I begin with the name of Allah, Who is Most kind, Most Merciful.

Name: Mohammad Bilal

ID: 14956

Course: Design and Analysis of Algorithms

***Instructor: Muhammad Adil** Asset: Prof.*

Program: BS(CS)

Question :1

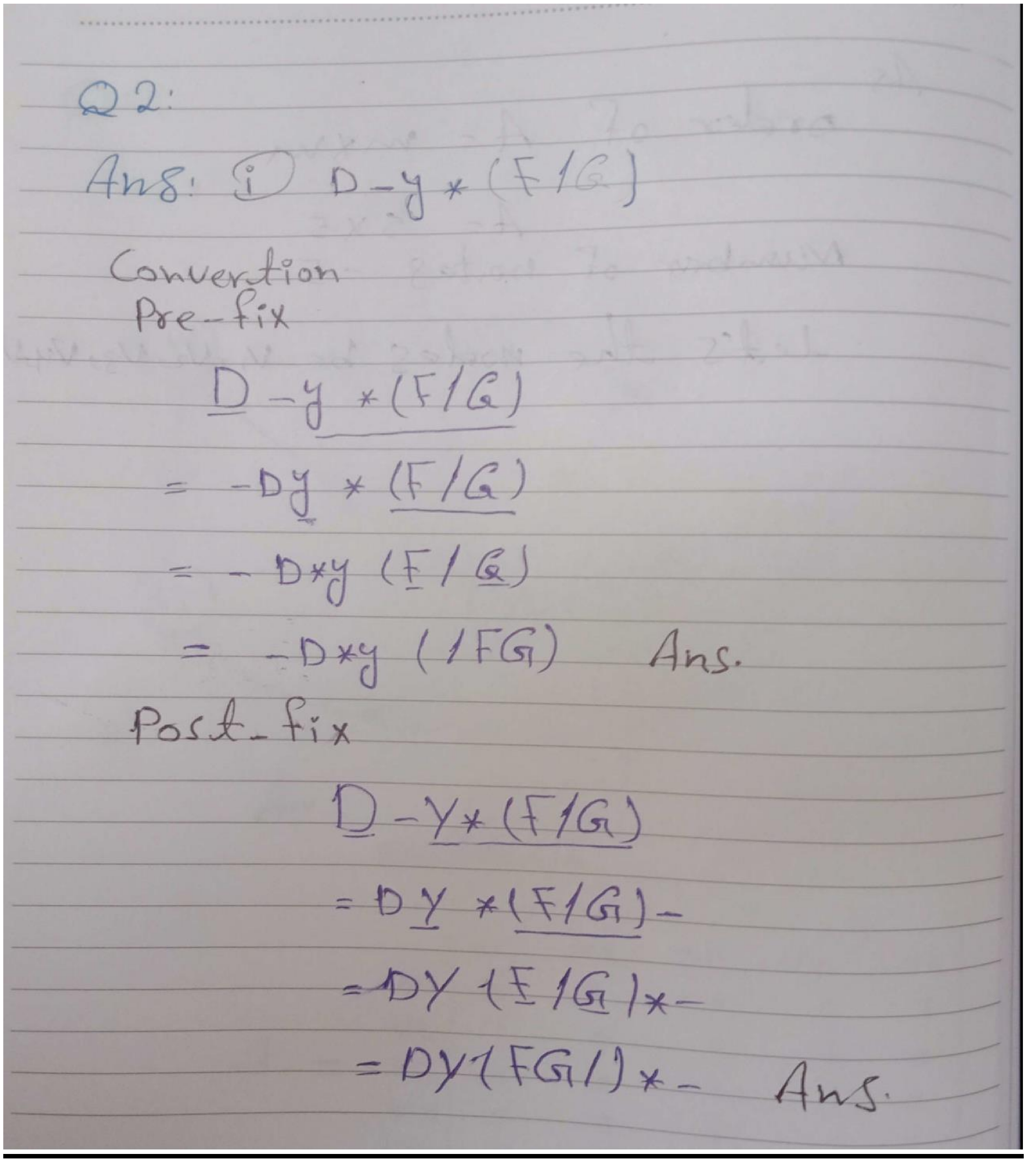
Answer:

1) A **vertex** is a Junction where something takes place in Graph.
2) Nodes that share the same Edge are called **Multiple / parallel Edges**
3) Two Edges that are incident on same Node are called **Adjacent edges**
4) A path between two nodes covering minimum number of nodes
5) is called **Simple path**
6) A Closed Path with more than three edges is called **Cycle**
7) A node with Zero In-Degree is called **Source Node**
8) A node with Zero Out-Degree is called **Sink**
9) **Isolated or Null graph** is a Graph with no pair of vertices having a common edge.
10) **Regular Graph** is a Graph where each node is of the same degree.
11) **Labeled Graph** is a Graph where each Edge is assigned a title.

Question :2

<u>Answer:</u>

I. . D – Y * (F / G)



Q2:

Ans: (i) D-y*(F/G)

Conversion
Pre-fix

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

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

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

$= -D*y\ (/FG)$ Ans.

