

Date: / /

Final-term Assignment Design Analysis and Algorithms

Name: - Syed Danish Ali

I.d = 14712

Department: - BSCCS

Submitted to: - Sir Adil.

(Q.1)
(Answer)

- 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)

Syed Danish Ali

14712

Q.2
(Ans)

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

Conversion :-

Pre-fix Notation :

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

Post-fix Notation :-

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

(3)

Date: / /

14712

Syed Danish Ali

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

Conversion:-

Pre-fix Notation:-

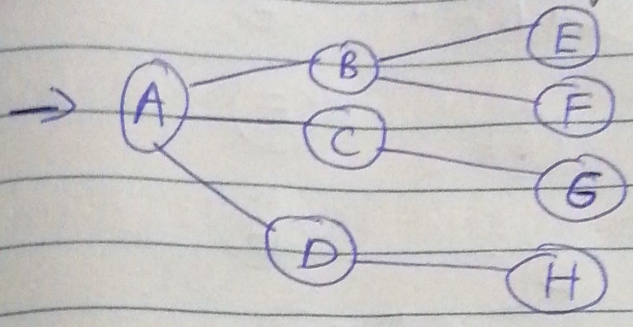
$$\begin{aligned}
 & \underline{T/W \wedge R + S * M - Y \wedge K} \\
 & = + \underline{T/W \wedge R} \underline{S * M - Y \wedge K} \\
 & = + / \underline{T \wedge W \wedge R} - \underline{S * M} \underline{Y \wedge K} \\
 & = + / T \wedge W R - * S M \wedge Y K
 \end{aligned}$$

Post-fix Notation:-

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

(Q.3)
(Answer)

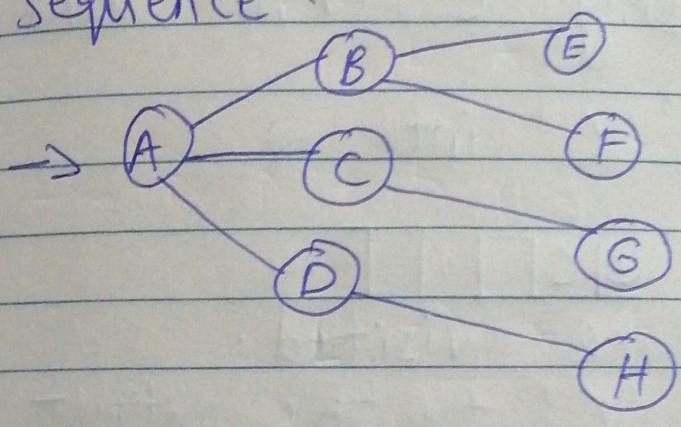
Bread First Technique:-



① (*) Root 'A' is current Working Node (CWN).

(*) Mark 'A' visited.

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

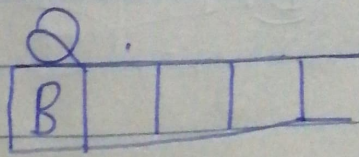


Output Sequence :

A,

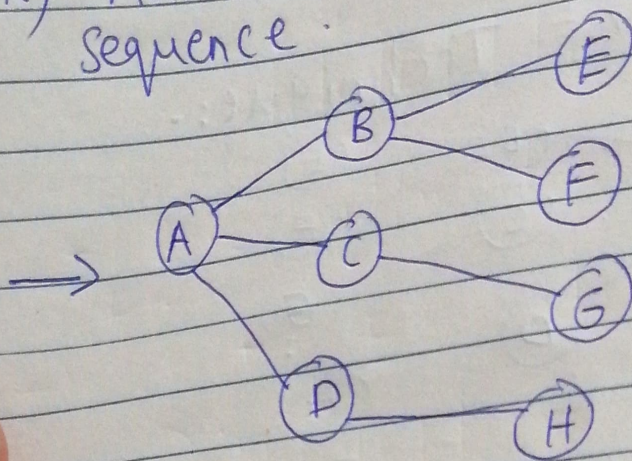
② (*) A is adjacent to B, C and D.

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



ed Danish Ali

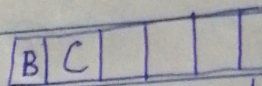
- (*) Mark 'B' visited.
- (*) Add 'B' to the output sequence.



