

①

Name	ALI Raza
ID	# 14989
Dept	# BS (cs) 4th semester
submitted To	M. Adil Sir
Subject	# Design Analysis and Algorithms

Q NO 18
Answers

- (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.

(2)

Q No 28

Answer: (i) $D - Y * (F/G)$

Conversion
prefix Notations

$$\begin{aligned} & \underline{D} - \underline{Y} * \underline{(F/G)} \\ & = - \underline{D} \underline{Y} * \underline{(F/G)} \\ & = - D * Y (F/G) \\ & = - D * Y (/ F G) \text{ Ans} \end{aligned}$$

Post-fix Notation

$$\begin{aligned} & \underline{D} - \underline{Y} * \underline{(F/G)} \\ & = D Y * (F/G) - \\ & = DY (F/G) * - \\ & = DY (FGI) * - \text{ Ans} \end{aligned}$$

(3)

$$(ii) \quad T/W \wedge R + S * M - Y \wedge K.$$

Convention:-

prefix Notation.

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

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

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

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

Post-fix Notation

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

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

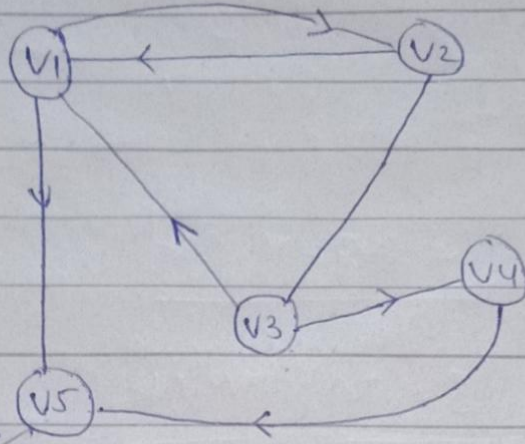
$$= T \underline{W} \wedge \underline{R} / \underline{S} * \underline{M} \underline{Y} \wedge \underline{K} - +.$$

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

(4)

Q 4g

Ans



	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
						<hr/> 7

NO of Nodes = $m = 5$
order of $A = M \times M$
 $= 5 \times 5$
 $= 25$

(5)

(i) NO 5e-

Answer

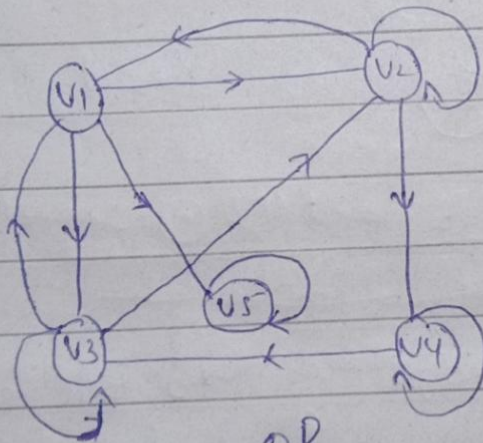
$$A[] = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

As

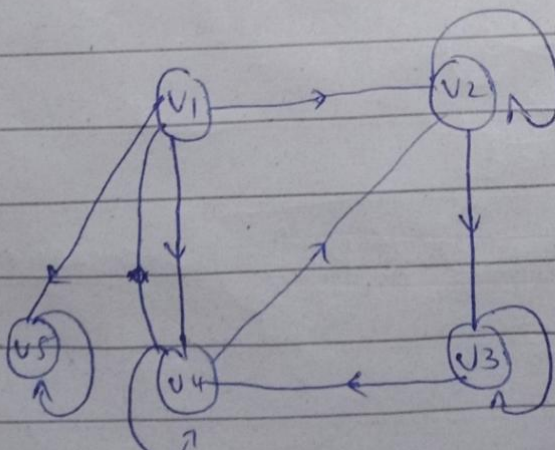
$$\begin{aligned} \text{order of } A &= m \times m \\ &= 5 \times 5 \\ &= 25 \end{aligned}$$

So No of nodes = 5

lets the nodes be v_1, v_2, v_3, v_4, v_5 .



