Name : *Muhammad Musa*

Department : *BS(CS)*

Semester : *4th*

ID # : *15366*

Assignment : *Final Term Assignment*

Subject : *Design and Analysis of Algorithm*

Submitted To : *Muhammad Adil Sir*

Dated : *24th June 2020*

Muhammad Adil Sir , BS (CS)

Name : Muhammad Musa
ID # : 15366
Subject : Design and analysis of Algorithm
Deptt : BS (CS)
Assignment : Final Term Assignment
Date : 24-06-2020

Q1.
Ans.

i. Vertex.
ii. Adjacent Vertices/Nodes
iii. Adjacent Edges
iv. Simple Path
v. Cycle.
vi. Source Node.
vii. Sink.
viii. Isolated/Null Graph.
ix. Regular Graph.
x. Labeled Graph

Q2.
Ans: (i) D - Y * (F/G)

Convertion:

Pre-fix Notation:

D - Y * (F/G)

= - D Y * (F/G)

= - D * Y (F/G)

= - D * Y (/FG) . Ans.

Post-fix Notation:

D = Y * (F/G)

= D Y * (F/G) -

= D Y (F/G) * -

= D Y (F G /) * - Ans.

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

Conversion:

Pre-fix Notation:

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

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

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

= + / T ^ W R - * S M ^ Y k Ans.

Post-fix Notation:

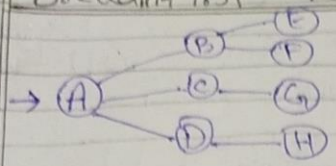
T/W ^ R + S * M - Y ^ k

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

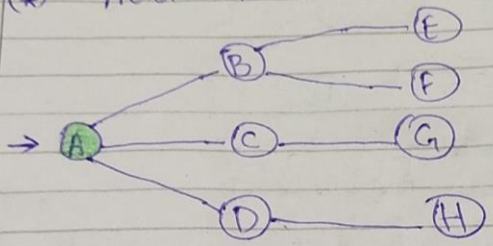
= T W ^ R / S * M Y ^ k - +

= T W R ^ / S M * Y k ^ - + Ans.

Q3:
Ans: Breadth-First Technique:

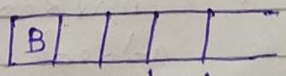


① (*) Root "A" is current Working Node (CWN).
(*) Mark "A" visited.
(*) Add "A" to the output sequence.

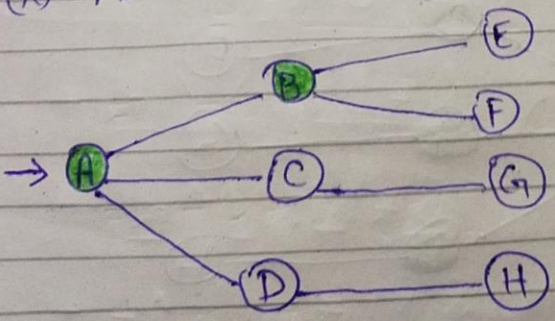


Output Sequence:
A,

② (*) A is adjacent to B, C and D.
(*) Select "B" and push it into Q



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

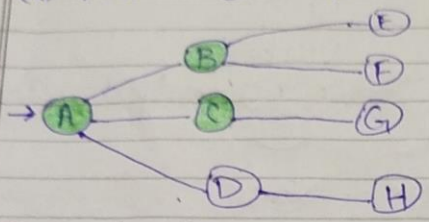


Output Sequence:
A, B

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

| B | C | | | |

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



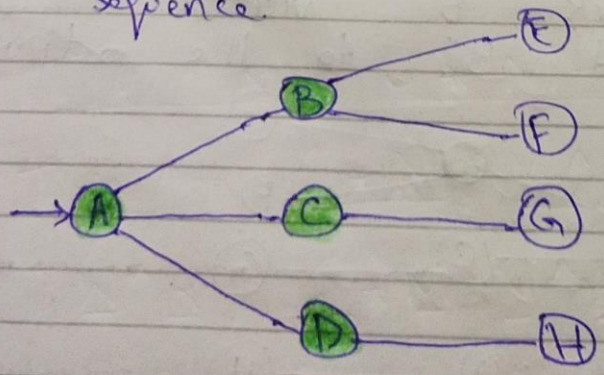
Output Sequence:
A, B, C

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

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

| B | C | D | | | |

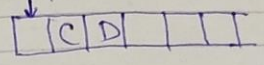
(*) "D" is marked visited.
(*) "D" is added to the output sequence.