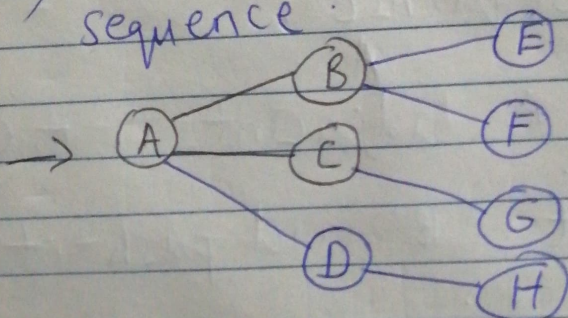
Output Sequence: A, B

- 3 (*) Accessing 'C' from CWN is 'A'.

(*) Push 'C' into Q.



- (*) Mark 'C' visited.
- (*) Add 'C' to the output sequence.



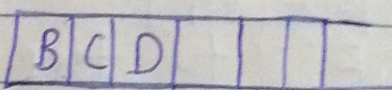
Syed Danish Ali

14712

Output Sequence :-
= A, B, C

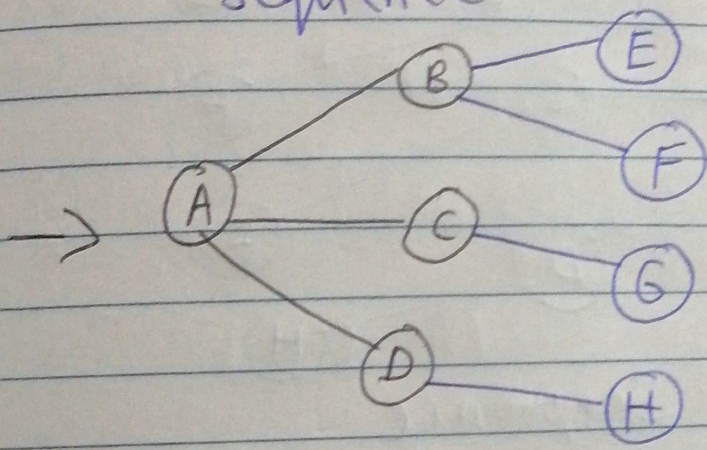
4 (*) From CWN i.e 'A', the adjacent node 'D' is selected.

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



(*) 'D' is marked visited.

(*) 'D' is added to the output sequence.



Output Sequence :-
=

A, B, C, D

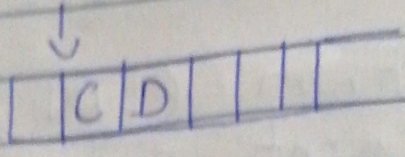
(*) Now as there are no more nodes adjacent to CWN i.e 'A' so update CWN.

(*) Select 'B' as CWN.

Syed Danish Ali

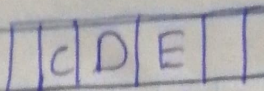
14712

(* Pop it from Q.



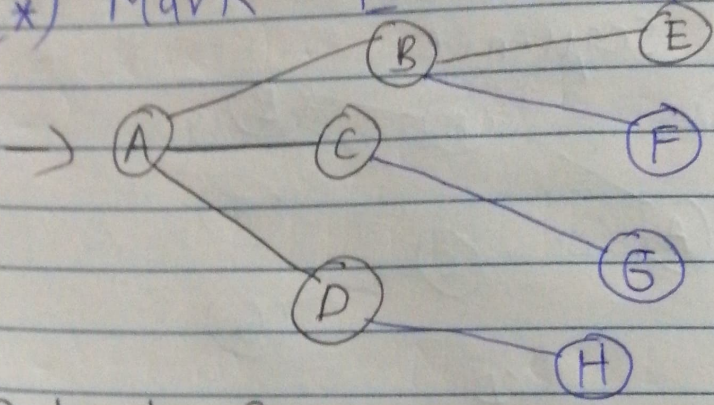
5 (*) B is adjacent to E and F.

(* Select 'E' and push it into Q.



(* Add 'E' to the output sequence.

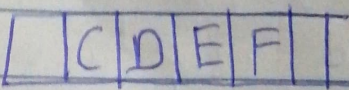
(* Mark 'E' visited.



Output Sequence: A, B, C, D, E

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

(* Push 'F' into Q.

