<u>Final Term Exam</u>

<u>ID:</u> <u>11533</u>

<u>Name:</u> <u>Ashir Ali Khan</u>

<u>Semester:</u> <u>12th</u>

<u>Subject:</u> <u>Design and Analysis of Algorithm</u>

<u>Teacher:</u> <u>Sir Adil</u>

Q1 Fill in the blanks

i Node

ii Adjacent nodes

iii Adjacent edges

iv Shortest path

v Cycle

vi Source

vii Sink

viii Bipartite Graph

ix Regular Graph

x Weighted Graph.

2020/6/24 14:26

Q2 (i) D - Y * (F/G)

Conversion:

Pre-fix Notation:

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

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

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

$= -D * Y(/FG)$

Post-fix Notation:

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

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

$= DY(\underline{F}/\underline{G}) * -$

$= DY(FG/) * -$

2020/6/24 14:26

(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

Post-fix Notation:

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

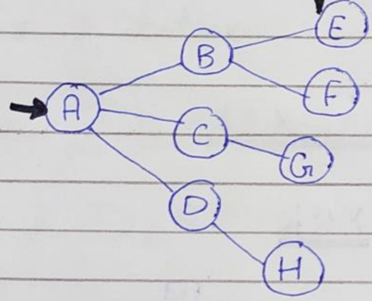
= T/W ^ R S * M - Y ^ K +

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

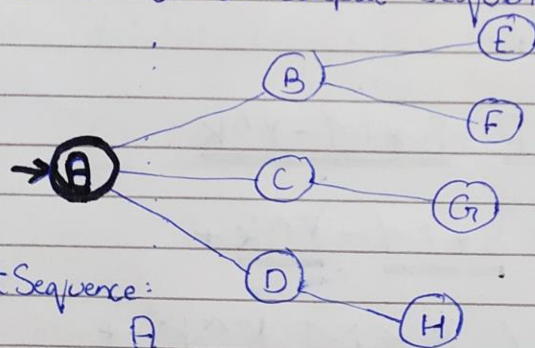
= T W R ^ / S M * Y K ^ - +

Q3 Apply Breadth First Technique on the given

Breath-First Technique:



1. • Root "A" is current working node (CWN).
   • Mark "A" visited
   • Add "A" to the output sequence.

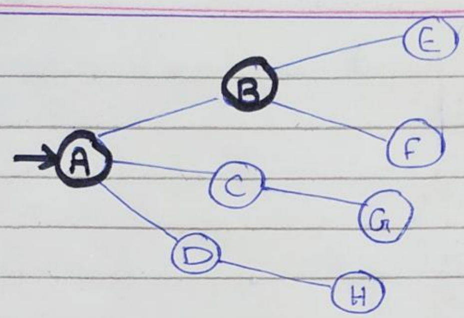


Output Sequence:
A

2. • A is adjacent to B, C and D.
   • Select "B" and push it into ~~queue~~ Q

   [B| | | |]

   • Mark "B" visited
   • Add "B" to the output sequence.



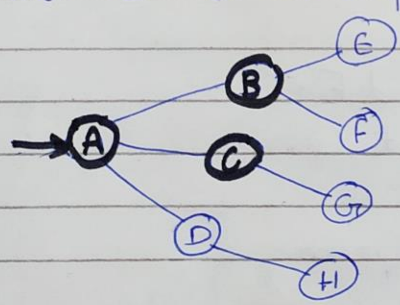
Output Sequence:

A, B

3 • Accessing "C" from CWN is "A"
• Push "C" into Q

B	C				

• Mark "C" visited.
• Add "C" to the output sequence.



Output Sequence:

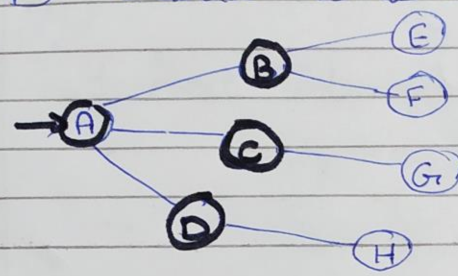
A, B, C

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.



Output Sequence:

A, B, C, D

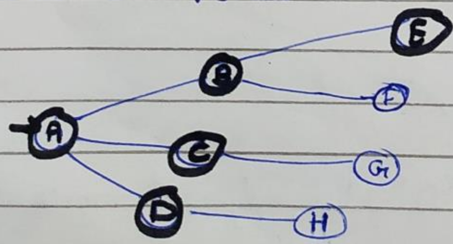
- Now as there are no more nodes adjacent to CWN i-e "A". So update CWN.
- Select "B" as CWN.
- Pop it from Q

	C	D			

- B is adjacent to E and F.
- Select "E" and push it into Q.