Post-fix

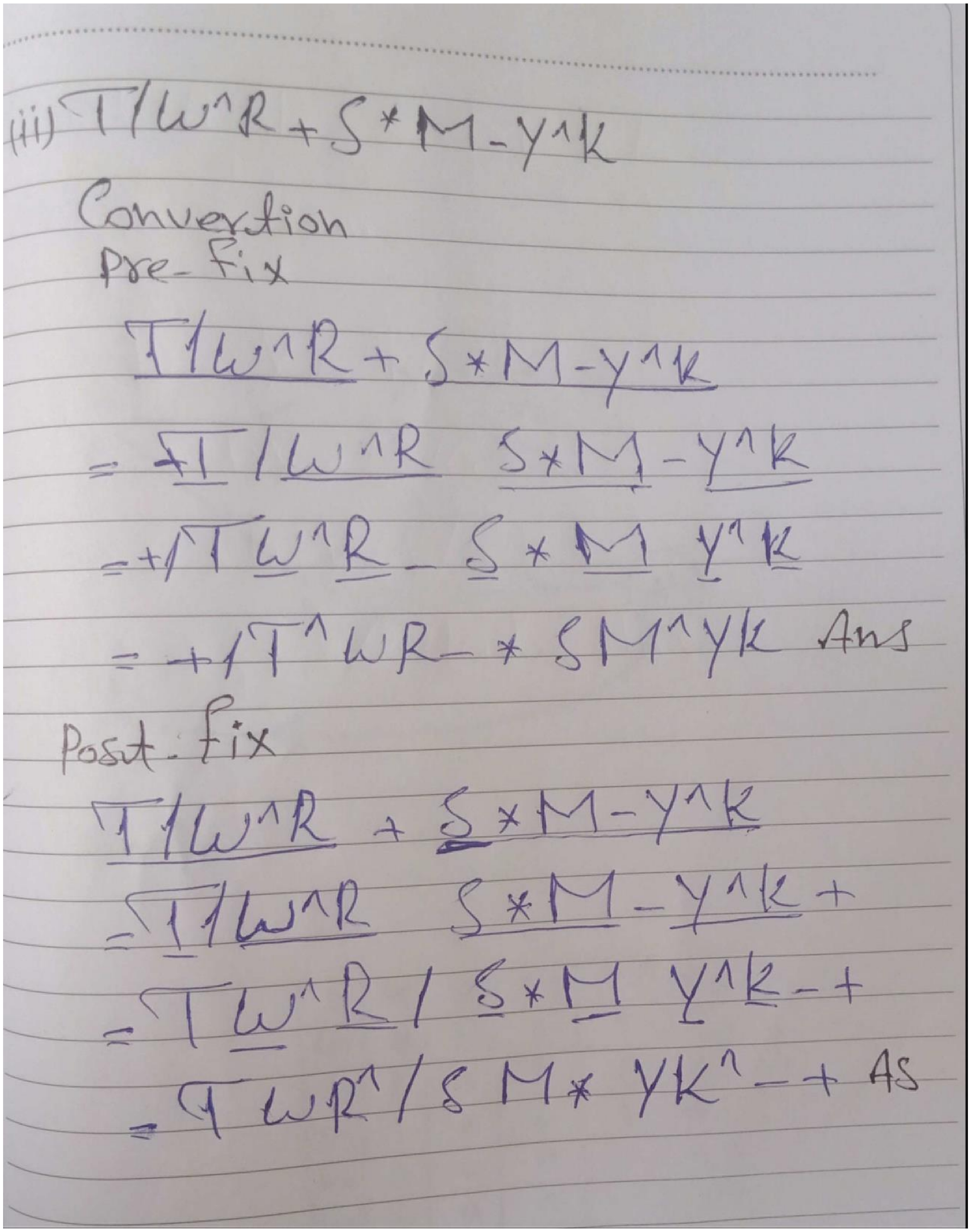
$\underline{D}-\underline{Y*(F/G)}$

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

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

$= DY(FG/)*-$ Ans.

