

$$\# A - B + (M - N) \times (O + P) = Q \vee R \wedge S \times T + Z$$

<u>INPUT</u>	<u>slack</u>	<u>Postfix Exp.</u>
A		A
-	-	A
B	-	AB
+	+	AB-
(	+ (	AB-
M	+ ( M	AB-M
^	+ ( ^	AB-M
N	+ ( ^ N	AB-MN
)	+ ( ^ N )	AB-MN
X	+ X	AB-MN A
(	+ X (	AB-MN A
O	+ X ( O	AB-MN A
+	+ X ( +	AB-MN A O
P	+ X ( + P	AB-MN A O P
)	+ X	AB-MN A O P +
-	-	AB-MN A O P + X +
R	-\	AB-MN A O P + X + Q
S	-\	"
X	-\ \	AB-MN A O P + X + Q R
T	-\ \ \	"
+	- X	AB-MN A O P + X + Q R S
Z	- X	AB-MN A O P + X + Q R S T
	+	AB-MN A O P + X + Q R S T X
	+	AB-MN A O P + X + Q R S T X - Z

Prefix to Postfix [Same Precedence + Associativity Left  
Stack is Push]

R + L - M \* N + ( O P ) \* W / U / V \* T + Q

Q + T \* V / U / W \* P \ O ( + N \* M - L + K

INPUT      Stack      OUTPUT Prefix exp.

Q	*	Q
+	+	Q
T	+	QT
*	+	QT
V	* *	QTV
/	+ *	QTV
U	+ *	QTVU
/	+ */	QTVU
W	+ */	QTVUW
*	+ */ *	QTVUW
)	+ */ *	QTVUW
P	+ */ *	QTVUWP
\	+ */ *	QTVUWP
O	+ */ *	QTVUWPO
(	+ */ *	QTVUWPO
+	++	QTVUWPO1+//*
N	++	QTVUWPO1+//N
*	++ *	"
M	++ *	QTVUWPO1+//NM
-	++ -	QTVUWPO1+//NM-
L	++ -	QTVUWPO1+//NM-L

+ ++-+ STVUWPO1\*//\* NM\*L

K ++-+ STVUWPO1\*//\* NM\*L K+-++

↓ reverse

++-+KL\* MN\*// \* AOPWUVTQ

8

5

6

4

(~~++-+KL~~ ~~MN\*~~ N + (O1P) \* W / U / V \* T + Q

++-+KL \* MN\*// \* AOPWUVTQ

++-+KL\* MN\*// \* AOPWUVTQ

AOPWUVTQ

AOPWUVTQ

AOPWUVTQ

AOPWUVTQ

AOPWUVTQ

AOPWUVTQ

AOPWUVTQ

## Evaluation of prefix Exp( $R \rightarrow L$ )<sup>Scans</sup>

$$d + b \cdot \Phi(d) = d / e + f$$

$- + a * b.c \rightarrow /d \wedge e.f$

- + a \* b c / d - e f

~~- + 2 4 3 9 / 1 6 1 2 3~~

$$+2 * 3 \ 4 / 16 \ 8 \quad [2^3]$$

$$- + 2 * 3 4 \underline{2} \quad [16 / 8]$$

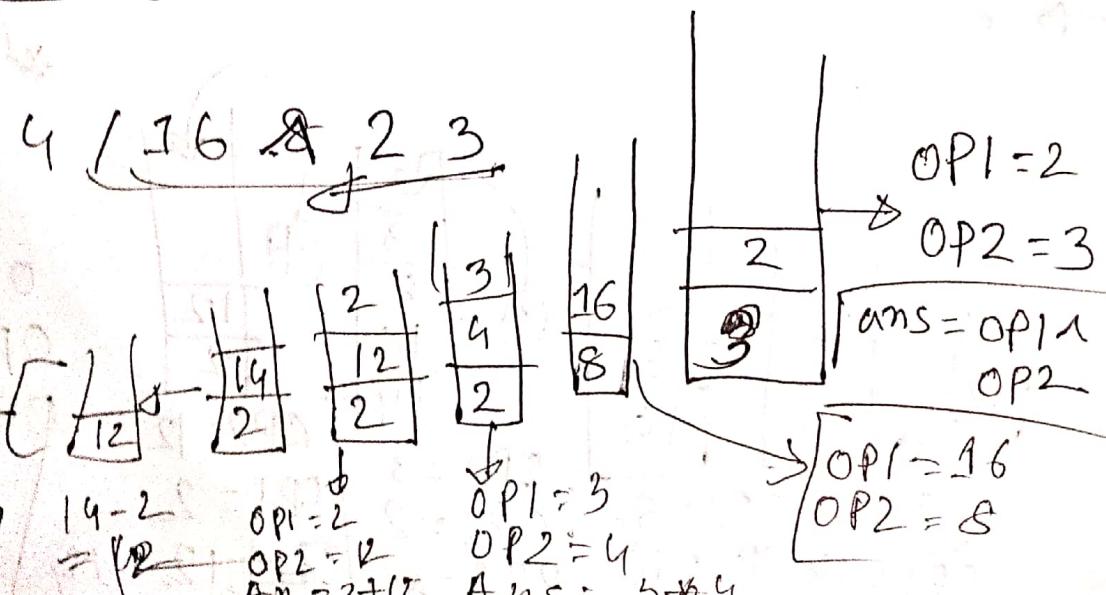
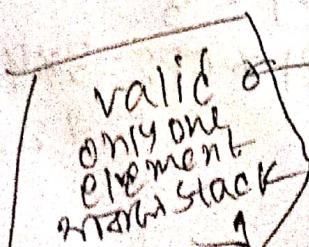
$$\begin{array}{r} \boxed{+ 2} \quad \boxed{12} \quad 2 \\ \hline \end{array} \quad [3 * 4]$$

$$= \boxed{242} \quad [2+12]$$

12 [19-2]

## using stack

$$- + 2 * 3 \ 4 \ \cancel{176 \ 823}$$



## Left to right evolution of postfix expression

# ~~a+b\*c-d/e+f~~

$a b c * + d e f / -$  [postfix]

$$\xrightarrow{2} \underline{\underline{3 \ 4 * + 16}} \ 2 \ 3 \ \underline{\underline{\wedge \ / -}} \quad \begin{cases} a=2, b=3, c=4 \\ d=16, e=2, f=3 \end{cases}$$

