

NAME : JANZAID KHALIQ

ID : 15386

PAPER : DATA STRUCTURE & ALGORITHMS

EXAM : FINAL TERM 2020

TEACHER : SIR ADIL

DATE : 24/06/2020



# Page # 1

Q1: Sort the given list using Insertion Sort.  
56, 59, 45, 40, 43, 55

Solution:-

$$n = 6$$

$$\text{Steps} = n - 1$$

$$= 6 - 1 = 5$$

Step # 1 : element = 59

56, (59), 45, 40, 43, 55

element 59 is large from 56

so

56, 59, 45, 40, 43, 55

Step # 2 : element = 45

56, 59<sup>↗</sup>, (45), 40, 43, 55

56<sup>↗</sup>, (45), 59, 40, 43, 55

45, 56, 59, 40, 43, 55

Step # 3 : element = 40

45, 56, 59<sup>↗</sup>, (40), 43, 55

45, 56<sup>↗</sup>, (40), ~~59~~, 43, 55

45<sup>↗</sup>, (40), 56, 59, 43, 55

40, 45, 56, 59, 43, 55

Step # 4 : element = 43

40, 45, 56, 59<sup>↗</sup>, (43), 55

40, 45, 56<sup>↗</sup>, (43), 59, 55



## Page # 2

40, 45, (43), 56, 59, 55

40, (43), 45, 56, 59, 55

~~40, 43, 45, 56, 59, 55~~

40, 43, 45, 56, 59, 55

Step # 5 : Element = 55

40, 43, 45, 56, 59, (55)

40, 43, 45, 56, (55), 59

40, 43, 45, 55, 56, 59

So

40, 43, 45, 55, 56, 59

List has been sorted.



Q2 :- Construct Binary Trees from given list of numbers and they verify tree.

25, 15, 35, 17, 33, 36, 25, 13, 15, 40, 38, 42, 20

Solution :-

Rules :-

1) whatever is the first number of the list becomes the Root of the Binary Tree.

2) if (value of child node element  $\geq$  value of Parent node element)

then

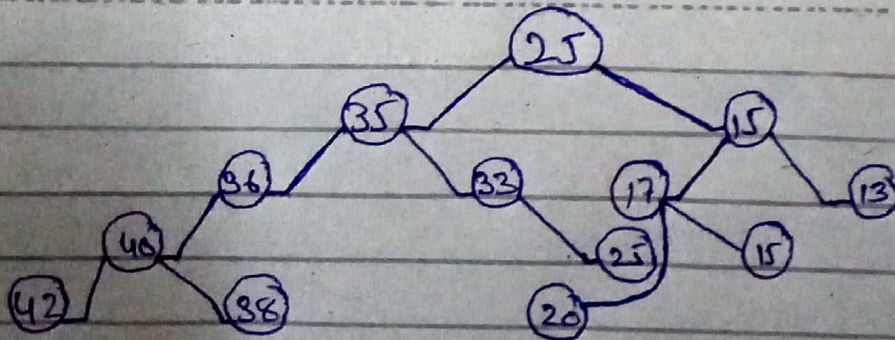
Make new number the left child node

else

Make new number the Right child node.

So,

25, 15, 35, 17, 33, 36, 25, 13, 15, 40, 38, 42, 20



Verification using in order Traversal :-

42, 40, 38, 36, 35, 33, 25, 25, 20, 17, 15, 15, 13



Q4 - Apply all the three Binary Tree Traversal Techniques on each of the Tree constructed in Q #3.

Solution :-

∴ Rules :-

There are three Traversing techniques for Binary Trees.

1) In-order Traversal :-

- Visit Left-Sub Tree
- Process Parent Node
- Visit Right-Sub Tree

2) Pre-order Traversal :-

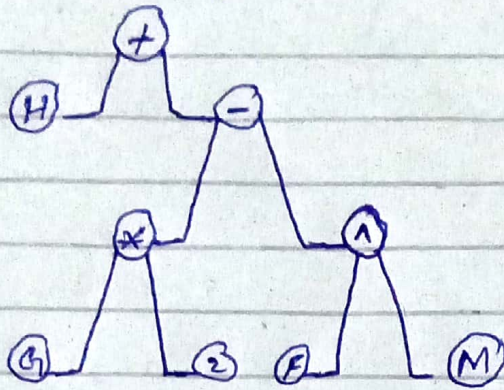
- Process Parent node
- Visit Left Sub Tree
- Visit Right Sub Tree

3) Post-order Traversal :-

- Visit Left-Sub Tree
- Visit Right Sub Tree
- Process Parent Node



i)



In order Traversal:-

H, +, G, \*, 2, -, F, ^, M

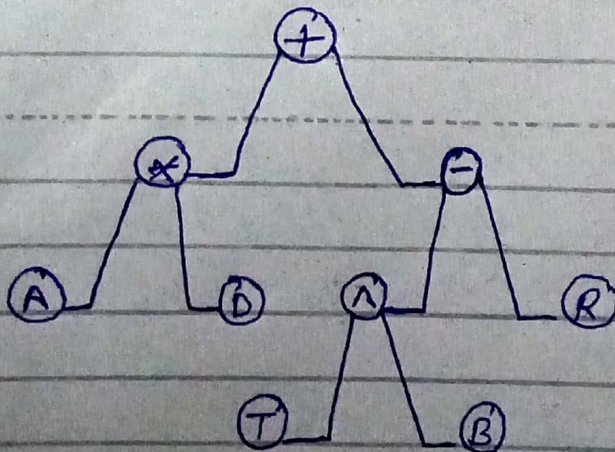
Pre - order Traversal:-

+, H, -, \*, G, 2, ^, F, M

Post order Traversal:-

H, G, 2, \*, F, M, ^, -, +

ii)



In order Traversal:-

A, \*, D, +, T, ^, B, -, R

Pre order Traversal:-

+, \*, A, D, -, ^, T, B, R

Post-order Traversal:-

A, D, \*, T, B, ^, R, -, +



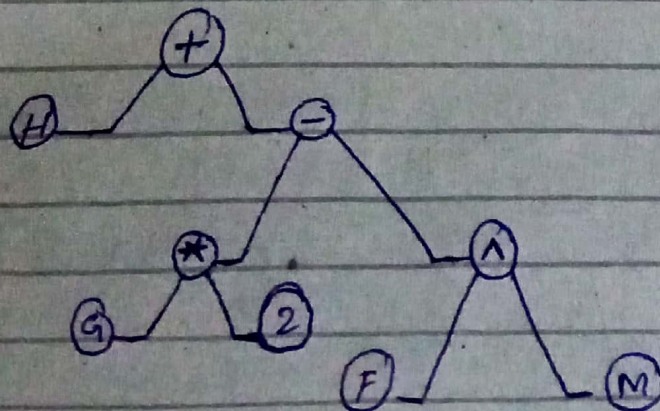
Q3 :- Construct Binary Trees from given mathematical expressions

- i)  $H + G * 2 - (F \wedge M)$
- ii)  $A * D + T \wedge B - R$

Solution :-  $\therefore$  Rules :-

- An operator can be a Parent Node as well as a child Node.
- An operand can never be a Parent Node. An operand is always a leaf/terminal node.
- Always select an operator of least priority.
- In case of more than one operator with same priority, select the left most one.
- Properly underline both the corresponding sides of the selected operator.
- Left side of the selected operator becomes the left-sub tree.
- Right side of the selected operator becomes the right-sub tree.

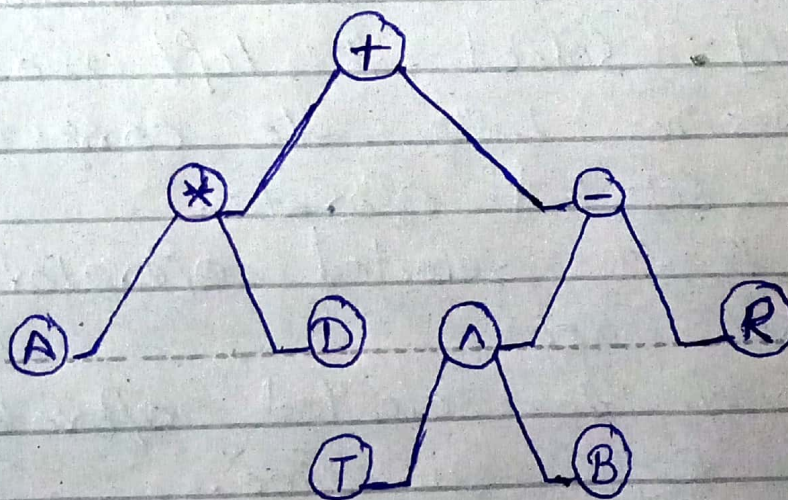
i)  $H + G * 2 - (F \wedge M)$





Page # 7

ii)  $A * D + T \wedge B - R$





Page # 8

Q5 :- Fill in the blanks.

- ① Nodes
- ② Branch
- ③ Zero
- ④ Same family
- ⑤ older
- ⑥ Younger
- ⑦ Non-Linear
- ⑧ Forest