II. T / W ^ R + S * M – Y ^ K



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

Convertion

Pre-fix

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

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

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

= +/T^WR-*SM^YK Ans

Post-fix

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

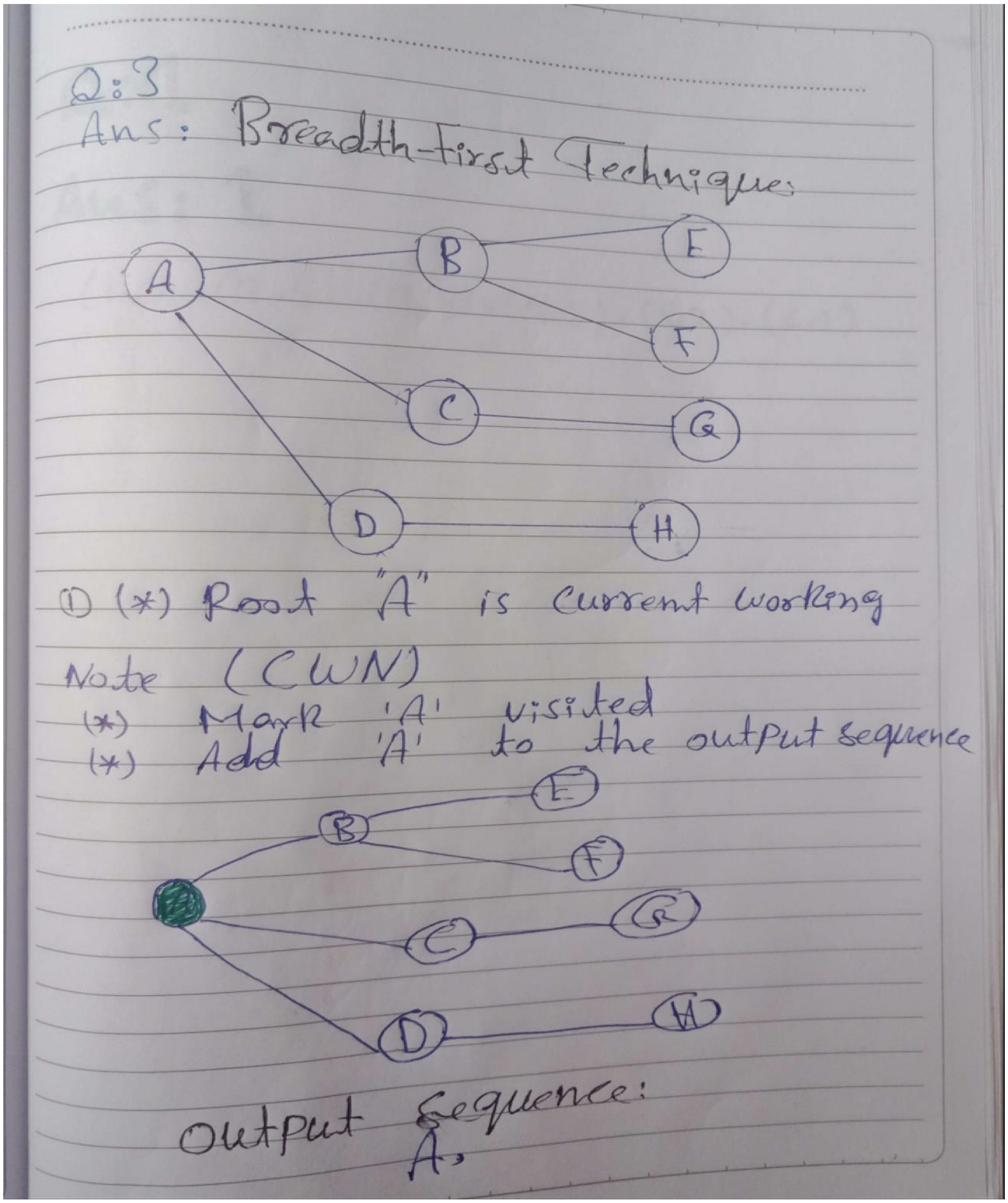
= T/W^R S*M-Y^K+

= TW^R/ S*M Y^K-+

= TWR^/SM*YK^-+ As

Question :3

Answer:



Q:3

Ans: Breadth-First Technique:

A
B
C
D
E
F
G
H

① (*) Root "A" is current working Note (CWN)

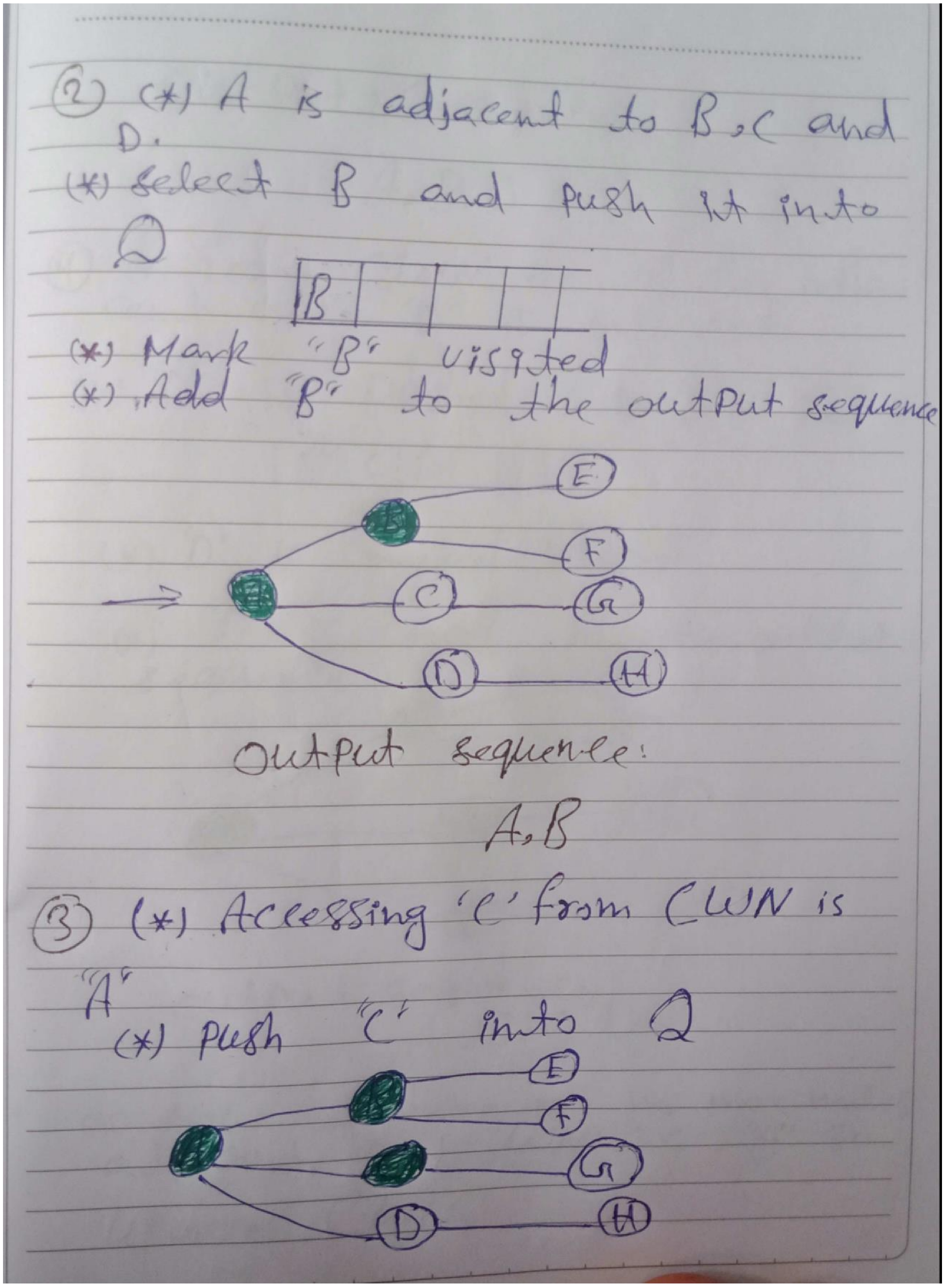
(*) Mark 'A' visited

(*) Add 'A' to the output sequence

B
E
F
C
G
D
H

Output Sequence:

