

NAME

Muhammad Amir

Id

14633

Department

BSCS

Semester

4th

Subject

Design & Analysis of Algo-
-rithm

Date

24/06/2020

Teacher

SIR Muhammad Adil.

Q1

Fill in the blanks.

- i- Vertex
- ii- Multiple / parallel edge
- iii- Adjacent edges
- iv- Simple path
- v- Cycle
- vi- Source node
- vii- Sink
- viii- Isolated OR Null graph
- ix- Regular graph
- x- Labeled graph.

Q.2

i) Pre-fix & Post-fix

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

Pre-fix:-

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

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

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

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

Post-fix:-

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

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

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

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

ii) $T/W^R + S^*M - Y^N K$

Pre-fix:-

T/W^R + S^{*}M - Y^NK

+ T/W^R S^{*}M - Y^NK

+ T/W^R - S^{*}M Y^NK

+ / T W^R - S^{*}M Y^NK

+ / T^N WR - S^{*}M Y^NK

+ / T^N WR - *SM Y^NK

+ / T^N WR - *SM^N YK

Post-fix:-

T/W^R + S^{*}M - Y^NK

T/W^R S^{*}M - Y^NK +

T/W^R S^{*}M Y^NK - +

T W^R / S^{*}M Y^NK - +

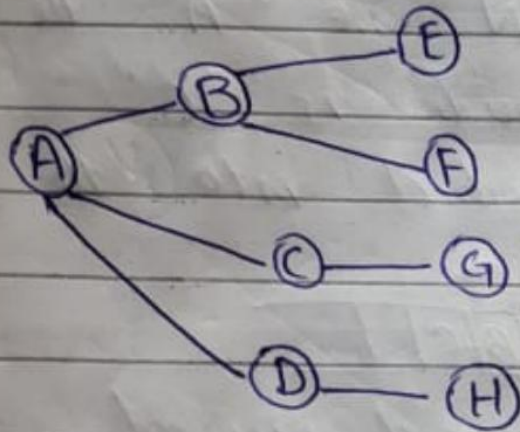
T W^R / S^{*}M Y^NK - +

T W^R / SM^{*} Y^NK - +

TWR^N / SM^{*} YK^N - +

Q.3

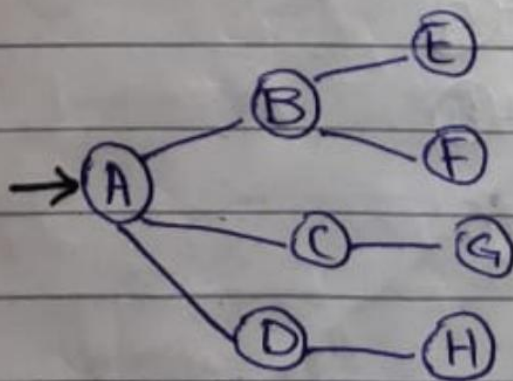
Breadth-First Technique :-



① * Add root a to the output sequence

* Mark A visited.

* A is CWN.



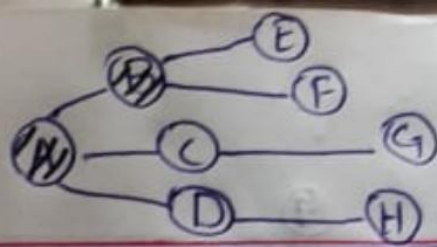
output sequence:

A

② * A is adjacent to B, C and D.

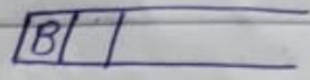
* select B and push it into queue.

2



* Add B to the output sequence.

* Mark B visited



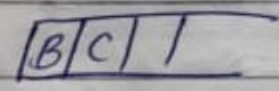
Output Sequence:

A, B

3

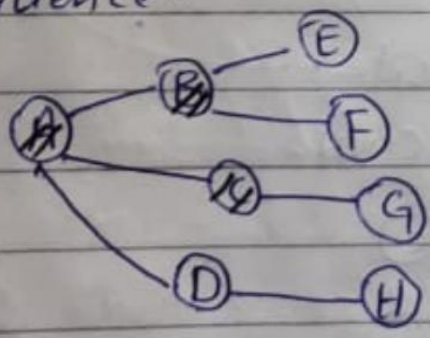
* From CWN i.e., 'A' the adjacent node is 'C'

* 'C' is pushed into the queue.



* 'C' is marked visited

* 'C' is added to output sequence.