OR

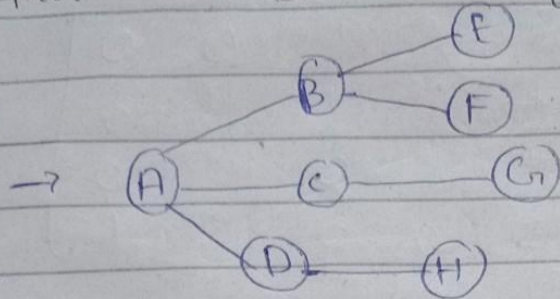


The required graph.

(6)

Q38

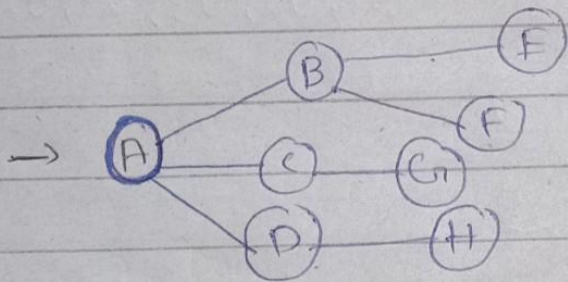
Ans: Breadth Techniques



(1) Root "A" is current working Node (CWN).

(*) Mark "A" visited.

(*) Add "A" to the output sequence.

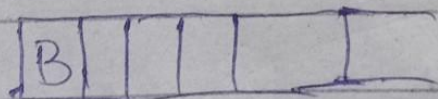


Output sequence:

A,

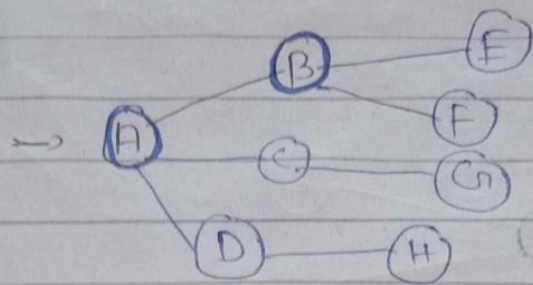
(2) (*) A is adjacent to B, C and D

(*) Select "B" and push it into Q.



(*) Mark "B" visited

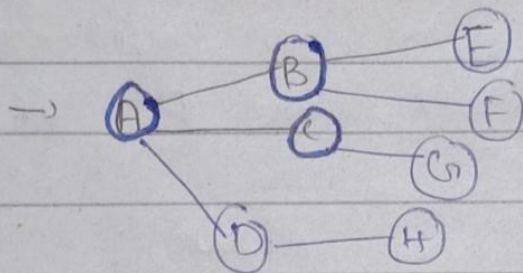
(*) Add "B" to the output sequence



Output sequence →

(3) (*) Accessing "C" from CWN i.e "A".

(*) Add "C" to the output sequence



Output sequence →

A, B, C

(4) (*) From CWN i.e "A" the adjacent node "D" is selected.

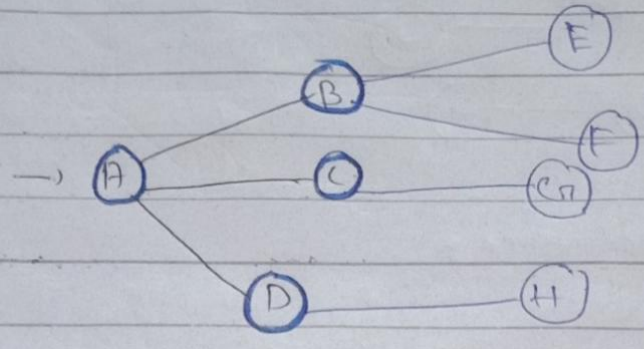
(*) "D" is pushed into the Q.

[B | C | D | | |]

(*) "D" is marked visited

(8)

* "D" is added to the output sequence.