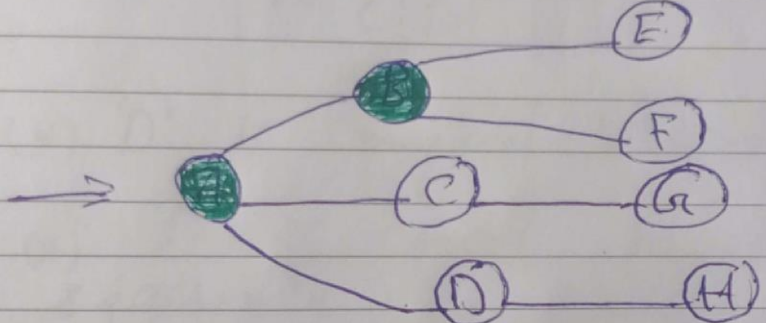
A,



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

B				

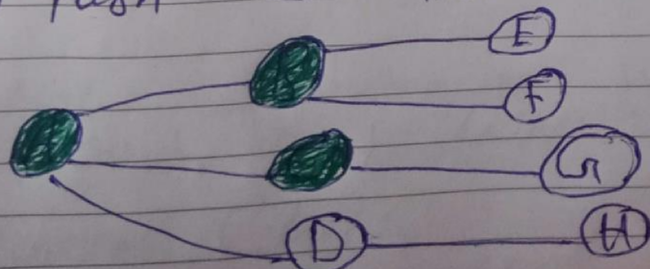
(*) Mark "B" visited
(*) Add "B" to the output sequence



Output sequence:

A, B

③ (*) Accessing 'C' from CWN is "A"
(*) push "C" into Q



Output Sequence

A, B, C

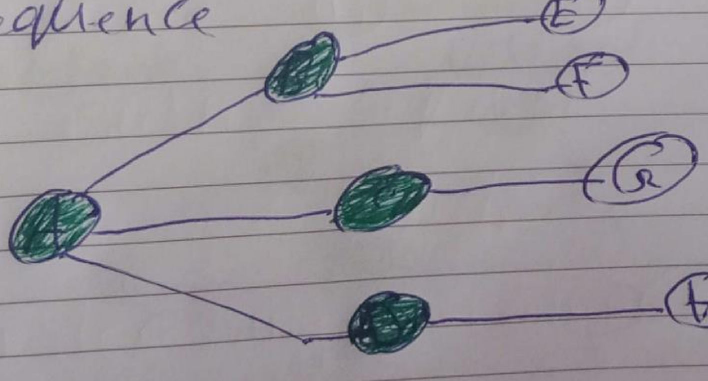
(4) (*) from CWN i.e "A" the adjacent node "D" is selected.

(*) "D" is pushed into the Q

B	C	D		

(*) "D" is marked visited

(*) "D" is add to the output sequence



output sequence
A, B, C, D

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

(*) Select 'B' as CWN.

(*) Pop it from Q

	C	D	

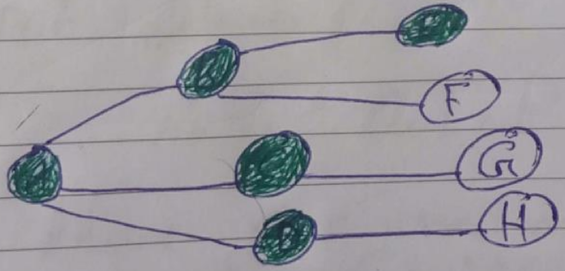
5 (*) B is adjacent to E and F

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

	C	D	E	

(*) Add 'E' to the output sequence

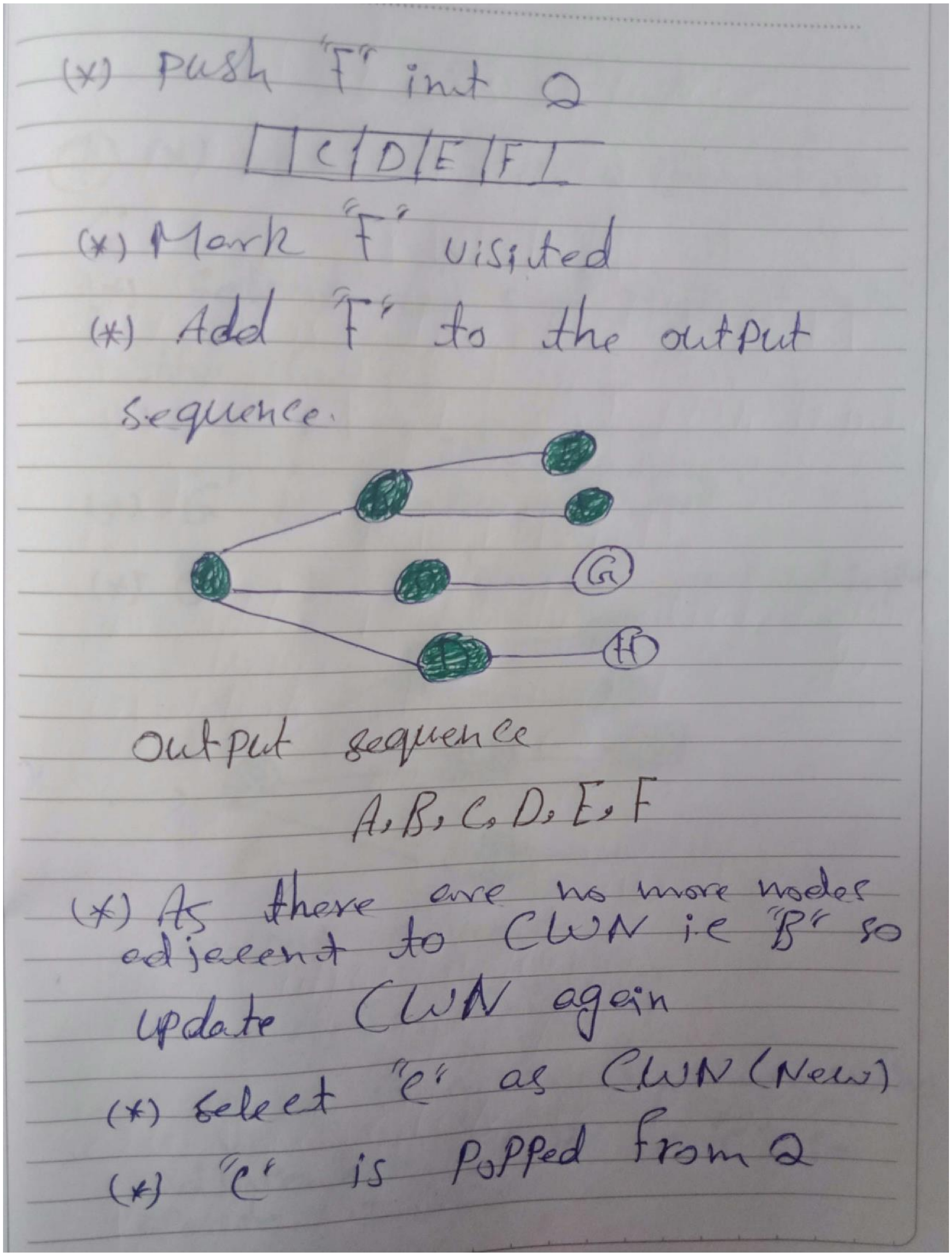
(*) Mark 'E' visited.