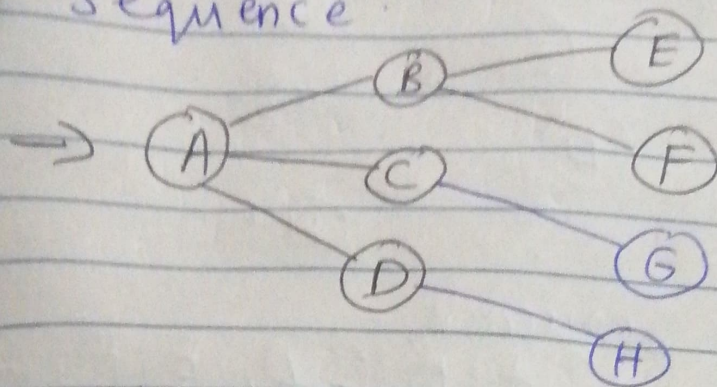


(* Mark 'F' visited.

Syed Danish Ali

14712

(* Add 'F' to the output sequence.



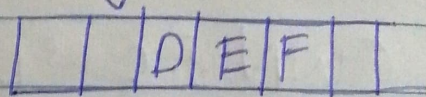
Output Sequence -

A, B, C, D, E, F

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

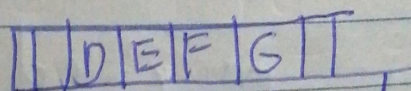
(* Select 'C' as CWN (New).

(* 'C' is popped from Q.



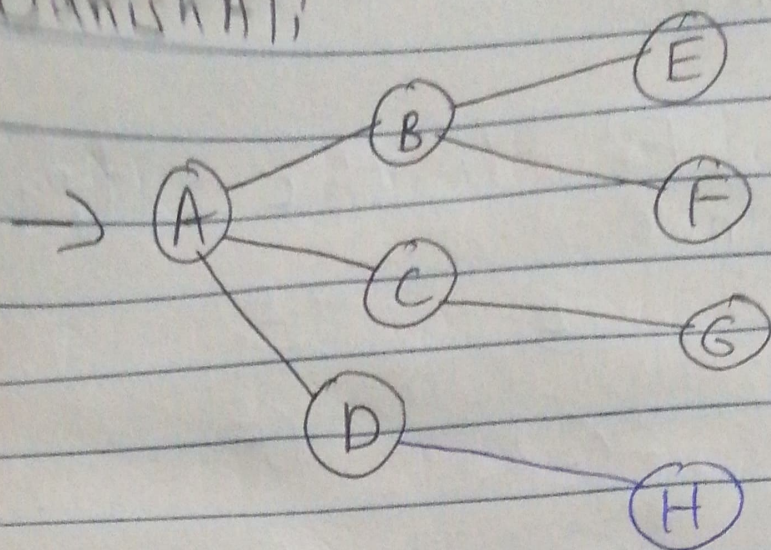
(7) (* Now 'C' is adjacent to 'G'.

(* select 'G' and push it into the Q.



(* 'G' is marked visited.

(* 'G' is added to output sequence.



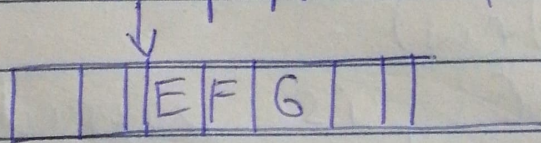
Output Sequence:-

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

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

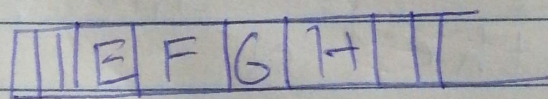
(*) 'D' is selected as new CWN.

(*) 'D' is popped from Q.



⑧ (*) From CWN i.e. 'D', adjacent node is H.

(*) 'H' is selected and is pushed into the Q.

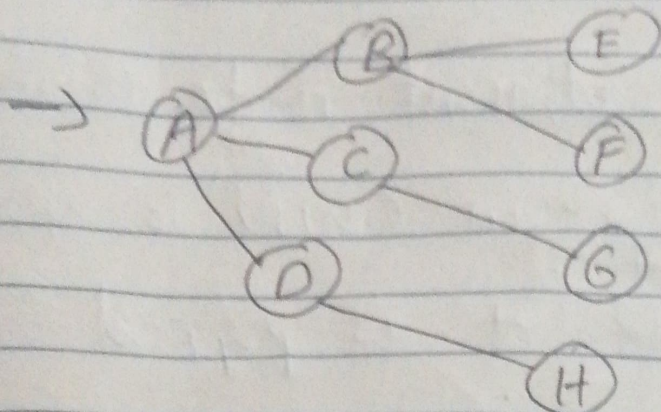


(*) 'H' is marked visited.

(*) 'H' is added to output sequence.

