Name Mashal Saleem

Id 14455

Subject Algorithms

Submitted to Sir Adil

Semester 4th

Deptt BSCS

# MCQS:-

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) Convert the following from infix to prefix and post-fix?

i) D-y * (F/G)

First we will convert this infix to pre-fix

pre-fix :-

We know that pre- means first so operator is placed before its corresponding operands.

Given:

D-y*(F/G)

converting to prefix:

D - y * (F / G)

Selecting - Operator as least priority and moving it to the left side. or before the operand.

- D y * (F / G) ①

Now least priority is * move it to the corresponding operand

-D y* (F/G)

-D * y (F/G) ②

Now least priority or remaining one brackets operator is to moved before operands.

-D * y (F/G)

-D * y (/FG) ③

So the prefix of given notation is -D*y(/FG).

post fix:-

Now we will convert given infix to post fix.

We know that post means after so we have to move the operator after the corresponding operands.

Conversion:-

Given

D-y * (F/G)

Selecting least priority operator - and moving it after the operands.

D - y * (F/G)

D y * (F/G) - ①

(3)

Now least priority is * moved after the corresponding operator.

D y * ( F / G ) -

D y ( F / G ) * - (2)

Now the remaining operator moved to after its corresponding operand.

D y ( F / G ) * -

D y ( F G / ) * - (3)

So the post fix of the given infix notation is D y ( F G / ) * -

ii T / W ^ R + S * M - y ^ K

prefix:-

Selecting least priority and moving it befor corresponding operand.

Given

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

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

When two operators are of both least priority then we choose the left one.

post fix:-

T / w ^ R + S * M - y ^ K

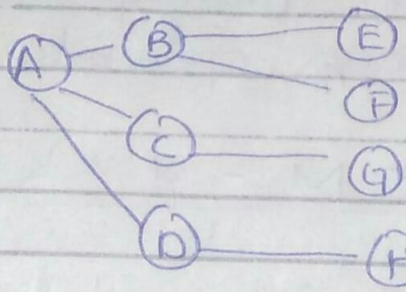
T / w ^ R S * M - y ^ K +

T w ^ R / S * M y ^ K - +

T w R ^ / S M * y ^ K - +

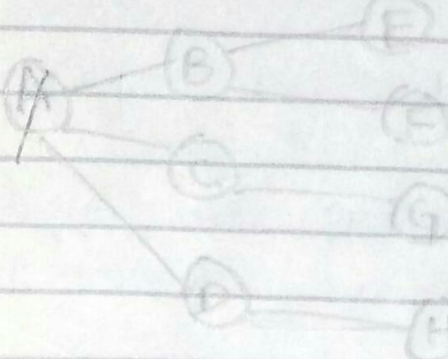
T w R ^ / S M * Y K ^ - +

3) Apply Breadth-First technique on the given tree.



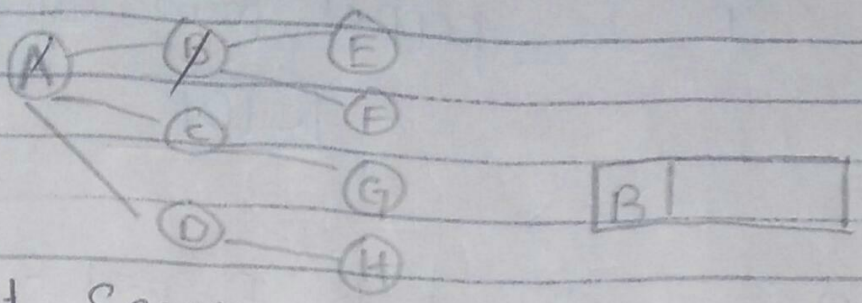

① Add root A to the output sequence.

* Mark A visited
* A is CWN



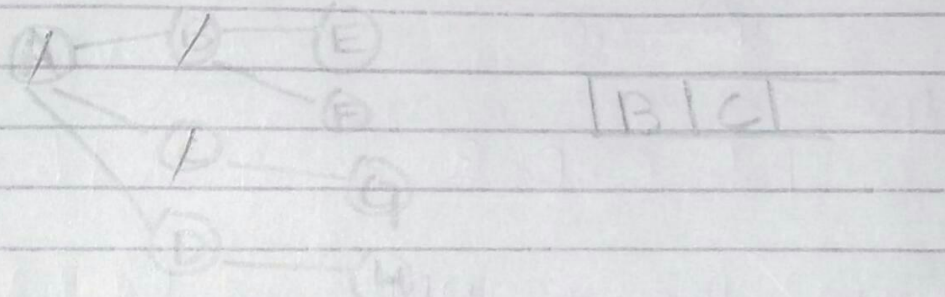
Output Sequence: A

(2) A is adjacent to B, C and D.
* Select B and push it into Queue.
* Add B to the output sequence
* Mark B visited



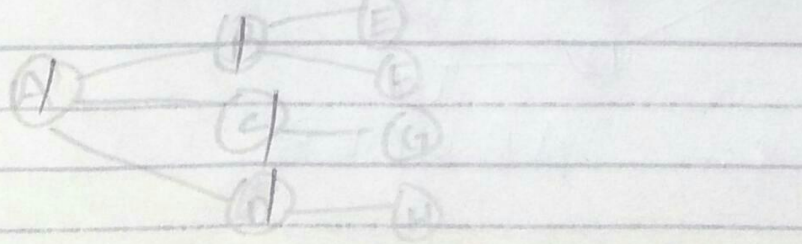

output Sequence.
A, B.