Output sequence

A, B, C, D, E

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

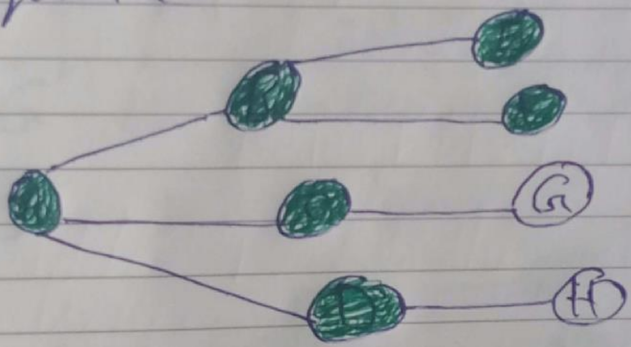


(*) push "F" int Q

	C	D	E	F

(*) Mark "F" visited

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



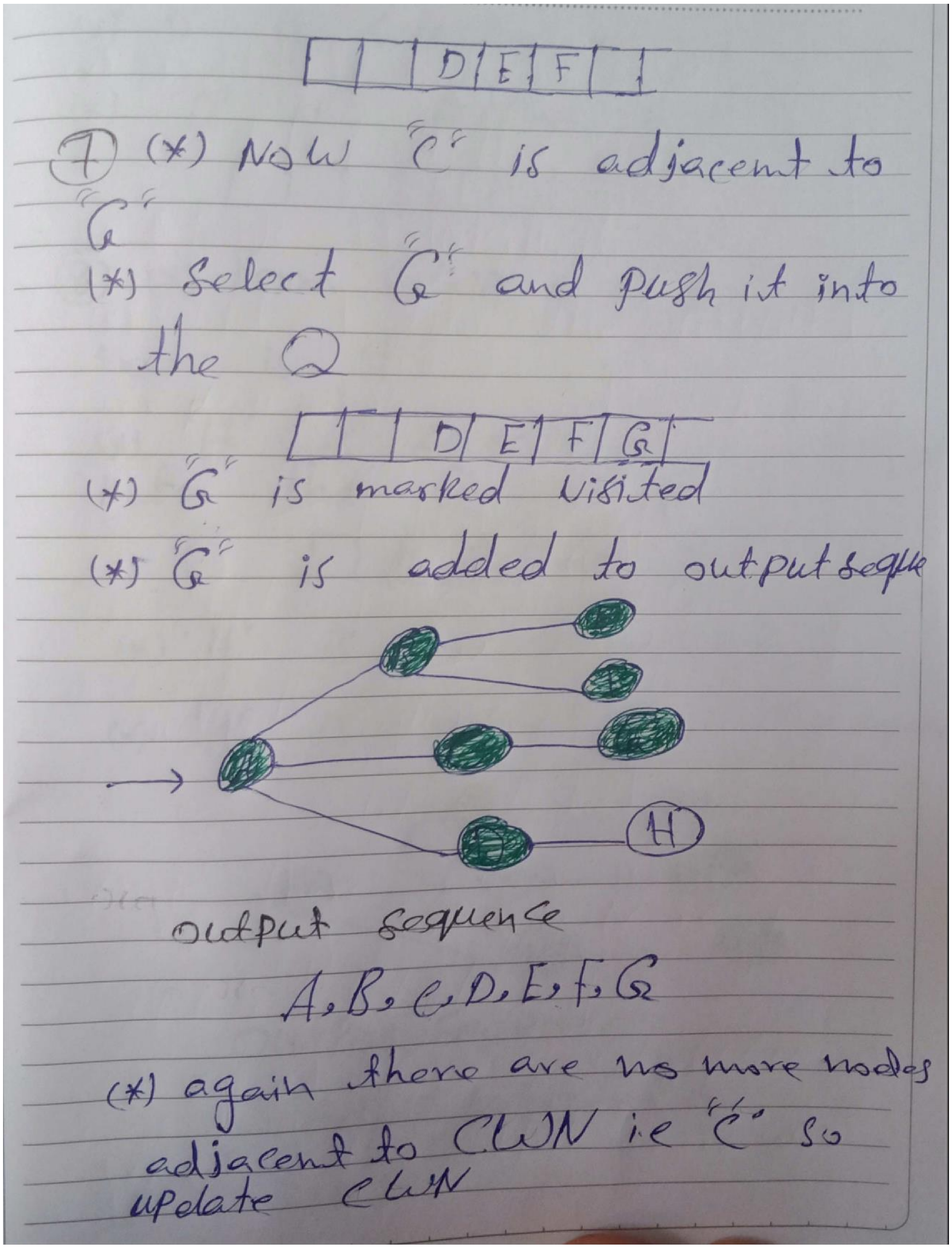
output sequence

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	

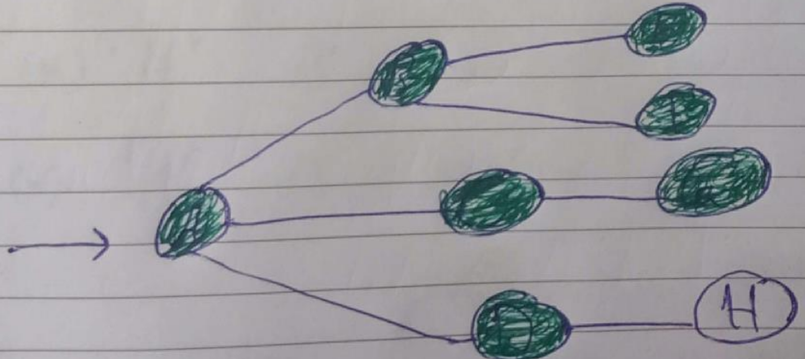
(7) (*) Now "C" is adjacent to "G"

(*) Select "G" and push it into the Q