Syed Danish Ali

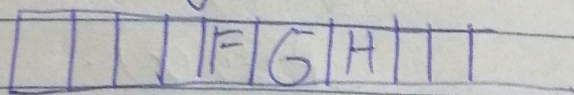
14712



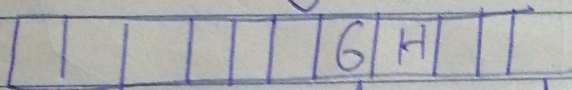
Output Sequence:-

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

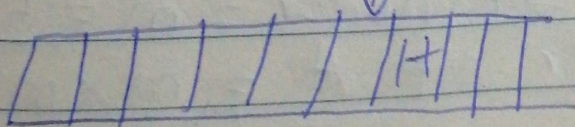
- (*) Now CWN is updated to 'E'
- (*) 'E' is popped from Q.



- (*) No adjacent node to 'E'.
- (*) Again, CWN is updated to 'F'.
- (*) 'F' is popped from Q.



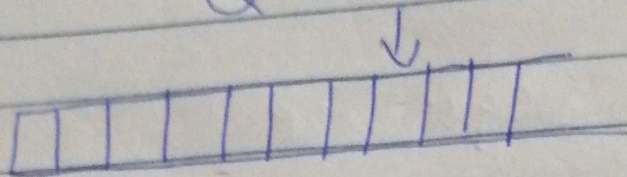
- (*) No adjacent node to 'F'.
- (*) Now again CWN is updated to 'G'.
- (*) 'G' is popped from Q.



(*) No adjacent node to 'G'

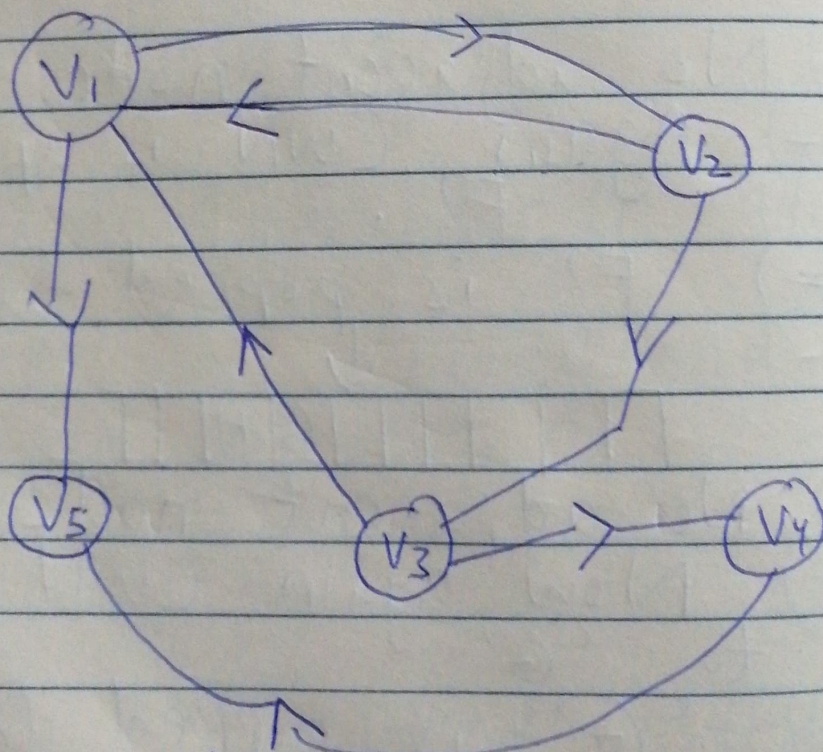
(*) Now again CWN is updated to 'H'

(*) ~~Q~~ 'H' is popped from Q.



(*) Q is now empty, so Breadth-First Search stops.

(Q.4)
(Answer)



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

Syed Danish Ali

14712

	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
						7

In-degree 2 1 1 1 2

(Q.5)

(Answer)