Output Sequence

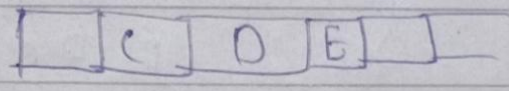
A, B, C, D

* Now as there are no more nodes adjacent to CWN i.e. A so update CWN.

* Select "B" as CWN.

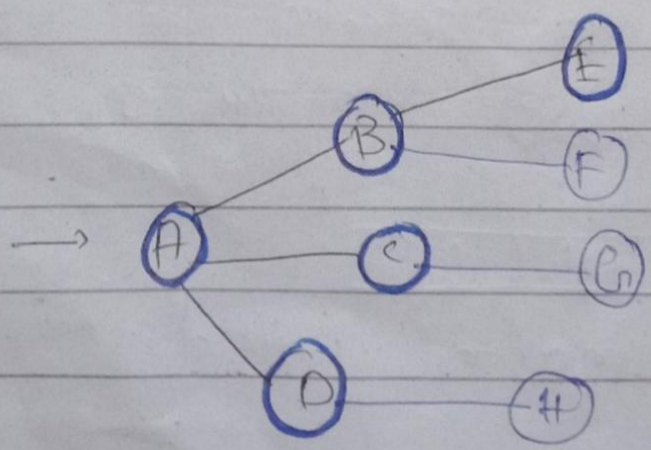
* Pop it from Q.

* B is adjacent to (E) and (F) * Select (E) and push it into Q.



* Add "E" to the output sequence.

* Mark "F" visited.

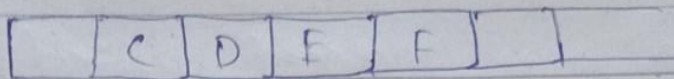


9

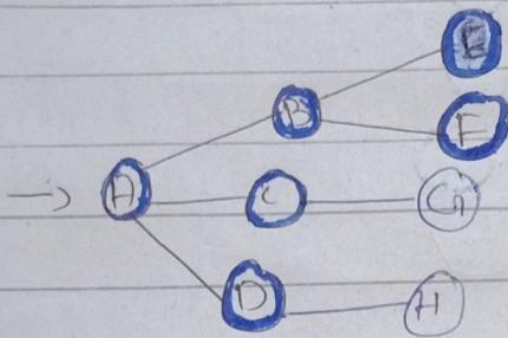
Output sequence

A, B, C, D, E

- ⑥ ⊕ From CWN i.e "B" access "F"
- ⊕ Push "F" into Q.



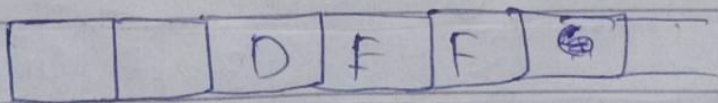
- ⊕ Mark "F" visited.
- ⊕ Add "F" to the output sequence.



Output sequence:-

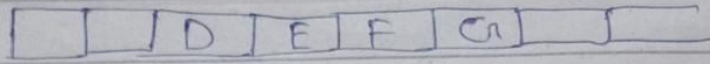
A, B, C, D, E, F

- ⊕ As there are no more nodes adjacent to CWN i.e "B", so update CWN i.e "B", so
- ⊕ "C" is popped from "Q".



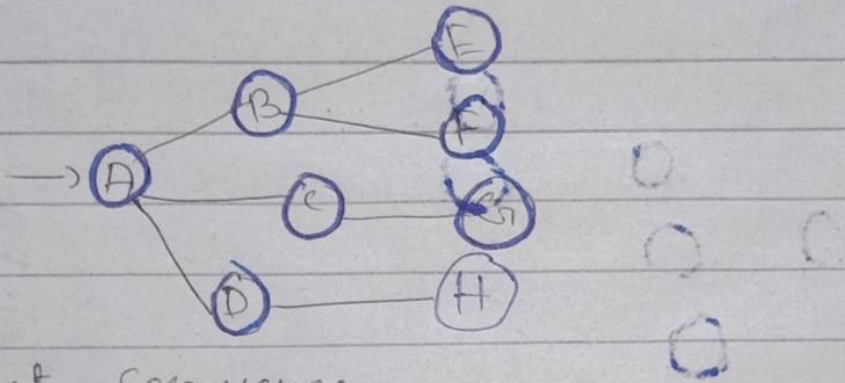
⊕ ⊕ Now "C" is adjacent to "G".