		D	E	F	G

(*) "G" is marked visited

(*) "G" is added to output seque



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

			E	F	G		

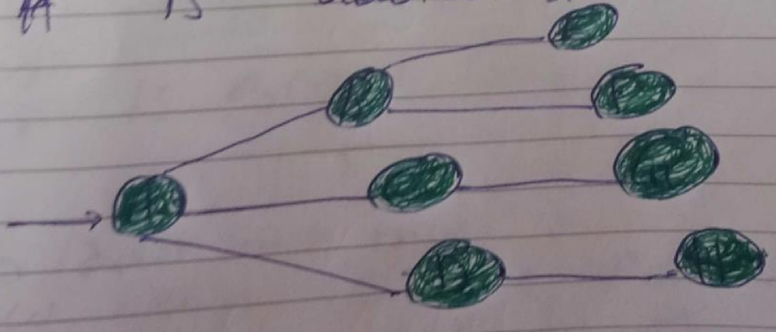
(8) (*) From CWN i.e 'D' adjacent node is H

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

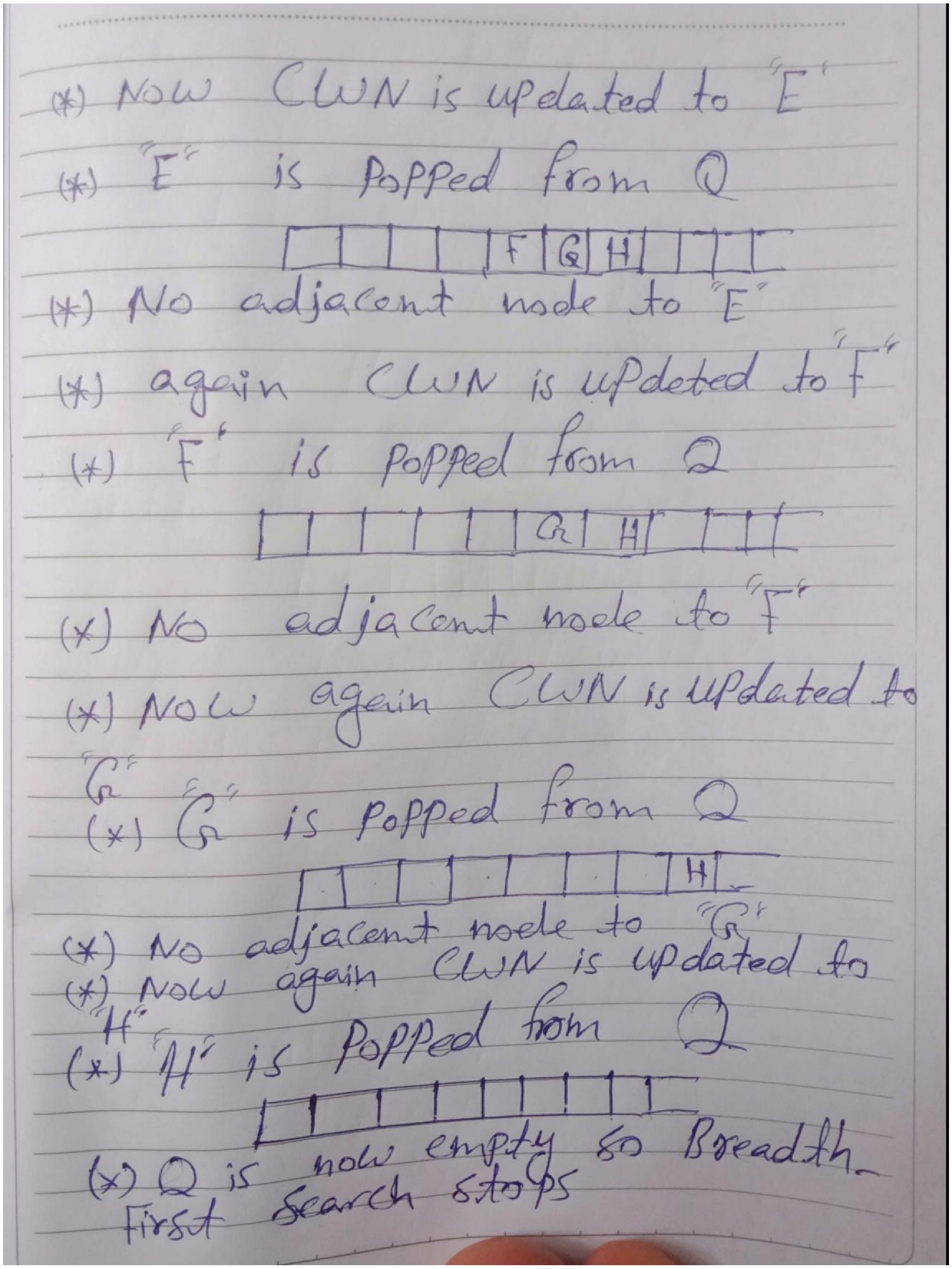
			E	F	G	H		

(*) '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 Q

				F	G	H		

(*) No adjacent node to "E"

