

①

Name

Sauood ur Rehman

ID

15031

Dept

Bc (cs) 4th semester

subject

Data Analysis & Algorithm

Assignment

Final Term

Q1

Ans

Q

- (1) vertex
- (2) Multiple / Parallel Edge
- (3) Adjacent Edge
- (4) Simple path
- (5) ~~cycle~~ cycle
- (6) Source Node
- (7) Sink
- (8) Solvable / Null graph
- (9) Regular Graph
- (10) Labeled Graph

2

Q2

Ans (a)

(i)

$D - Y * (F / G)$

Pre fix notation

$D - Y * (F / G)$

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

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

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

~~(b)~~

Post fix notation

$D - Y * (F / G)$

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

$D Y (F / G) * -$

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

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

3

~~(i)~~ (ii)

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

Prefix Notation

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

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

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

$$+ / T^N W^R - * S M^N Y K \quad \underline{\text{Ans}}$$

Post fix Notation

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

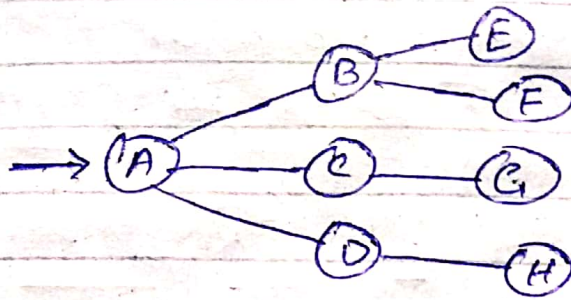
$$\underline{T/W^R} \underline{S * M - Y^N K} +$$

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

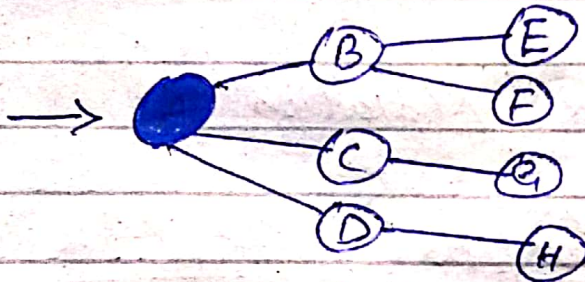
$$T W^R / S M * Y K^N - + \quad \underline{\text{Ans}}$$

Q3

Ans w



- ① * Root 'A' is current working Node
- * Mark 'A' visited
- * Add 'A' to the output sequence

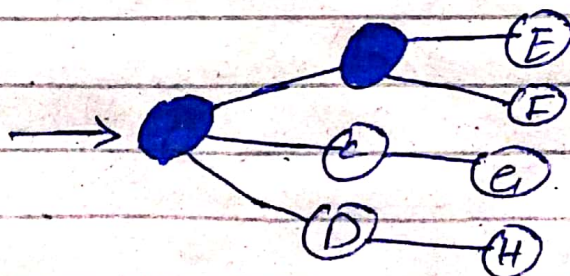


output sequence :