⊕ Select "G" and push it into the Q



⊕ "G" is marked visited.

⊕ "G" is added to output sequence.



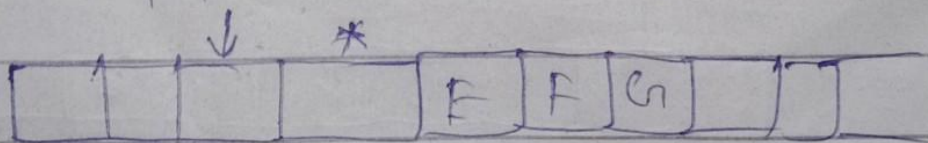
output sequence:-

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'

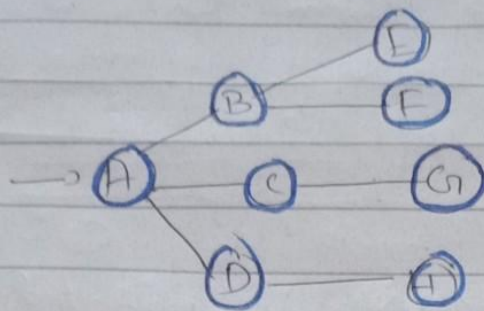
⊕ "D" is popped from "Q"



⊕ "H" is marked visited

⊕ "H" is added to output sequence

(11)

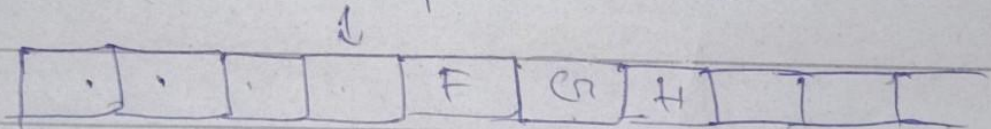


output sequence

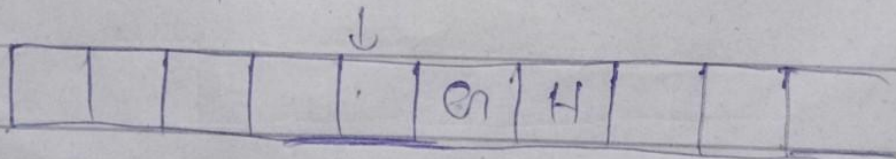
A, B, C, D, E, F, G, H.

Output sequence:-

- ⊕ Now CWN is updated to "E".
- ⊕ "E" is ~~added~~ popped from Q.



- ⊕ NO adjacent node to "E".
- ⊕ Again CWN is updated to "F".
- ⊕ "F" is popped from Q.

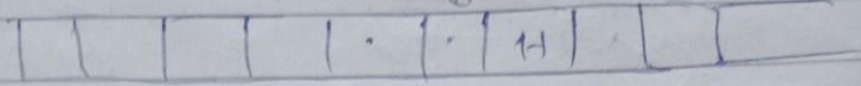


- ⊕ NO adjacent node to "F".

(12)

⊕ Now again cwn is updated to "G".

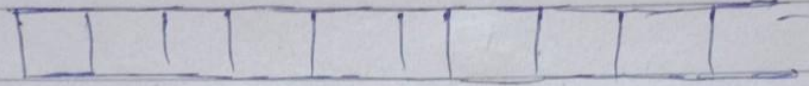
⊕ "G" is popped from Q.



⊕ No adjacent node to "G".

⊕ Now again cwn is updated to "H".

⊕ "H" is popped from Q.



⊕ Q is Now empty, so breadth First Search ~~upt~~ stops.

End