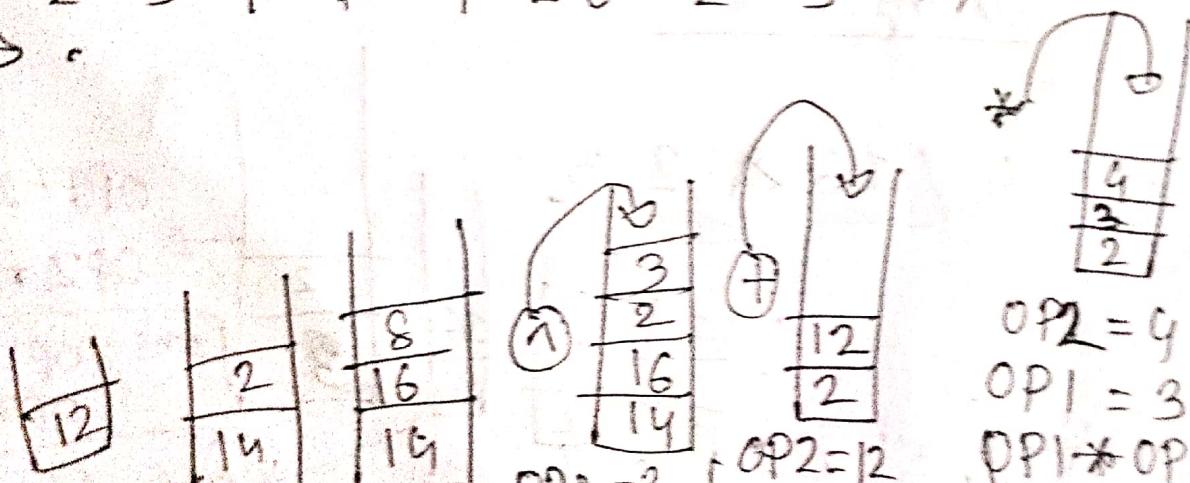
$$\underline{\underline{2 \ 12 + 16}} \ 2 \ 3 \ \underline{\underline{\wedge \ / -}} \quad 3 * 4 = 12$$

$$14 \ 16 \ 2 \ 3 \ \underline{\underline{\wedge \ / -}} \quad [2 + 12 = 14]$$

$$\begin{array}{r} 14 \ 16 \ 8 \ 1 \ - \\ \underline{14 \ 2} \ - \\ 12 \end{array} \quad [16 / 8] = 2$$

## Using Stack

$\xrightarrow{2 \ 3 \ 4 \ * \ + \ 16 \ 2 \ 3 \ \wedge \ / \ -}$



$$\begin{array}{l} OP2 = 2 \\ OP1 = 14 \end{array}$$

$$\begin{array}{l} OP2 = 3 \\ OP1 = 2 \end{array}$$

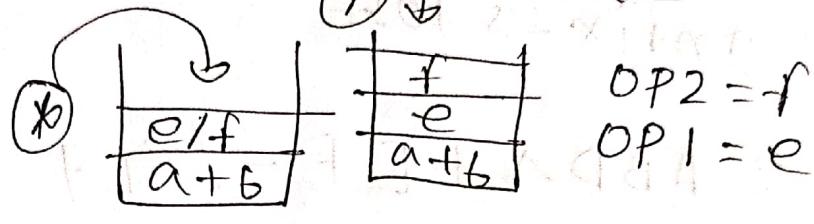
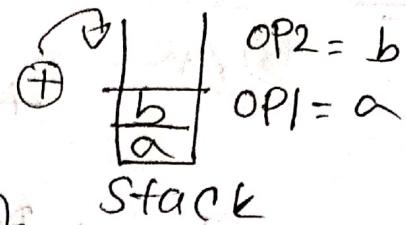
$$\begin{array}{l} OP2 = 12 \\ OP1 = 2 \\ 2 - 8 = -6 \end{array}$$

$$\begin{array}{l} OP2 = 4 \\ OP1 = 3 \\ OP1 * OP2 = 3 * 4 \\ = 12 \end{array}$$

## Postfix to Infix [L → R]

$$a b + e f / *$$

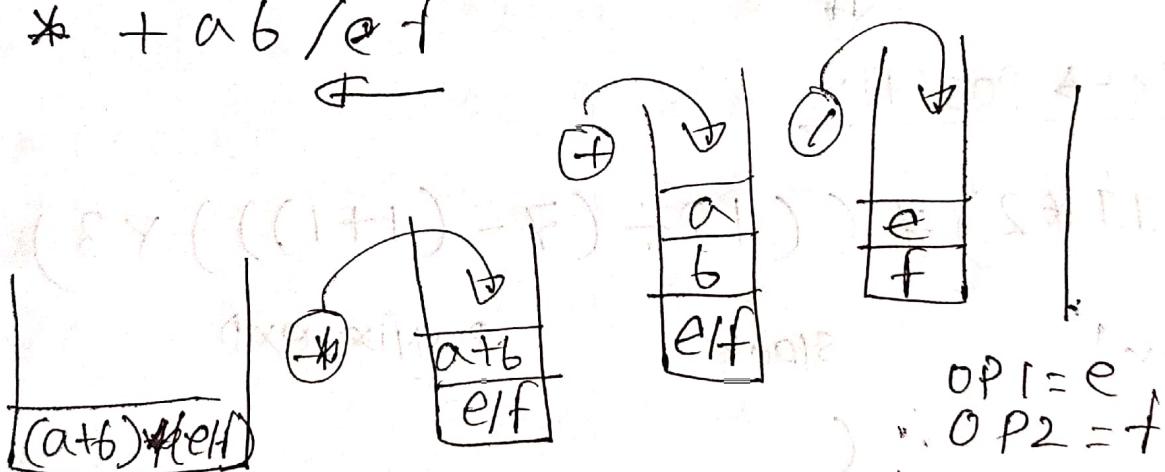
$$\rightarrow (a+b) * (e/f)$$



OP2 = e/f  
OP1 = a+b

## Prefix to Infix [R → L]

$$* + a b / e f$$



$$(a+b)* (e/f)$$

Solved      Infix  $\rightarrow$  Prefix  
③ An       $+ / + A \wedge B D - E F G$

Infix  $\rightarrow$  Post

③ ABD \wedge + E F - / G I +

MQD QUS

Infix  $\rightarrow$  Prefix / Infix  $\rightarrow$  Postfix and eval  
 $(32 - 11 * 2) * ((15 \div (7 - (1+1))) \times 3)$