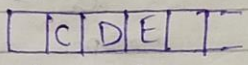
(5)

Output Sequence:
A, B, C, D

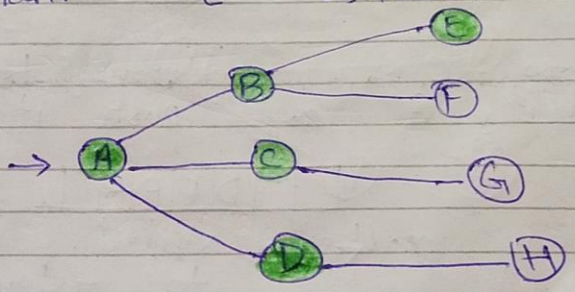
(*) Now as there are no more nodes adjacent to CWN i.e "A", So update CWN.
(*) Select "B" as CWN.
(*) Pop it from Q.



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

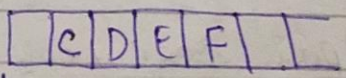


(*) Add "E" to the output sequence.
(*) Mark "E" visited.



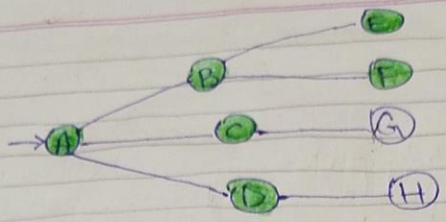
Output Sequence:
A, B, C, D, E

(6) (*) From CWN i.e "B" access "F".
(*) Push "F" into Q.



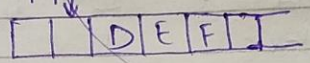
(*) Mark "F" visited.
(*) Add "F" to the output sequence.

(6)

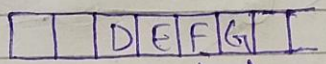


Output Sequence:
A, B, C, D, E, F

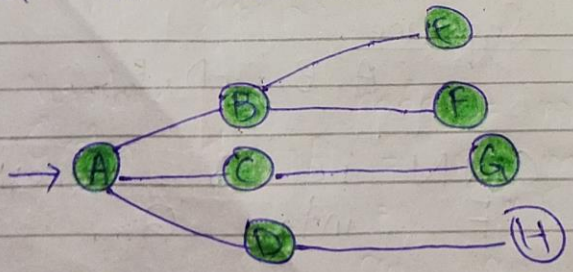
(*) As there are no more nodes adjacent to CWN i.e "B", so update CWN again.
(*) Select "C" as CWN. (New)
(*) "C" is popped from Q.



(7) (*) Now "C" is adjacent to "G".
(*) Select "G" and push it into the Q.

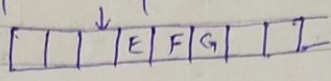


(*) "G" is marked visited.
(*) "G" is added to output sequence

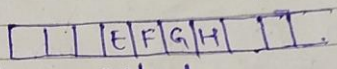


Output Sequence:
A, B, C, D, E, F, G

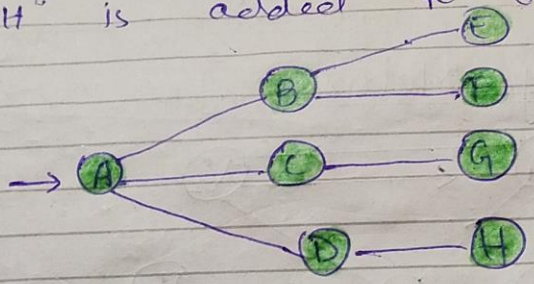
(*) Again there are no more nodes adjacent to CWN i.e "C". So update CWN.
(*) "D" is selected as new CWN.
(*) "D" is popped from Q.



