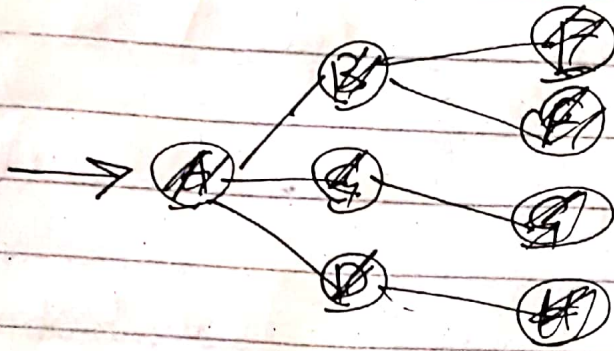
output sequence
A, B, C, D, E, F

* Now curN is updated to

* 'C' is popped queue



10



O.S

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

* Now CWN is updated to 'E'

* 'E' is popped from Queue



* No adjacement node to 'H'

* Now again CWN is updated to 'F'

* 'F' is popped from Queue



* No adjacent node to 'G'

* Now again CWN is updated to 'G'

* 'G' is popped from Queue



II

X. No adjacent node to 'G'

X. Now again own is updated to 'H'

X. H is popped from Queue



X. No of adjacent nodes to 'H'

X. Queue is empty, so BFS stops