4

Infix  $\rightarrow$  Postfix

$(32 - 11 * 2) * ((15 \div (7 - (1+1))) \times 3)$

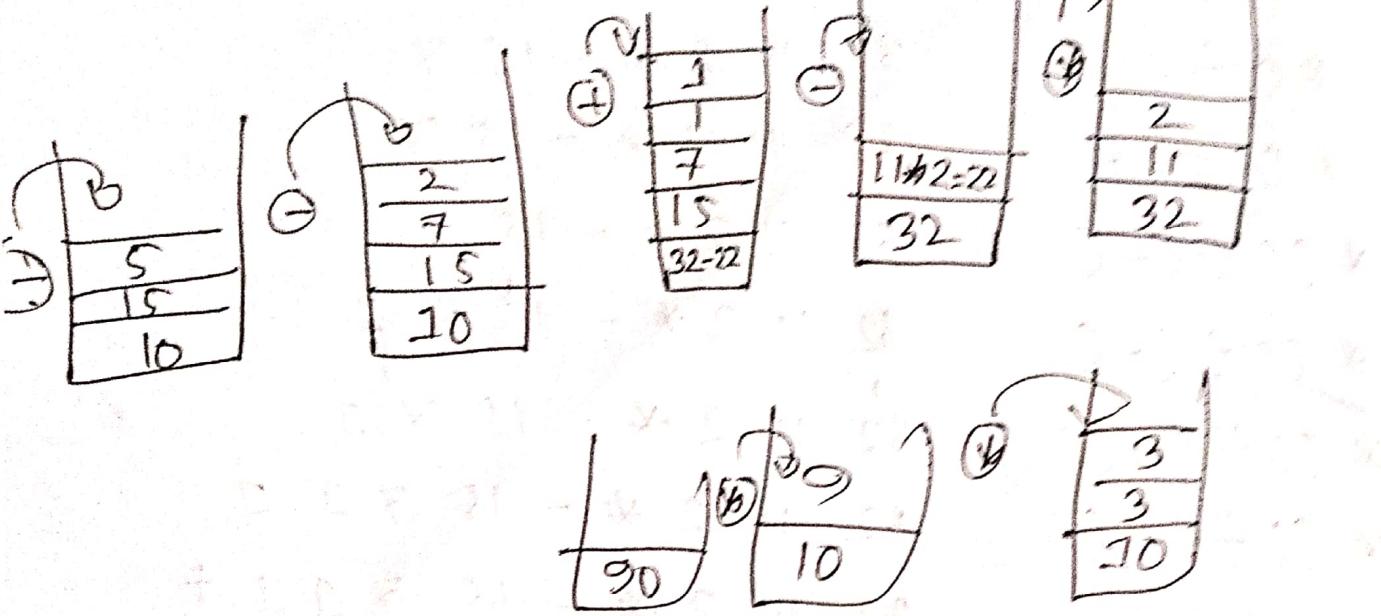
Infix	Stack	Postfix EXP
(	(	
32	(	32
-	(-	32
11	(-	32 11
*	(-*	32 11
2	(-*	32 11 2
)	.	32 11 2 -*
*	*	32 11 2 -*

	*	32 11 2 *
1	*	32 11 2 * - 15
5	*	32 11 2 * - 15
÷	*	32 11 2 * - 15
7	*	32 11 2 * - 15
-	*	32 11 2 * - 15
1	*	32 11 2 * - 15
+	*	32 11 2 * - 15
1	*	32 11 2 * - 15
)	*	32 11 2 * - 15
)	*	32 11 2 * - 15
)	*	32 11 2 * - 15
*	*	32 11 2 * - 15
3	*	32 11 2 * - 15
)	*	32 11 2 * - 15

## Evaluation of Postfix

32 11 2 \* - 15 7 2 2 + - ÷ 3 % 10  
 →

90



Postfix

Stack

32

32

11

32, 11

2

32, 11, 2

\*

32, 22

-

10

15

10, 15

7

10, 15, 7

1

10, 15, 7, 1

1

10, 15, 7, 1, 1

+

10, 15, 7, 2

-

10, 15, 5

Postfix

Stack

÷

10, 3

3

10, 3, 3

\*

10, 9

\*

90.

$$(60 * 3 - 12 / 4) * 2 + (32 - 11 * 2)$$

infix	stack	postfix
(	(	
60	(	60
*	(*	60
3	(*	60, 3
-	(-	60, 3, *
12	(-	60, 3, *, 12
/	(-1	60, 3, *, 12,
4	(-1	60, 3, *, 12, 4
)		60, 3, *, 12, 4, 1, -
*	*	60, 3, *, 12, 4, 1, -
2	*	60, 3, *, 12, 4, 1, -
+	+	60, 3, *, 12, 4, 1, -, 2
(	+ (	60, 3, *, 12, 4, 1, -, 2, *
32	+ (	60, 3, *, 12, 4, 1, -, 2, *, 32
-	+ (-	60, 3, *, 12, 4, 1, -, 2, *, 32, 11
11	+ (-	60, 3, *, 12, 4, 1, -, 2, *, 32, 11
*	+ (- *	60, 3, *, 12, 4, 1, -, 2, *, 32, 11, 2
2	+ (- *	60, 3, *, 12, 4, 1, -, 2, *, 32, 11, 2
)		60, 3, *, 12, 4, 1, -, 2, *, 32, 11, 2 * - +

Q  
push(1)  
push(2)  
pop(2)  
push(3)  
push(4)  
pop(4)  
push(3)  
pop(5)  
pop(3)  
pop(1)

→ { 2, 4, 5, 3, 1 }

push(1)  
pop(1)  
push(2)  
push(3)  
push(4)  
push(5)  
pop(5)  
pop(4)

not possible

{ 1, 5, 4, 2, 3 }

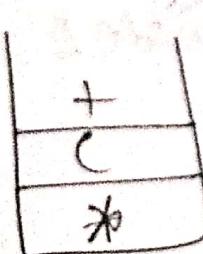
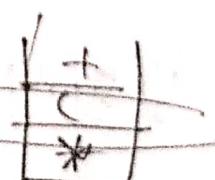
push(1)  
pop(1)  
push(2)  
push(3)  
pop(3)  
push(4)  
push(5)  
pop(5)  
pop(4)  
pop(2)

→ { 1, 3, 5, 4, 2 }

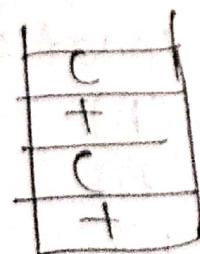


2012 (6)