(*) again CWN is updated to "F"

(*) "F" is popped from Q

					G	H		

(*) No adjacent node to "F"

(*) Now again CWN is updated to "G"

(*) "G" is Popped from Q

						H	

(*) No adjacent node to "G"

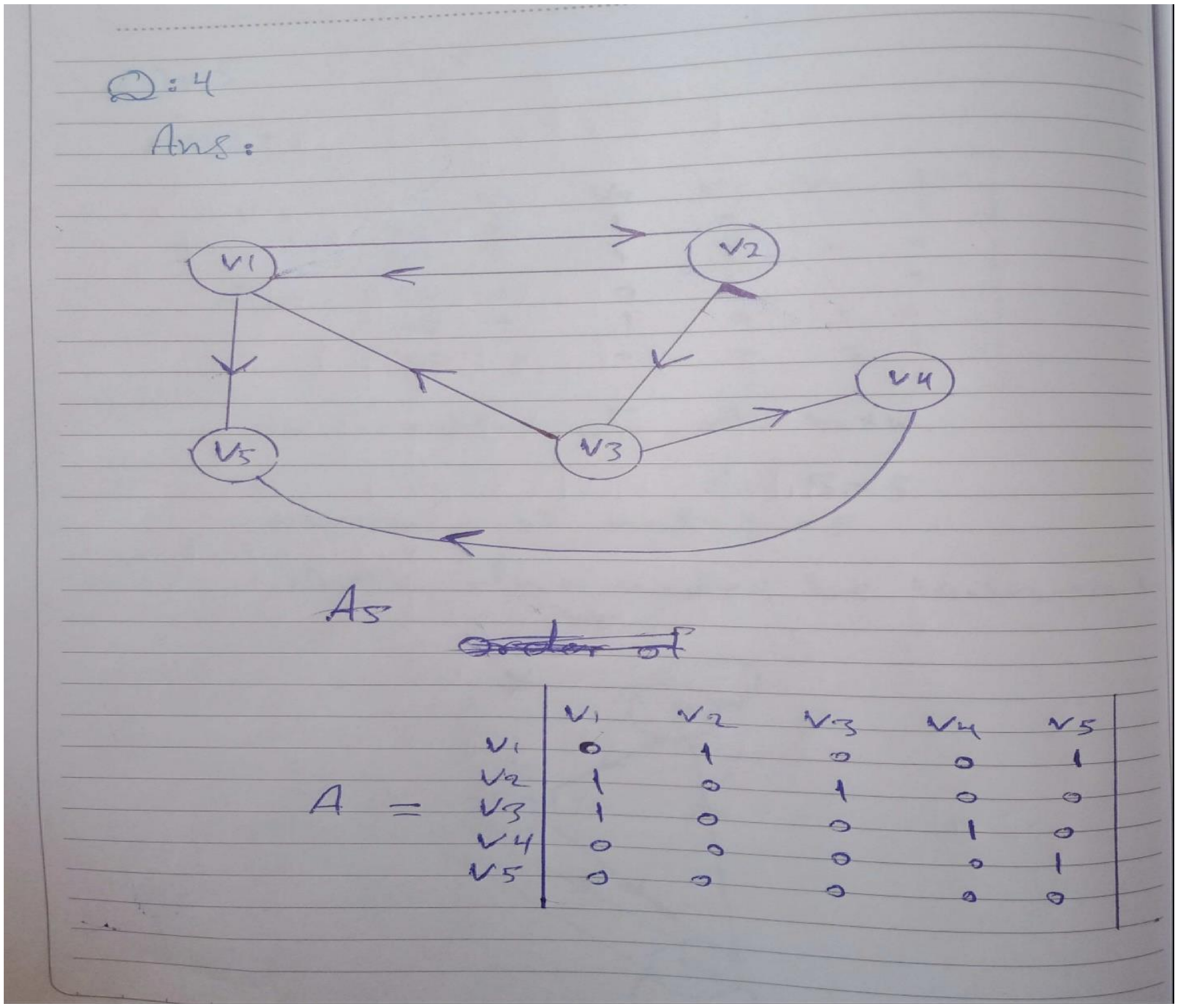
(*) Now again CWN is updated to "H"

(*) "H" is Popped from Q

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

Qeustion :4

Answer:



Q : 4

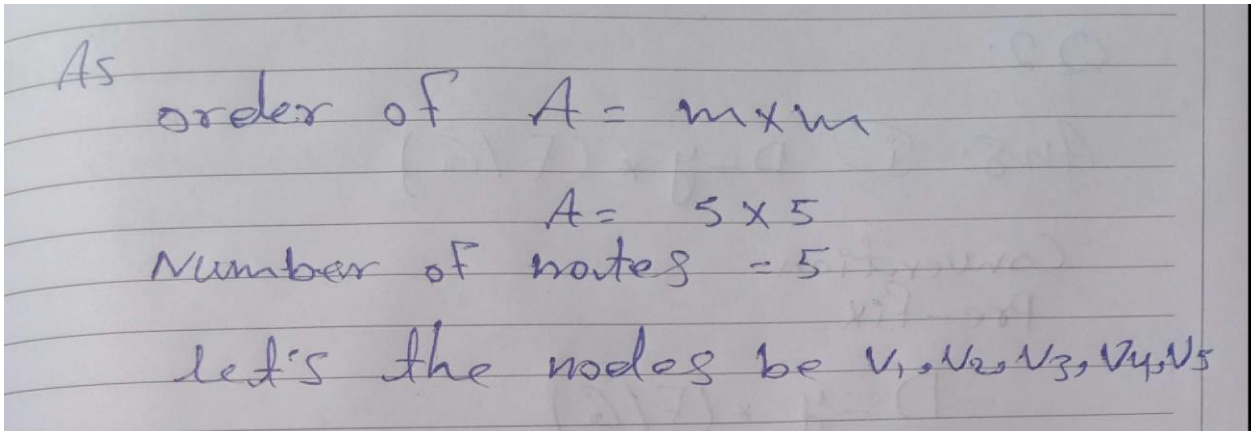
Ans:

$$\begin{tikzcd}
V1 \arrow[rr, bend left] \arrow[dd] & & V2 \arrow[ll, bend left] \arrow[dl] & \\
 & & & V4 \arrow[dlll, bend left] \\
V5 & V3 \arrow[uul] \arrow[urr] & &
\end{tikzcd}$$

As

~~order of~~

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



As order of $A = m \times m$

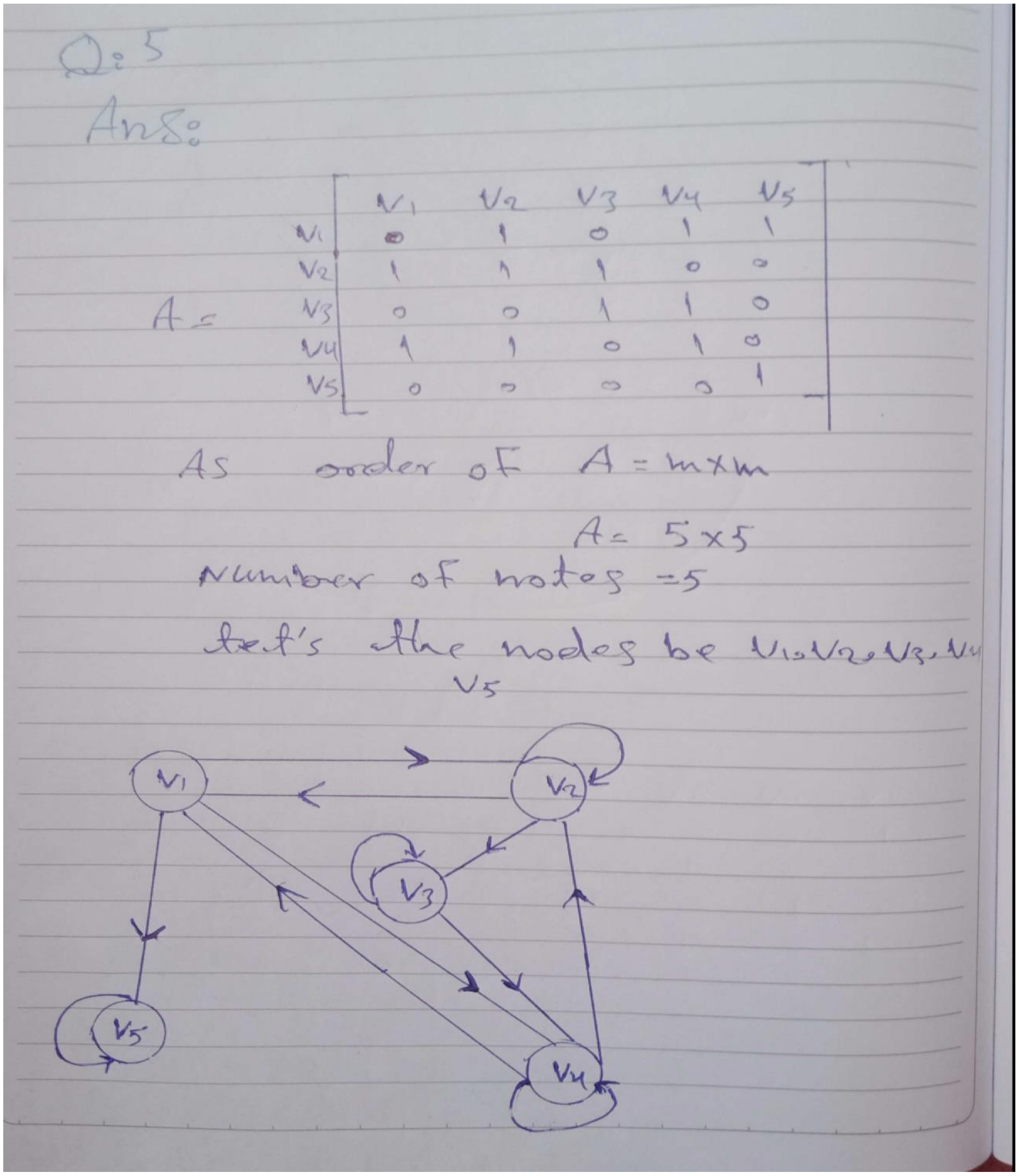
$A = 5 \times 5$

Number of notes $= 5$

let's the nodes be V_1, V_2, V_3, V_4, V_5

Question :5

Answer



Q: 5

Ans:

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

As order of $A = m \times m$

$A = 5 \times 5$

Number of notes $= 5$

let's the nodes be V_1, V_2, V_3, V_4 V_5

$$\begin{tikzcd}
V_1 \arrow[rr, bend left] \arrow[dd] \arrow[ddrr, bend left] & & V_2 \arrow[ll, bend left] \arrow[loop right] \arrow[dl] \\
& V_3 \arrow[loop left] \arrow[dr] & \\
V_5 \arrow[loop left] & & V_4 \arrow[uull] \arrow[uu] \arrow[loop below]
\end{tikzcd}$$

End