- 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

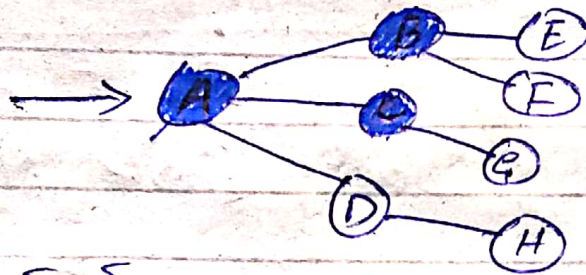


output sequence is A, B

- (3) * Accessing 'c' from CWN is 'A'
 * Push 'c' into form Q

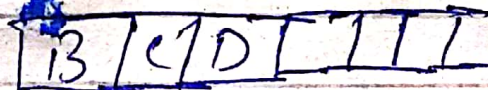


- * Mark 'c' visited
 * Add 'c' to the output sequence

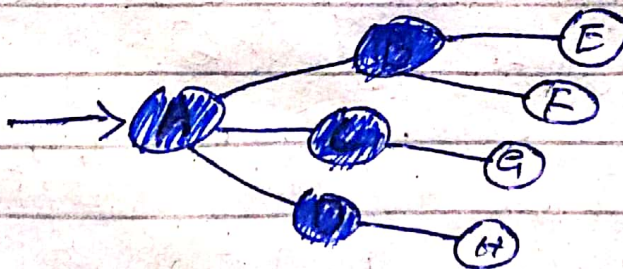


output sequence is
 A, B, C

- (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



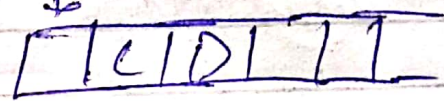
output sequence is
 A, B, C, D

⑥

* Now as There are no more nodes adjacent to curr i.e 'A' so update curr.

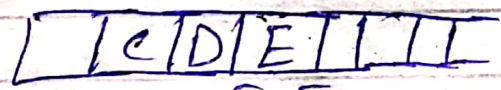
* select 'B' as curr

* Pop it from Q



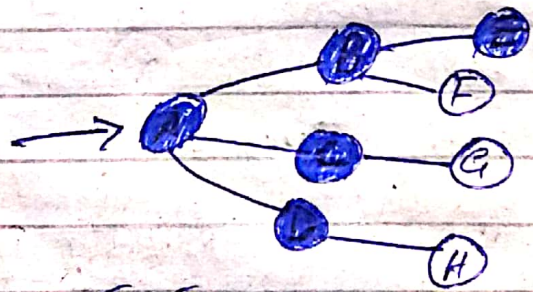
⑦ * B is adjacent to E and F

* select 'E' and push it into Q



* Add 'E' to the output sequence

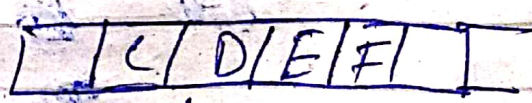
* Mark 'E' visited.



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

⑧ * From curr i.e 'B' access 'F'

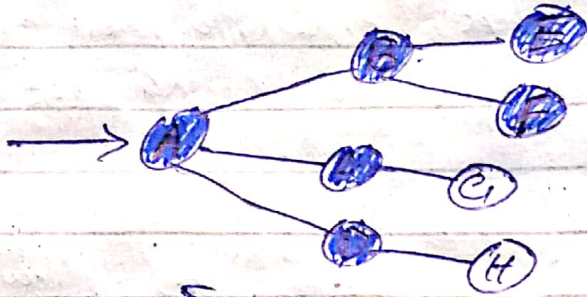
* Push 'F' into Q



* Mark 'F' visited

* Add 'F' to the output sequence.

(7)



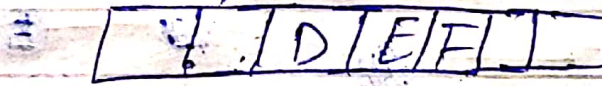
output sequences

A, B, C, D, E, F

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

* select 'C' as cnw (New)

* 'C' is popped from Q



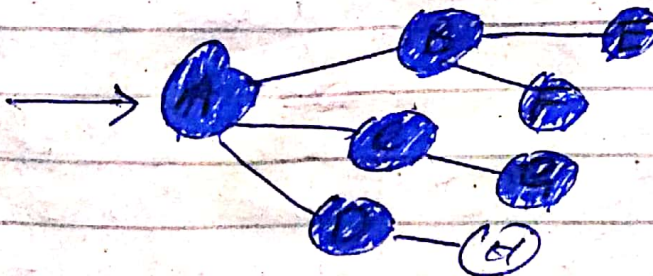
(7) * Now 'C' is adjacent to 'G'

* select 'G' and push it into the Q



* 'G' is marked visited

* 'G' is added to output sequences.



output sequence is

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

8

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

* 'D' is selected as new curr.