(3) From CWN i.e "A" the adjacent node is "C"
* Now push C to the Queue
* Mark "C" as visited
* Add "C" to output sequence




out put sequence:- A B C

(4) Now "D" is also adjacent to "A".
D is pushed into the Queue.
Mark D as visited
Add D to output sequence.



(6)

Output Sequence:
A B C D

* Now CNN is updated
* 'B' is selected as new CNN
* 'B' is popped from Queue

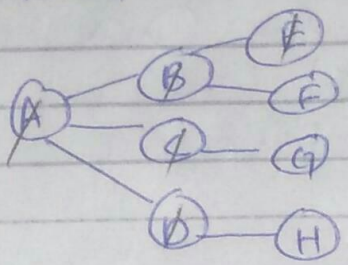
| C | D |

(5)

* B is adjacent to E and F
* E is selected and pushed into the Queue.
* E is visited

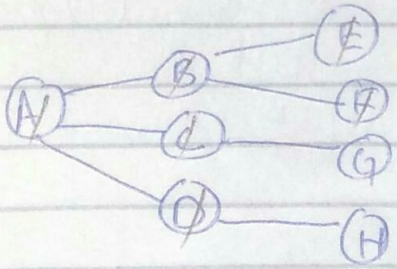
| C | D | E |

* Add E to the output sequence.




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

(6) From the CNN i.e B the adjacent node F is selected
F is pushed into the Queue.
Mark F as visited.
Add F to output sequence




| C | D | E | F |

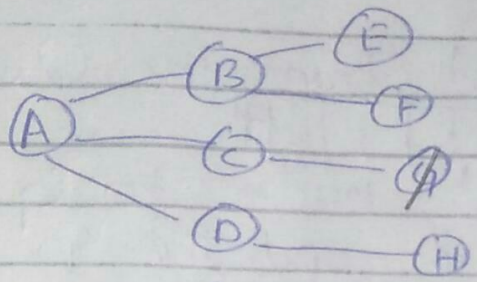
O.S :- A B C D E F

Now CWN is updated to C
C is popped from Queue.

| | | D | E | F |

7 From CWN i.e 'C' the adjacent node is G
Push G into the Queue.
Mark G as visited.
Add G to O.S

| | | D | E | F | G |




O.S :- A B C D E F G

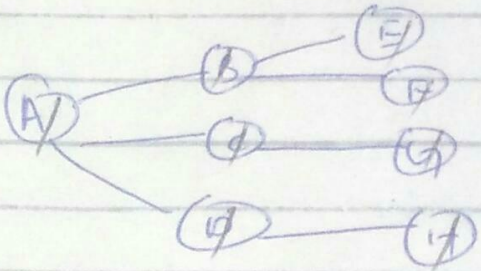
* Now CWN is updated to 'D'
* 'D' is popped from queue

| | | E | F | G |

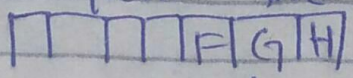
8 Now 'H' is adjacent node to 'D'
Push H to Queue
Add H to O.S

| | | E | F | G | H |

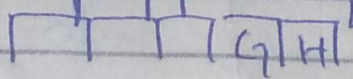
O.S A B C D E F G H



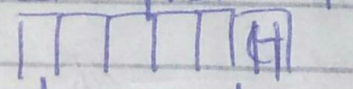
* Now CWN is updated to "E"
* popped 'E' from the Queue



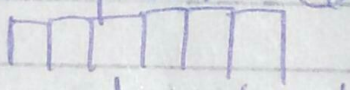
* No adjacent node to 'H'
* No again CWN is updated to F
* Now F is popped from the Queue



* Now adjacent node to 'F'
* Now again CWN is updated to 'G'.
* G is popped from the Queue.

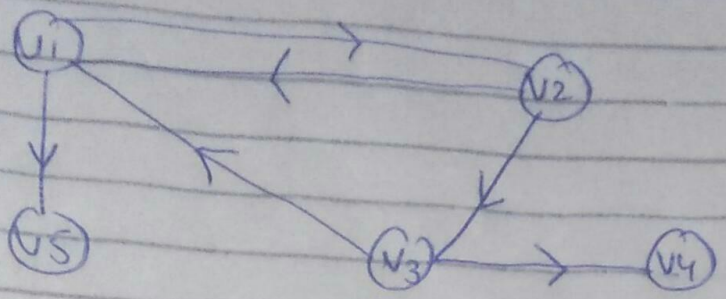


* No adjacent node to G
* Now CWN is updated to 'H'
* H is popped from Queue.



No adjacent node to 'H'
Queue is empty SO Breadth first search is stop.

Q4 Design adjacency matrix




no of nodes = M = 5

order is = M × M

= 5 × 5

Means matrix have 5 rows or 5 columns

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

putting values

$a_{44}$ means $V_4$ to $V_4$ and its zero because there is no loop $V_1$.

$a_{11} = 0$ : no edge from $V_1$ to $V_1$

$a_{12} = 1$ ; " " $V_1$ to $V_2$

$a_{13} = 0$ " "

so the matrix is now

P-1-0

= 5x5

(b)

$$Z = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \text{out degree} \\ 2 \\ 2 \\ 2 \\ 1 \\ 0 \end{array}$$

In degree 21112

(7)

# Q5 Directed Graph:-

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

Order of A = m x m

= 5 x 5

no of nodes = 5

Lets nodes be $v_1, v_2, v_3, v_4, v_5$