(8) (*) From CWN i.e "D", adjacent node is H.
(*) "H" is selected and is pushed into the Q.

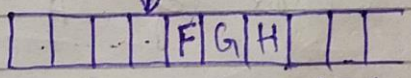


(*) "H" is marked visited.
(*) "H" is added to output Sequence.

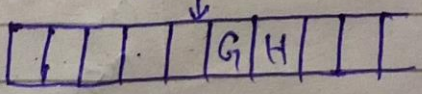


Output Sequence: A, B, C, D, E, F, G, H

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

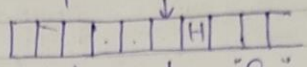


(*) No adjacent node to "E".
(*) Again CWN is updated to "F".
(*) "F" is popped from Q.



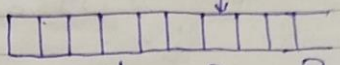
(*) No adjacent node to "F".

(*) Now again CWN is updated to "G".
(*) "G" is popped from Q.



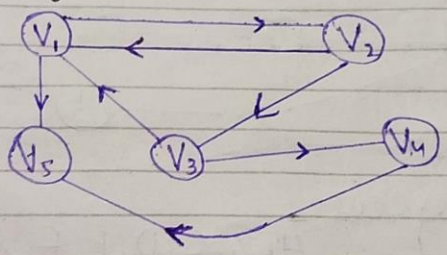
(*) No adjacent node to "G".

(*) Now again CWN is updated to "H".
(*) "H" is popped from Q.



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

Q4:
Ans: Adjacency Matrix:



In this graph:

Number of nodes = m = 5
Order of A = m × m
= 5 × 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

$a_{14} = 0$; As there is no edge from V_1 to V_4

$a_{15} = 1$; As there is an edge from V_1 to V_5

$a_{21} = 1$; As there is an 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 an 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 an 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 an 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

$a_{44} = 0$; As there is no edge from V_4 to V_4

$a_{45} = 1$; As there is an 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

$a_{54} = 0$; 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

$$A = \begin{array}{c|ccccc|c} & V_1 & V_2 & V_3 & V_4 & V_5 & \text{outdegree} \\ V_1 & 0 & 1 & 0 & 0 & 1 & 2 \\ V_2 & 1 & 0 & 1 & 0 & 0 & 2 \\ V_3 & 1 & 0 & 0 & 1 & 0 & 2 \\ V_4 & 0 & 0 & 0 & 0 & 1 & 1 \\ V_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \text{Indegree} & 2 & 1 & 1 & 1 & 2 & (7) \end{array}$$

which is required Adjacency Matrix.

Qs:-

Ans: 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

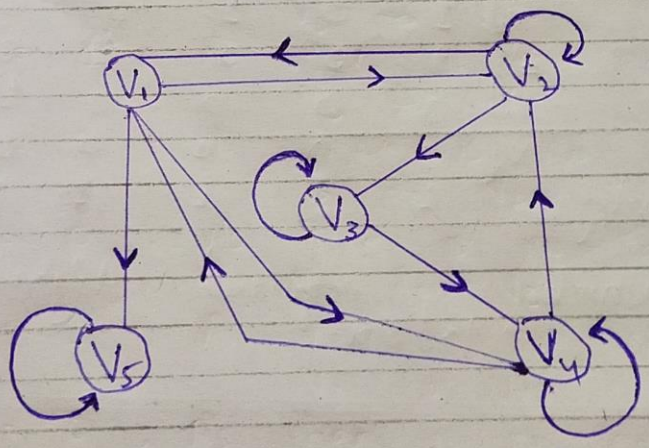
order of $A = m \times m$
$= 5 \times 5$

So

Number of nodes = 5

Let's the nodes be V_1, V_2, V_3, V_4 and V_5

A	V_1	V_2	V_3	V_4	V_5
V_1	0	1	0	1	1
V_2	1	1	1	0	0
V_3	0	0	1	1	0
V_4	1	1	0	1	0
V_5	0	0	0	0	1



Which is the required Directed Graph

<<THE END>>