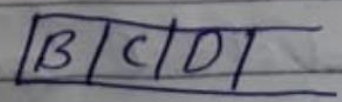
output sequence:-

A, B, C

4

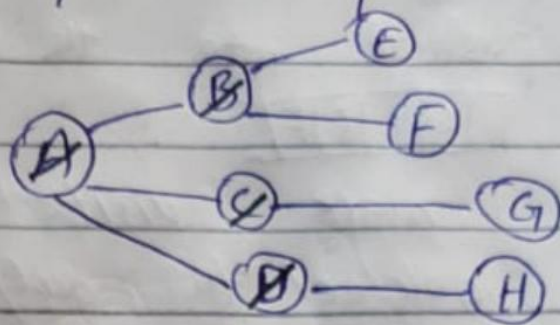
* As 'D' is also adjacent to 'A'

* D is pushed into the queue



* D is mark visited

* D is added to the output sequence.



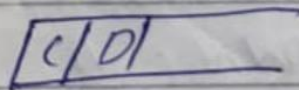
output sequence :-

A, B, C, D.

* Now CWN is updated

* 'B' is selected as new CWN.

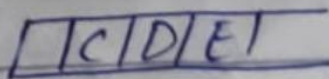
* 'B' is popped from queue.



⑤

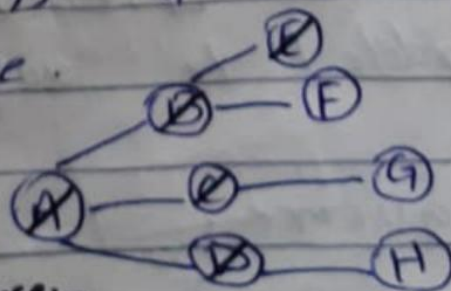
* B is adjacent to E and F

* E is selected and pushed into the queue



* E is marked visited

* 'E' is added to output sequence.

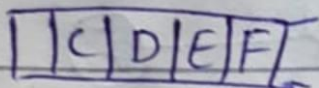


output sequence :-

A, B, C, D, E.

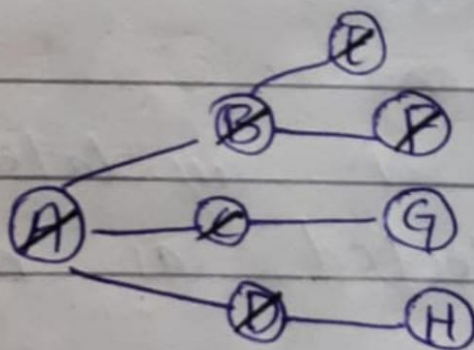
⑥ * From the CWN i.e. B the adjacent node F is selected.

* F is pushed into the queue



* F is mark visited.

* F is added to output sequence.

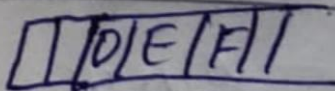


output sequence:-

A, B, C, D, E, F

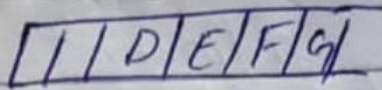
* Now CWN is updated to C.

* C is popped from queue



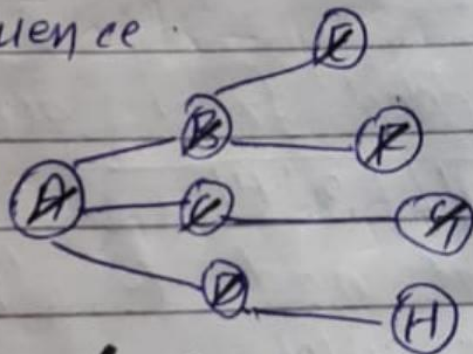
⑦ * From CWN i.e 'C', the adjacent node is G.

* G is pushed into the queue.



* G is marked visited.

* G is added to output sequence.

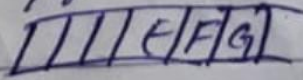


output sequence:-

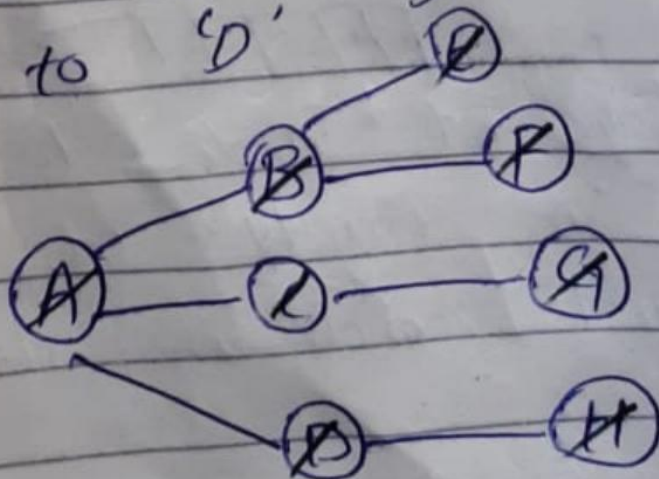
A, B, C, D, E, F, G

* Now CWN is updated to 'D'

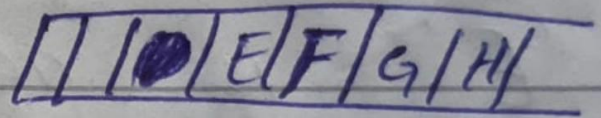
* 'D' is popped from queue



⑧ * 'H' is adjacent node to 'D'



- * H is pushed to queue.
- * H is mark 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 queue.

||||F|G|H|

* No adjacent node to 'H'.