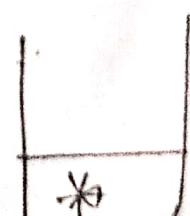
A A number



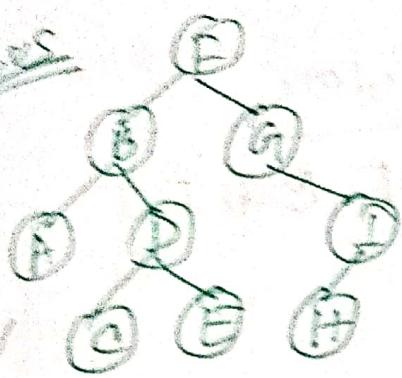
A number



A left parenthesis



A right parenthesis



Preorder

F B A D C E G I H

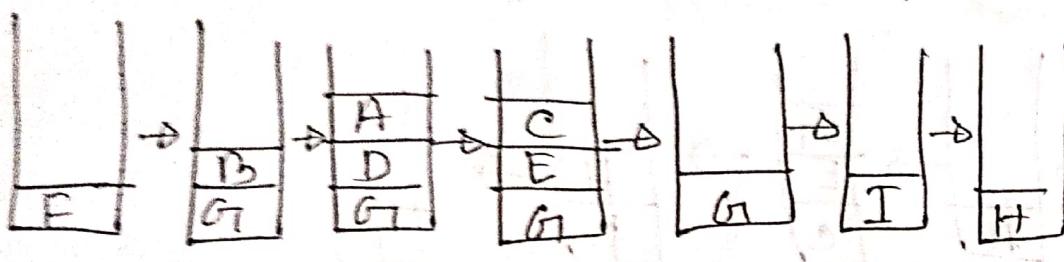
Inorder

A B C D E F G I H

Postorder

A C E D B H I G F

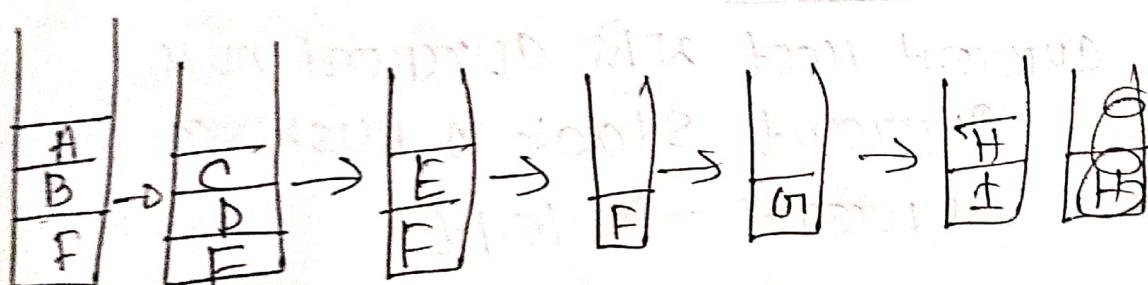
using stack Preorder



Process

F B A D C E G I H

using stack Inorder



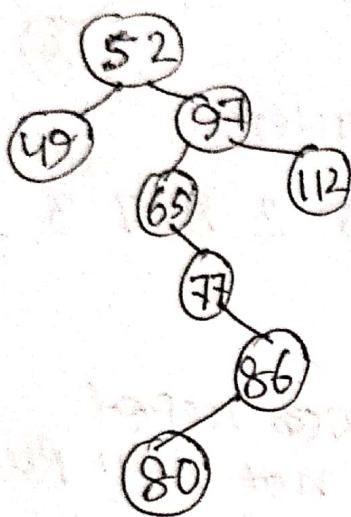
Process

A B C D E F G I H

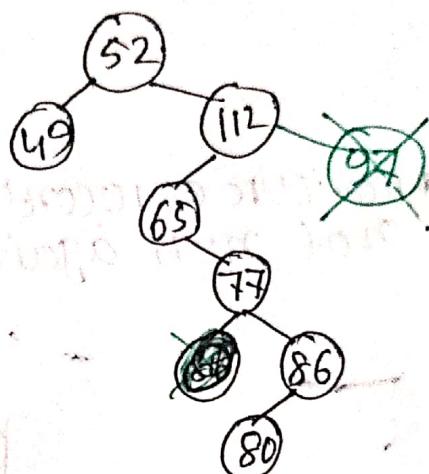
BST 2 (b)

Draw a BST

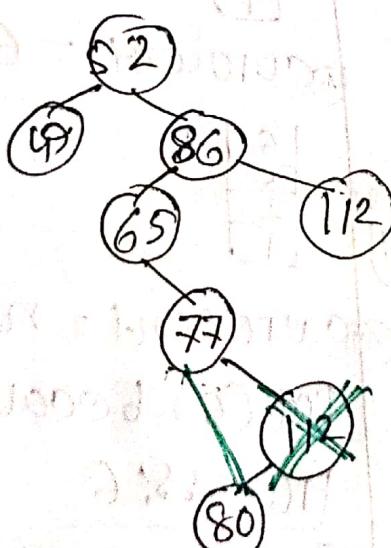
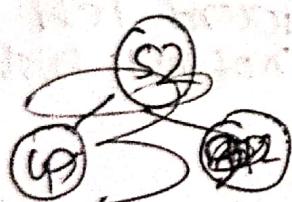
52, 97, 112, 65, 77, 49,  
86, 80



BST



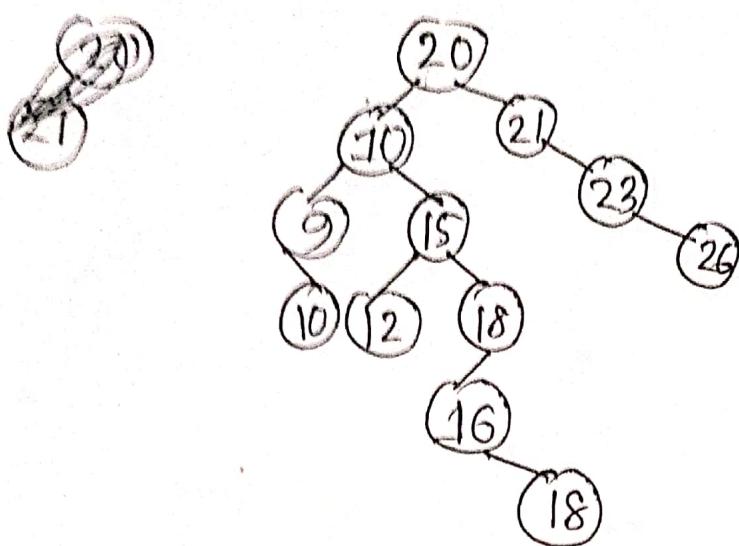
BST after delete(97)  
successor



BST after delete(97)  
predecessor

2012

20, 21, 10, 15, 18, 23, 26, 9, 12, 10, 16-18



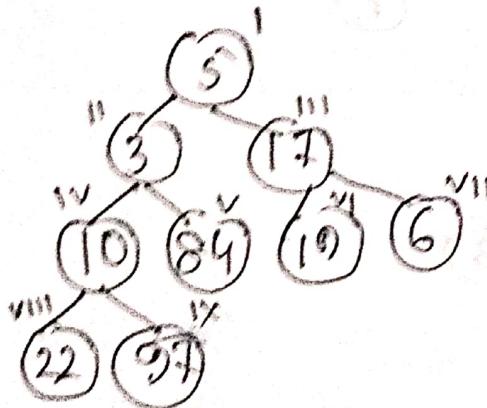
BST

Q If the node is 20  
 then its successor 21 [has no left child]  
 " Predecessor 18 [has no right child]

If the node is 10  
 then its successor 12 [has no left child]  
 " predecessor 10 [" " right child  
 [showed]

$$A = \{8, 3, 17, 10, 84, 19, 6, 22, 97\}$$

8	3	17	10	84	19	6	22	97
1	2	3	4	5	6	7	8	9



left child of  $i$ th node

$$= 2 * i$$

right child of  $i$ th node

$$= (2 * i) + 1$$

parent of  $i$ th node

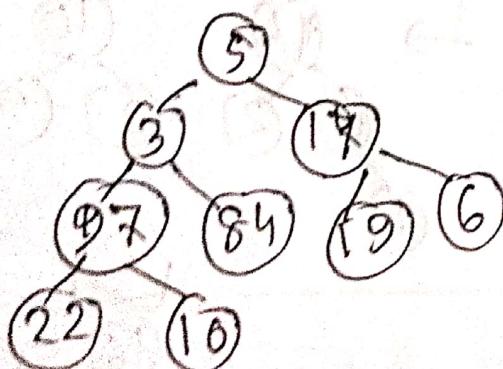
$$= \left[ \frac{i}{2} \right]$$

Here leaf node index

$$\text{leaf node } \frac{n}{2} - 10 \approx n$$

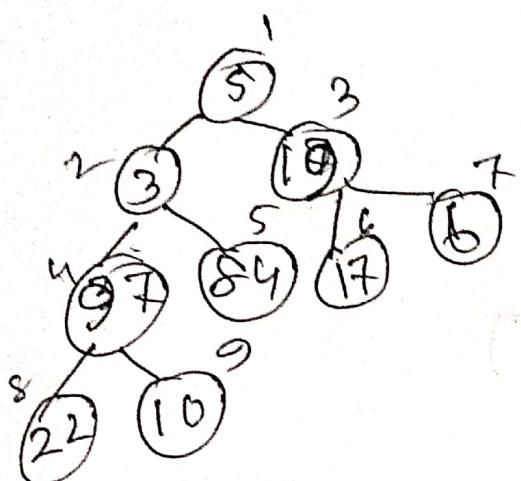
$\therefore$  last non-leaf node index = 4

swap  $A[4]$  and  $A[9]$



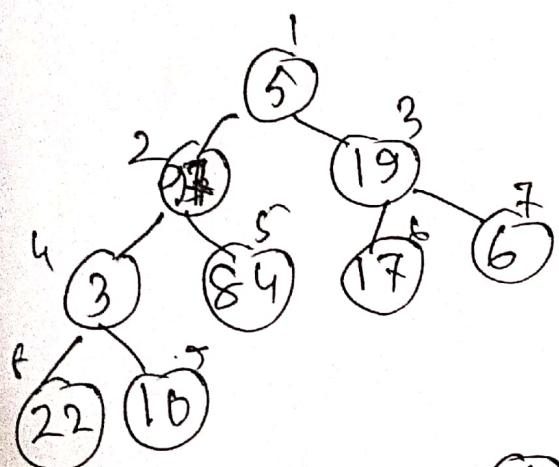
heapify : Index = 3

swap A[6] and A[3]

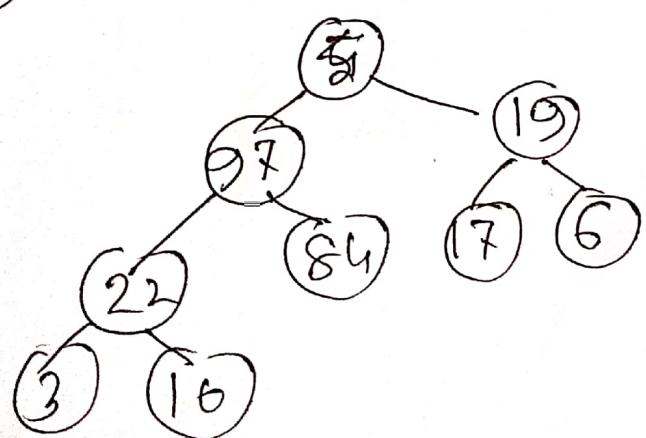


heapify : Index = 2

At first swap A[2] and A[4]

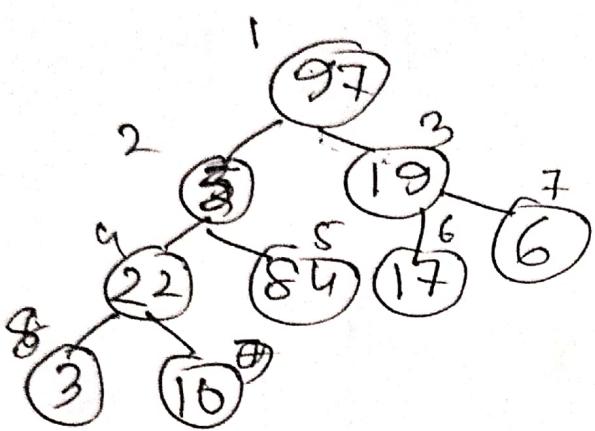


again swap  
A[4] and A[8]

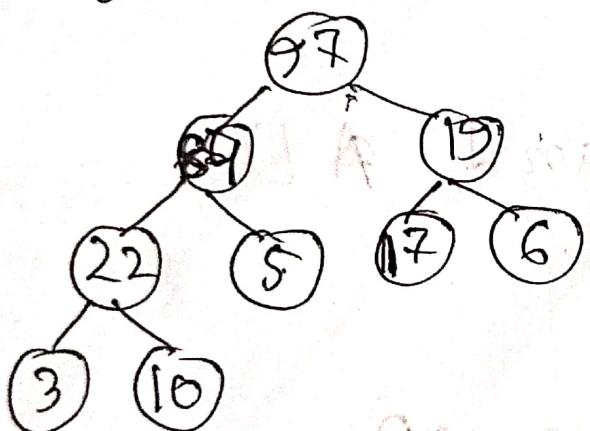


Heapify index 1

swap A[1] and A[3]

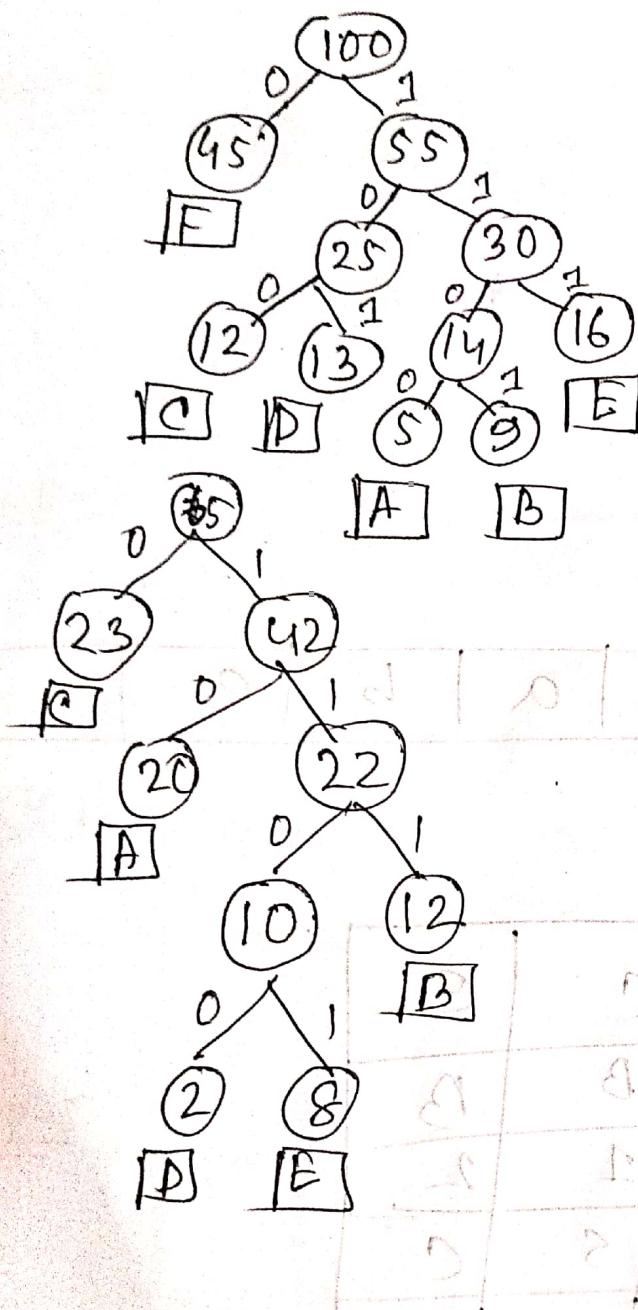


again swap A[2] and A[5]



max heap

Huffman code TC  $\leftarrow \frac{n}{2} \log_2 n$  Extract min number of unique character



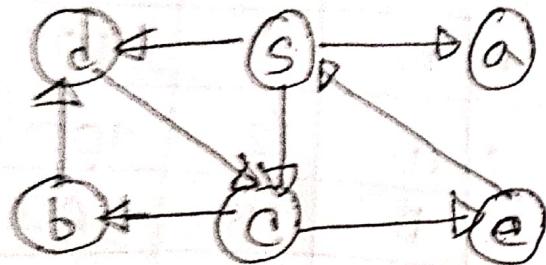
	Sym	freq	Huffman code
1	A	5	1100
e	B	9	1101
c	C	12	100
b	D	13	101
2	E	16	111
-	F	45	0

	Sym	freq	Huffman code
	A	20	10
	B	12	111
	C	23	0
	D	2	1100
	E	8	1101

BFS

(queue Data structure  
for finding the shortest  
path)

question - 2020



BFS

queue

s	.	c	d	a	b	e	None
---	---	---	---	---	---	---	------

vertex	d	s	a	b	c	e	
pre	B	B	B	B	B	B	
dis	1	0	1	2	1	2	
pre	S	null	S	C	S	C	

Step 1

aveve

[s | ]

visited

[ | | | | ]

Step 2

aveve

[c | d | a | ]

visited

[s | ]

Step 3

aveve

[d | a | b | e | ]

visited

[s | c | ]

Step 4

aveve

[a | b | e | ]

visited

[s | c | d | ]

Step 5

aveve

[b | e | ]

visited

[s | c | d | a | ]

Step 6

aveve

[e | ]

visited

[s | c | d | a | b | ]

Step 7

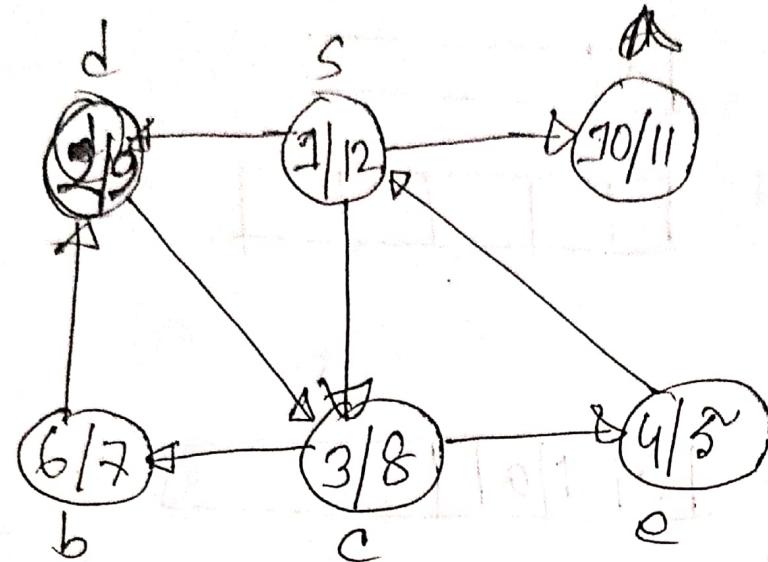
aveve

[ | | | ]

visited

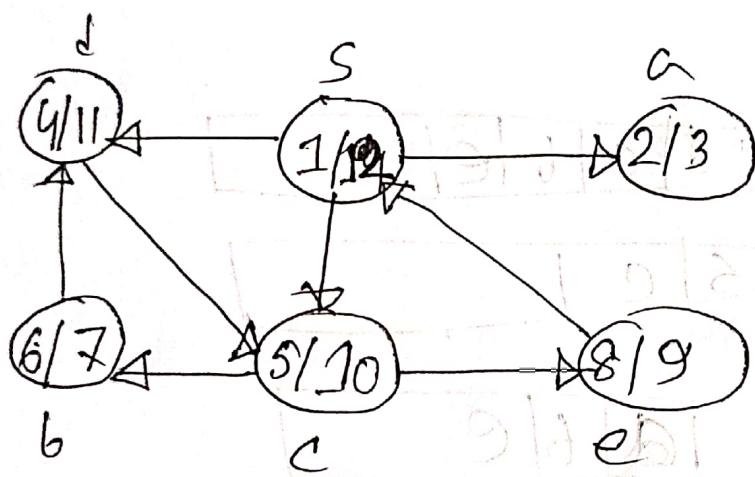
[s | c | d | a | b | ]

DES



no-1

FCFS



no-2

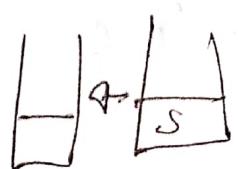
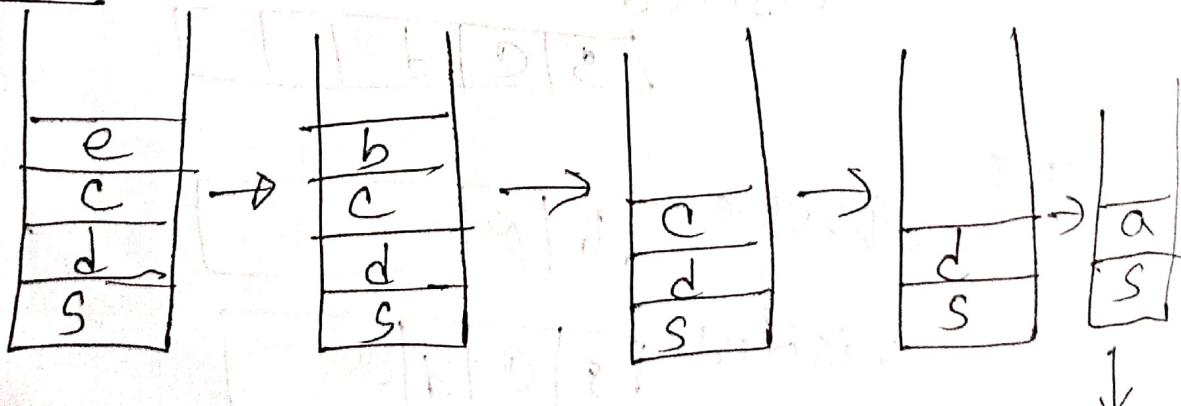
for no 1

stack

POP

e b c d a s

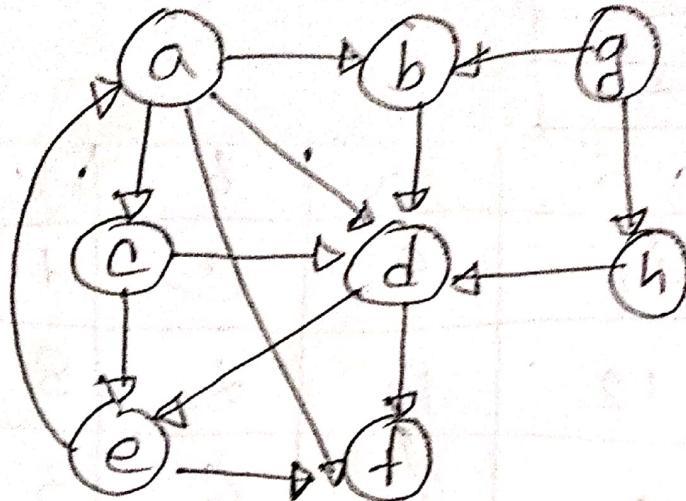
finishing time  
algorithm



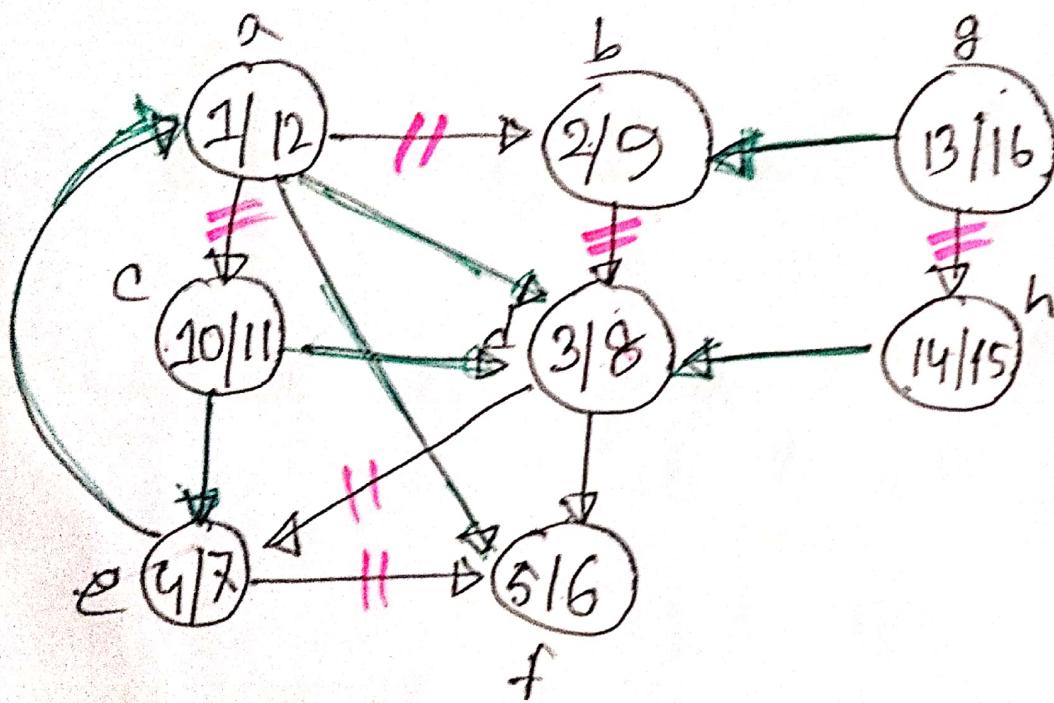
node	d	$1 \rightarrow s$	a	b	c	e
price	s	nil	s	c	d	c
finishing	9	12	11	7	8	5
discoverd time	2	1	10	6	3	4



Ques



$$n=8  
y=b$$



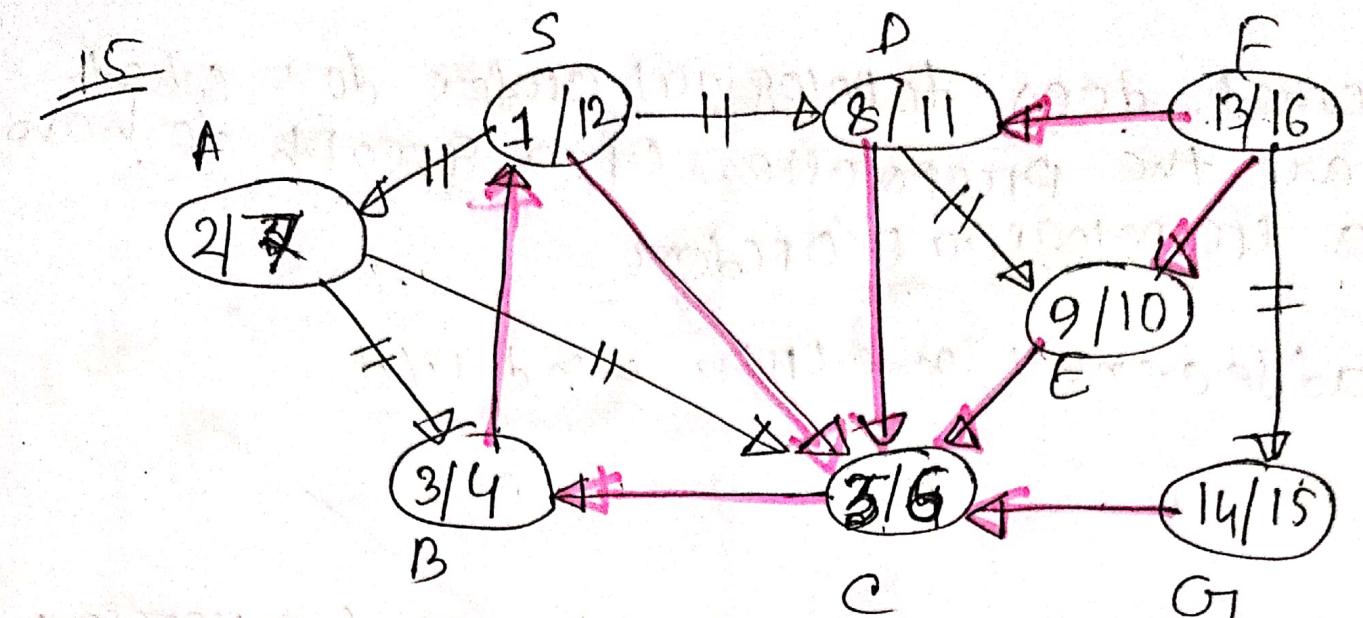
$$h=x  
d=j$$

tree edge : ab, bd, de, ef, ac, gh

back edge : ea,

forward edge : ad, af

cross edge : bg, hd, cd, ce



Tree edge: SA, AB, AC, SD, ~~SD~~, DE, FG

Backedge: ~~EC~~, EC, PC, CB, BS

forward edge: SC

CROSS edge: FD, FE, GC

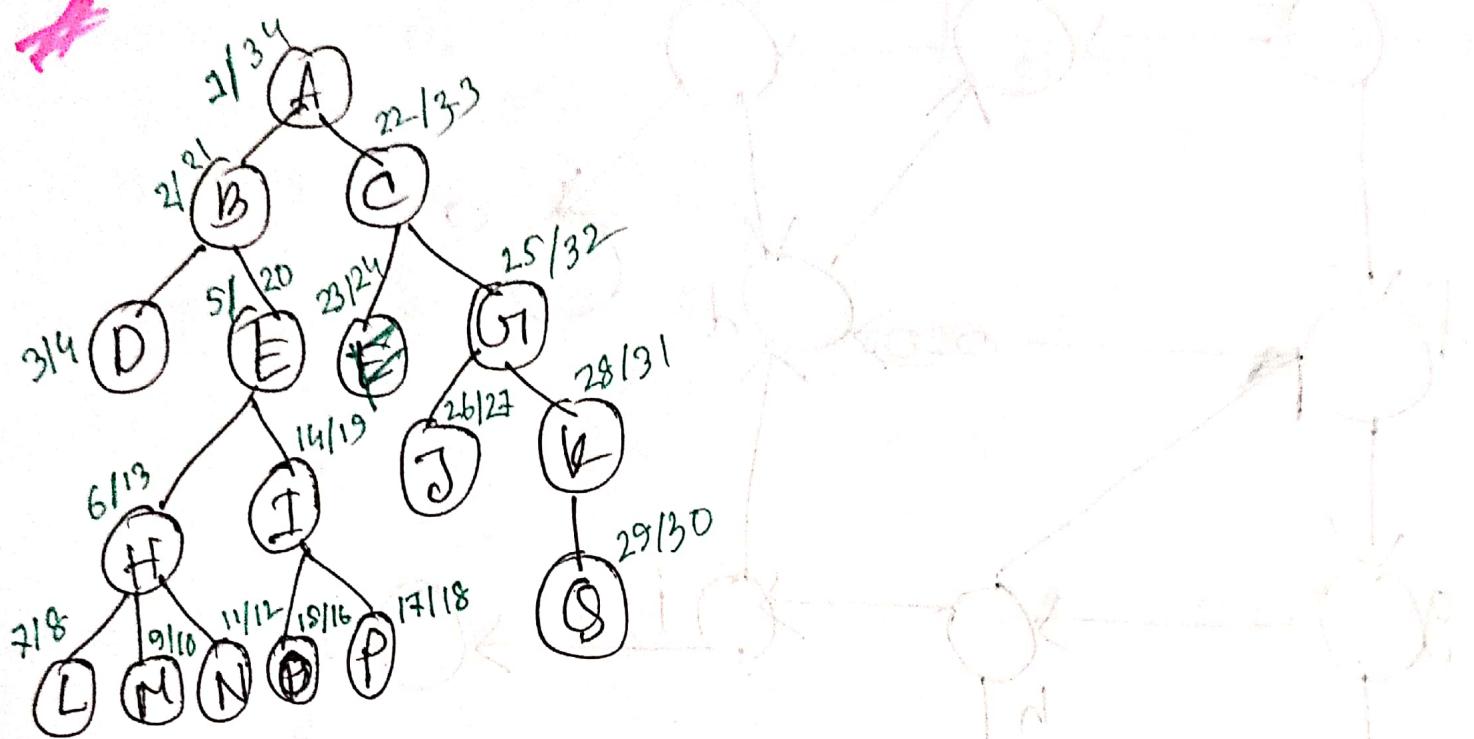
SC

EC

FD

FE

GC



BS Critical finish time

A C G K S J ~~F~~ B E I P O H N M L D

topological sort.