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

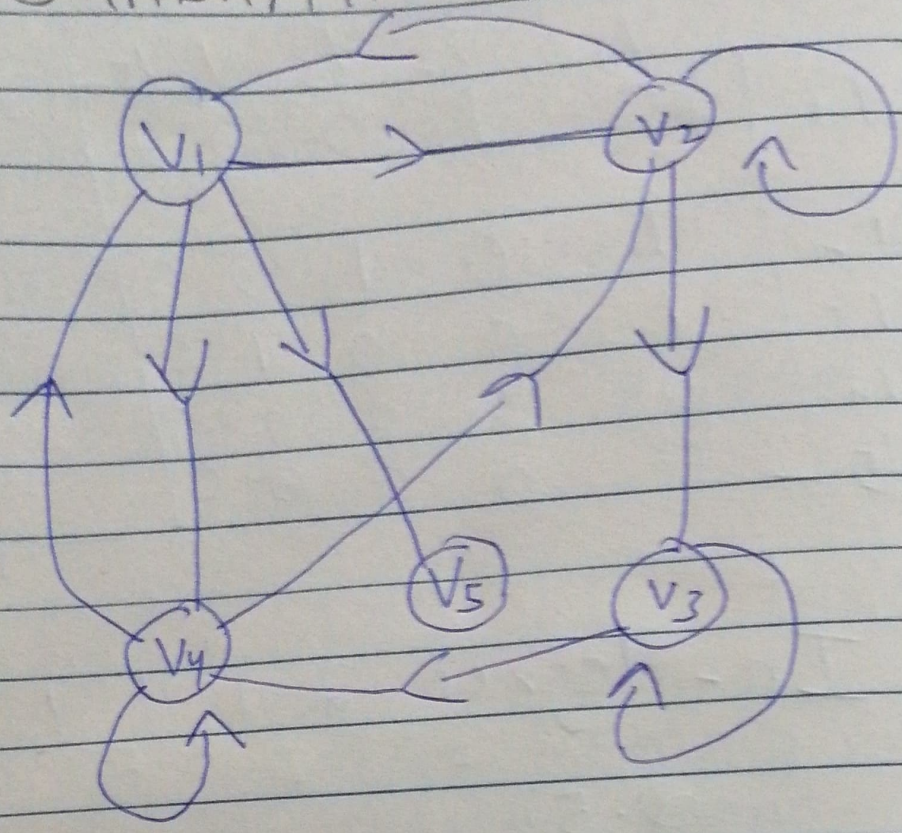
As

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

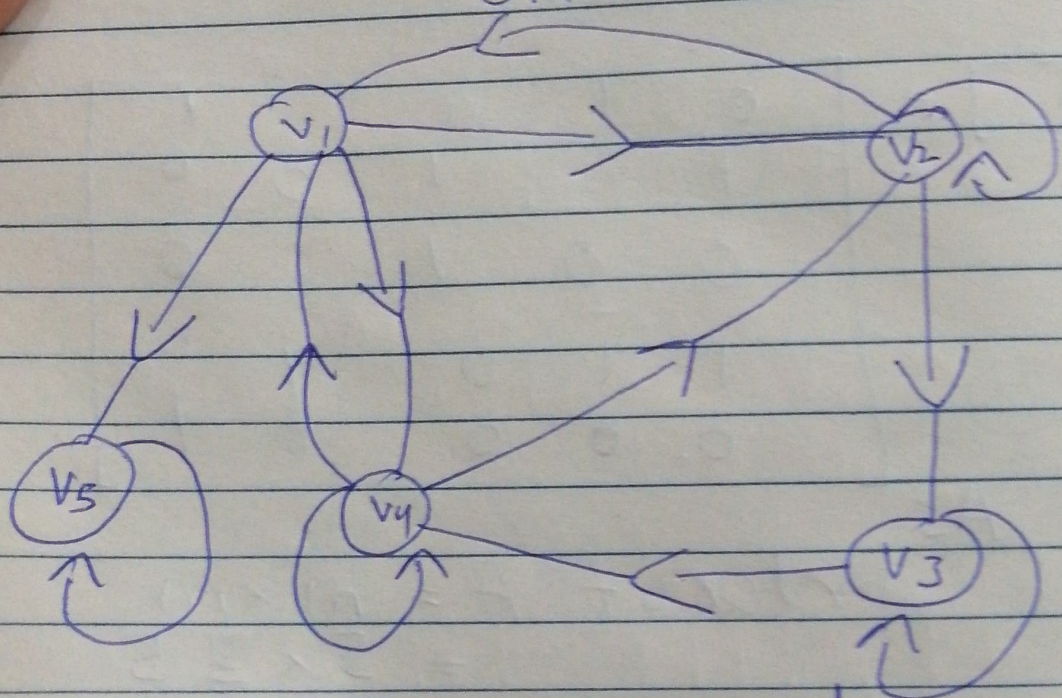
So No. of nodes = 5
 lets the nodes be V_1, V_2, V_3, V_4
 and V_5

Syed Danish Ali

14712



OR



The required graph