* Now again CWN is updated to 'F'.

* 'F' is popped from queue.

|||G|H|

* No adjacent node to 'F'.

* Now again CWN is updated to 'G'.

* G is popped from queue.

|||||H|

* No adjacent node to G.

* Now again CWN is updated to 'H'.

* H is popped from queue.

|||||

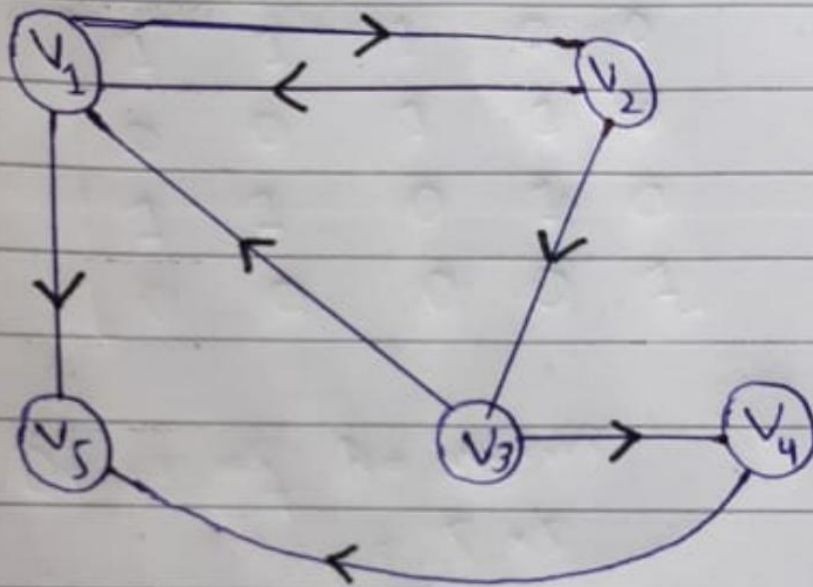
* No adjacent node to 'H'.

* Queue is empty. so,

Breadth first search is stop.

Q4

Adjacency Matrix:-



Number of nodes = $m = 5$

Order of $A = m \times m$

$= 5 \times 5$

	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

Q.5

Directed Graph:-

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

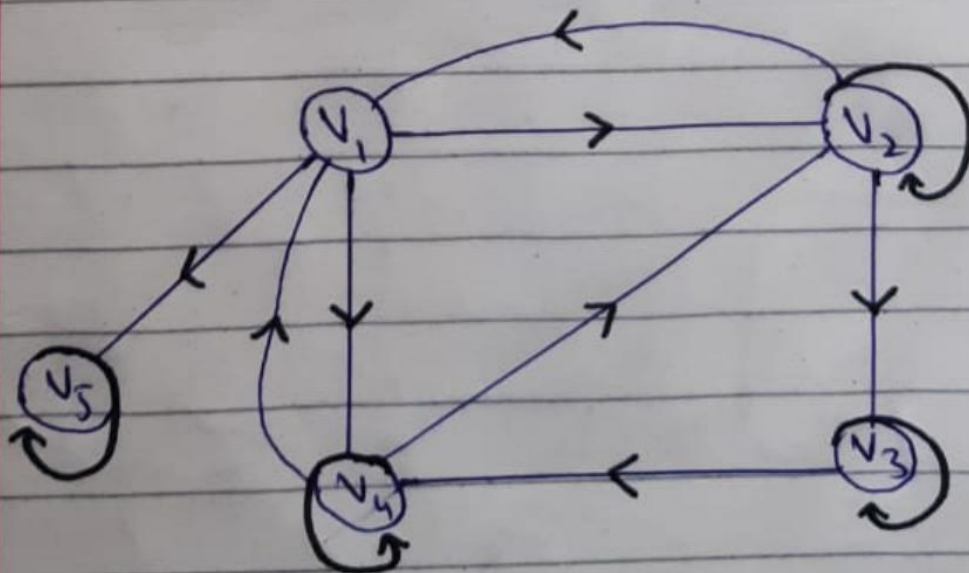
As

$$\begin{aligned} \text{Order of } A &= m \times m \\ &= 5 \times 5 \end{aligned}$$

So

$$\text{Number of nodes} = 5$$

lets the nodes be V_1, V_2, V_3, V_4, V_5



The required graph.