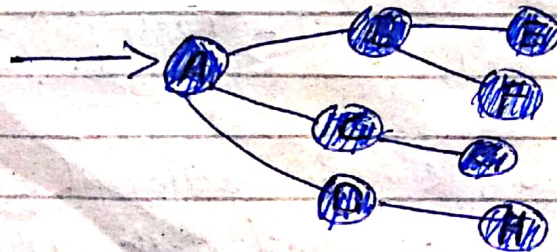
* 'D' is popped from Q

[| | E | F | D |]

9 From curr i.e 'D' adjacent node is H.

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

[| | E | F | H |]



output sequence

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

* Now curr is updated to 'E'

* 'E' is popped from Q

[| | F | G | H |]

* No adjacent node to 'E'

* Again curr is updated to 'F'

* 'F' is popped from Q

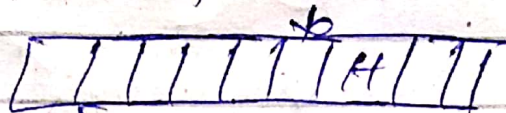
[| | G | H |]

9

* No adjacent node to 'F'

* Now again curr is updated to 'e'

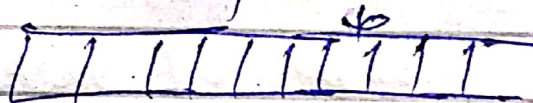
* 'e' is popped from Q



* No adjacent node to 'e'

* Now again curr is updated to 'H'

* 'H' is popped from Q

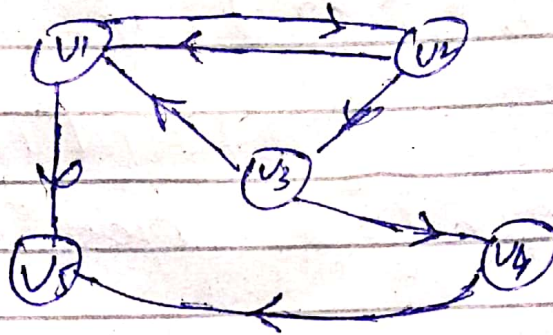


* Q is now empty so Breadth First search stops.

(10)

Q4

Ans 5 Adjacency Matrix



In this graph

Number of nodes = $m = 5$

Order of $A = m \times m$
 $= 5 \times 5$

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

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

(11)

$a_{14} = 0$; As There is no edge from v_1 to v_4

$a_{15} = 1$; As There is no edge from v_1 to v_5

$a_{21} = 1$; As There is no 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 no 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 no 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 no 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

(12)

$a_{44} = 0$: As There is no edge from v_4 to v_5

$a_{45} = 1$: As There is no 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

or

$a_{54} = 0$: 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

Hence

	v_1	v_2	v_3	v_4	v_5	outdegree	
A_2	v_1	0	1	0	0	1	2
v_2	1	0	0	1	0	0	2
v_3	1	0	0	0	1	0	2
v_4	0	0	0	0	0	1	1
v_5	0	0	0	0	0	0	0
Indegree	2	1	1	1	2	(7)	

which is required Adjacency

Matrix

13

Q5

Ans Q5

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

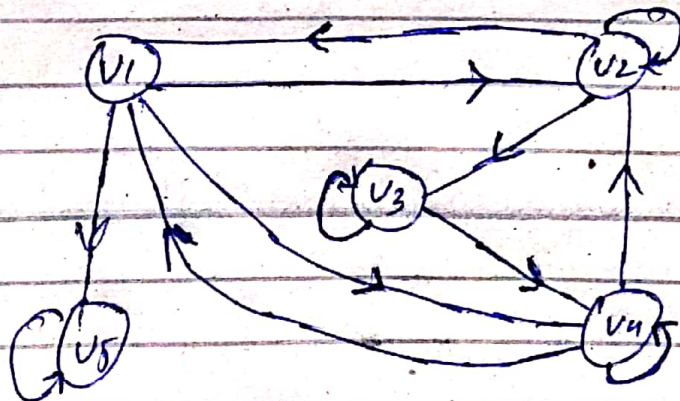
As order of $A = m \times m$
 $= 5 \times 5$

So

Number of nodes = 5

Let the nodes be v_1, v_2, v_3, v_4 and v_5

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



which is the required Directed Graph.