	C	D	E	

- Add "E" to the output sequence.
- Mark "E" visited.



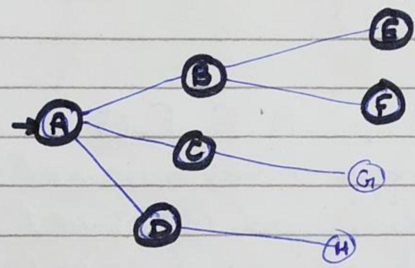
Output Sequence:

A, B, C, D, E

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

	C	D	E	F	

• Mark "F" visited.
• Add "F" to the output sequence.



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

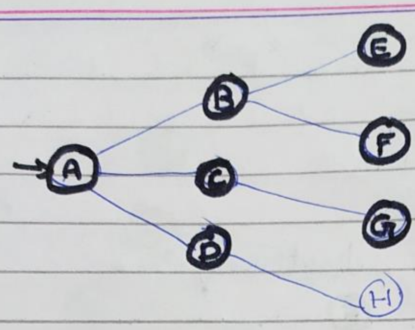
• As there are no more nodes adjacent to CWN i.e "B". So update CWN again
• Select "C" as CWN (New)
• "C" is popped from Q.

		D	E	F	

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

		D	E	F	G

• 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 ~~se~~ selected as new CWN.
- "D" is popped from the queue.

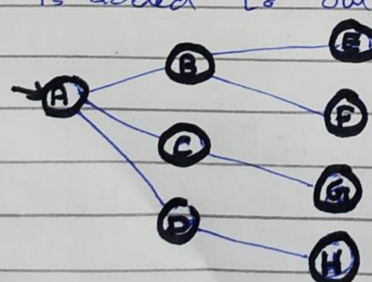
			E	F	G	

(8) • From CWN i.e "D" adjacent node is H

- "H" is selected and is pushed into the queue.

			E	F	G	H	

- "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 queue.

				F	G	H	

- No adjacent node to "E"
- Again CWN is updated to "F"
- "F" is popped from ~~queue~~ Q

.	.	.	.		G	H	

- No adjacent node to "F"
- Now again CWN is updated to "G"
- "G" is popped from ~~queue~~ Q

	.	.	.	.		H	

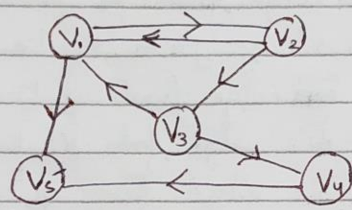
- No adjacent node to "G".
- Now CWN is updated to "H"
- "H" is popped from ~~queue~~ Q

.	.	.	.	.	.	.	.	

- Q is now empty, So Breadth First search stops.

Q4 Design Adjacency matrix for the graph

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

$a_{14} = 0$: 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

$$
\begin{array}{c|ccccc|c}
 & v_1 & v_2 & v_3 & v_4 & v_5 & \text{out degree} \\
\hline
v_1 & 0 & 1 & 0 & 0 & 1 & 2 \\
v_2 & 1 & 0 & 1 & 0 & 0 & 2 \\
v_3 & 1 & 0 & 0 & 1 & 0 & 2 \\
v_4 & 0 & 0 & 0 & 0 & 1 & 1 \\
v_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\text{Indegree} & 2 & 1 & 1 & 1 & 2 & 7
\end{array}
$$

Which is required adjency matrix.

Q5 Desig Directed graph for the given matrix

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

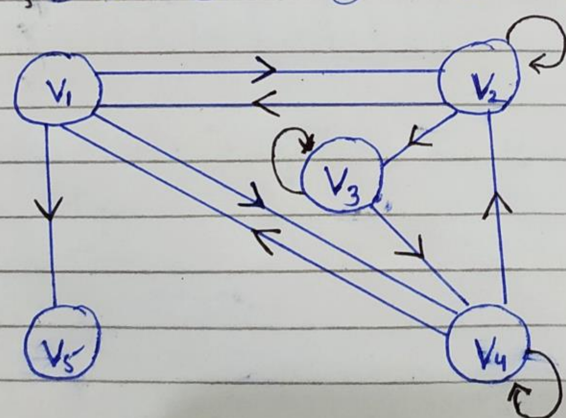
As

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

So number of nodes $= 5$

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

$$A = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{array} \begin{array}{c} \begin{array}{ccccc} V_1 & V_2 & V_3 & V_4 & V_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$



Which is the required graph: