

Question 01

1. a. PreOrder: A B C D E F G H I

Raj
Mech(ECE)

InOrder: B C A E D G H F I

b. PreOrder: G, B, G, A, C, K, F, P, D, E, R, H

InOrder: G, A, K, C, F, A, G, P, E, D, H, R

c. Inorder: B P R E S Q G

PreOrder: G E P B R S Q G

2. Evaluate the following prefix expression

$$^{*}++36+-1324$$

3. Convert the following infix expression to post fix notation.

$$((a+2) * (b+4)) - 1$$

4. Evaluate the postfix expression

a. 9, 6, 5, +, 5, *, +, 10, +

b. 15, 3, +, 2, 1, 8, *, 10, -

c. 8 2 3 ^ / 2 3 * + 5 1 * -

5. InOrder: E A C K F H D B G

PreOrder: F A E K C D H G B

6. Draw the Binary Search Tree (BST) from the insertion of the integer keys 58, 32, 96, 107, 26, 12, 11, 92, 10, 25, 51, 21. Traverse the tree in Pre, In and Post-order.

7. Draw the BST for the sequence 3, 1, 4, 6, 9, 2, 5; & one find traversals.

8. Draw the BST of first 10 Fibonacci numbers. & one find traversals.

9. Draw the BST for the sequence 58, 72, 96, 94, 107, 26, 12, 11, 9, 2 & find the traversals.

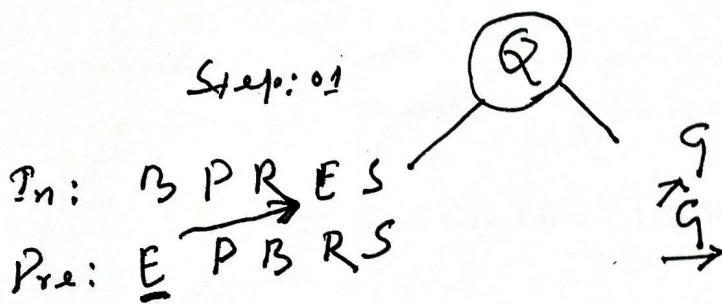
10. Convert $a+b*c+(d+e+f)*g$ to pre & post expression where $a=1, b=2, c=3, d=4, e=5, f=6$

Construct a BT

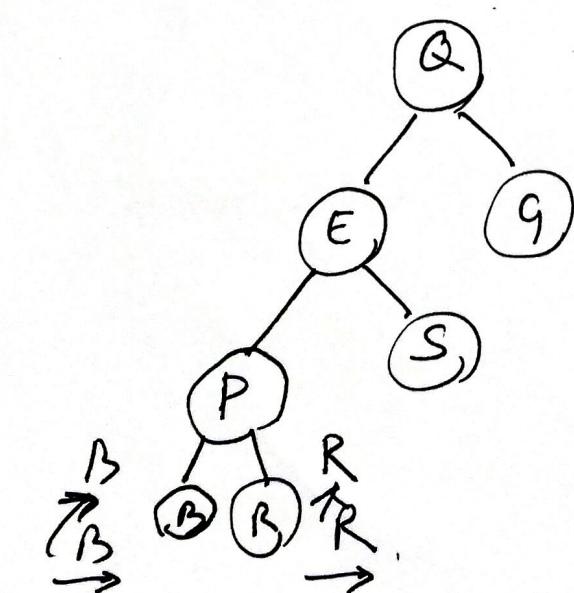
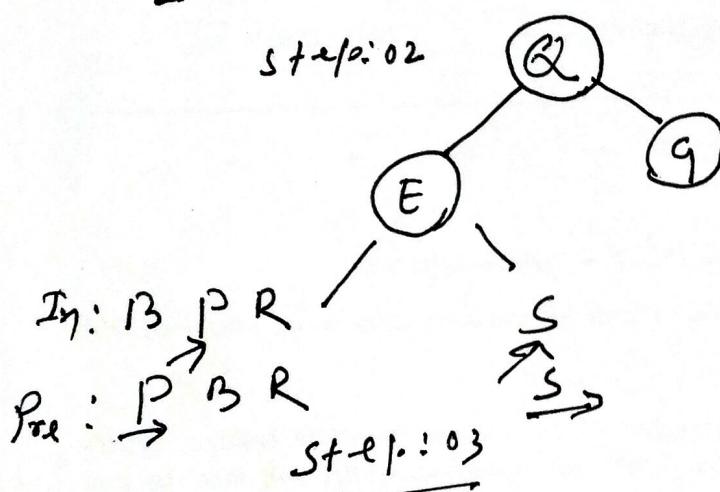
From Post-Order.

Construct a BT

InOrder: B P R E S Q G F { Inorder: B C D F }
PreOrder: Q E P B R S G { PreOrder: + 2 3 S - }



Inorder: B C D F
PreOrder: D F C



Construct a BT

From Post-Order.

& In Order.

Given:

PostOrder: D F E B G L J K H C A

InOrder: D B F E A G C L J H K

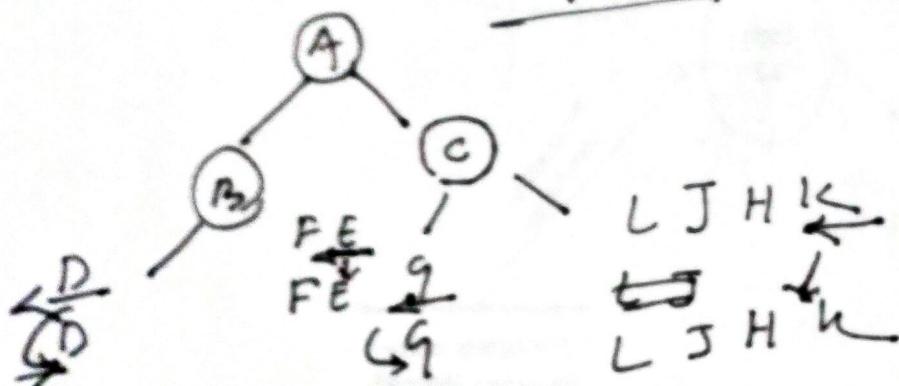
Note: Construct a BT, start-

Post: D F E B
In: D B F E

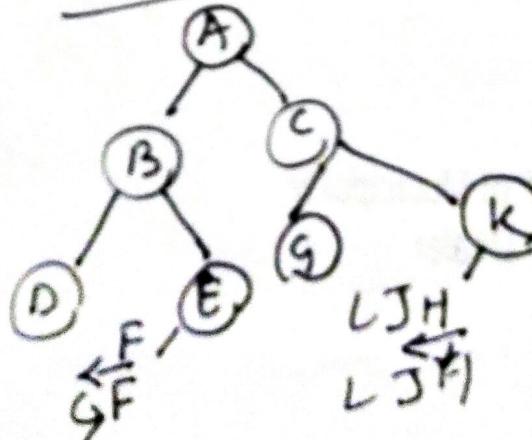


G L J K H C : Post-
G C L J H K : In

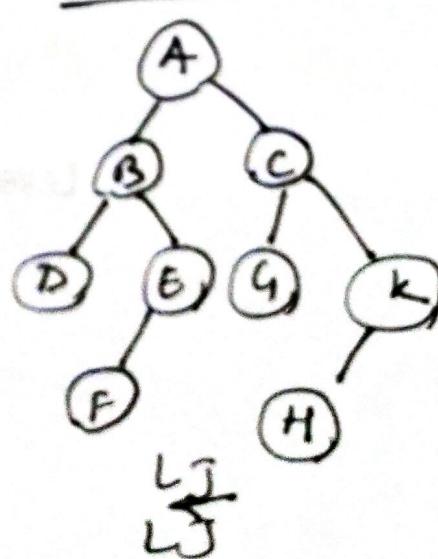
step: 02

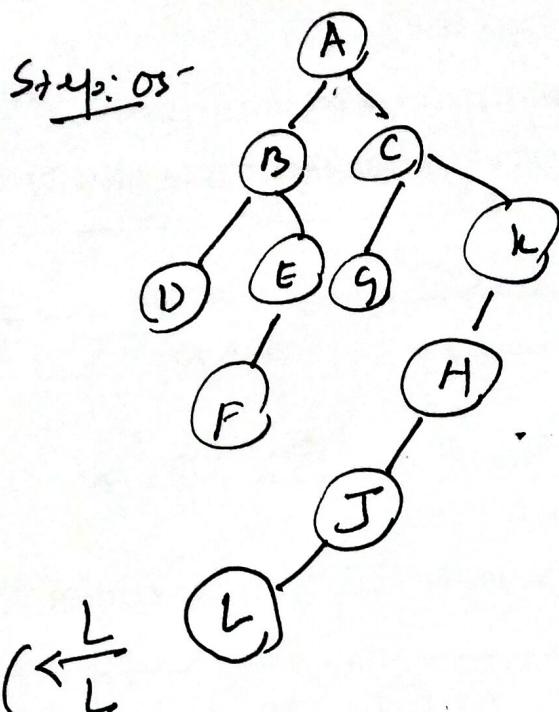


step: 03



step: 04

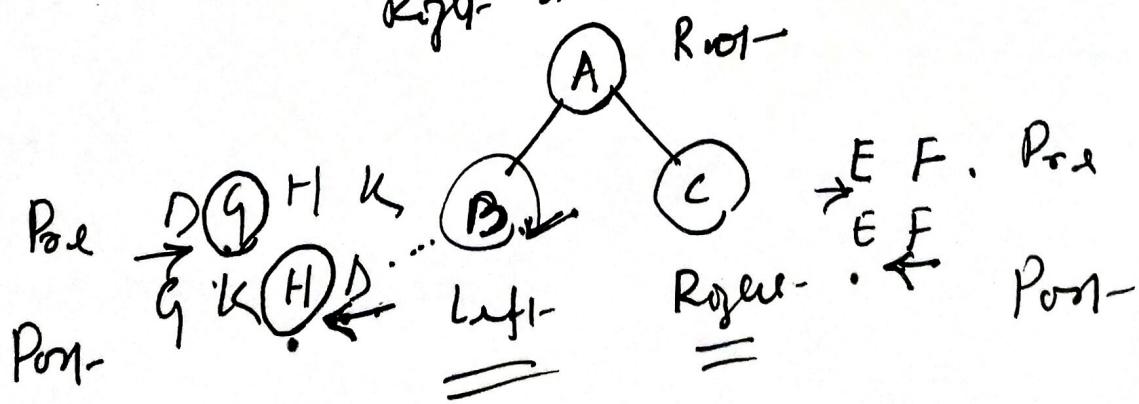




Construct a BT
from PreOrder & Post-Order.

Given : PreOrder : $\xrightarrow{L} \xrightarrow{R}$
 $A \xrightarrow{L} B \xrightarrow{R} D \xrightarrow{L} G \xrightarrow{R} H \xrightarrow{L} K \xrightarrow{R} C \xrightarrow{L} E \xrightarrow{R} F$
 Post-Order : $\xleftarrow{L} \xleftarrow{R} G \xleftarrow{L} K \xleftarrow{R} H \xleftarrow{L} D \xleftarrow{R} B \xleftarrow{L} E \xleftarrow{R} F \xleftarrow{L} C$

Note : { Pre that is left side
 } & Post that is right side.
 ① Adjacent elem. of left side
 is used as left node.
 } And Adjacent element of
 right side is used as
 right node.



Frequency Count:

The number of times the statement executing in a program / algorithm is known as frequency count.

Thus, the performance of a program (or algorithm) is measured in terms of "time complexity" that is total time taken by the algorithm to execute.

Ex: Consider the following statement-

$$x = n+1$$

Now, this statement will execute only once.

Hence Frequency count of this statement is said to be 1.

Now, Time Complexity ($T(n)$) is

$$T(n) = 1$$

OR, $T(n) = O(1)$ (constant-time)

Ex: if (a < 5) \rightarrow Executes once.

{

$$b = 10;$$

}

Frequency Count = 1

Exp:

$a = 1$; ————— Executes Once
 while ($a < 5$) ————— Executes 50x

{

 printf ("a = %d", a); ————— Executes five
 a++; ————— Executes five
 }

Hence, we can say that total frequency count of above code is :

$$1 + 6 + 5 + 5 = 17$$

$$T(n) = 17$$

$$\text{OR } T(n) = O(1)$$

Exp:

for ($i=1$; $i \leq n$; $i++$)
 {
 for ($j=1$; $j \leq i$; $j++$)
 printf ("%d", a[i][j]); ————— n
 }

Hence, frequency count - :

$$1 + (n+1) + n + n^2 = 3n + 1$$

$$T(n) = 3n + 1$$

$$\text{OR } T(n) = O(n) — \text{Linear.}$$

Exp:

for ($i = 1$; $i \leq n$; $i++$)

for (j = 1 ; j <= n ; j++)

(\Rightarrow) $\text{print-f}(\text{"x.y"}, \alpha[i]);$ $\text{m} \leftarrow$

The nested³ execution will be:

$$(1 + (n+1) + n) \times [(1 + (n+1) + n) + n] \\ = (2n+2) \times (3n+2)$$

$= 6n^2 + 10n + 4$ is the frequency count.

$$T(n) = 6n^2 + 10n + 4$$

OR $T(n) = O(n^2)$ (Square)

Ex: To find frequency count and time complexity.

for (i=1 ; i<=n ; i++)

for (j=1 ; j<=n ; j++)

for $\{$ $k=1$; $\text{let } z = n$; $k++$)

```
    }  
printf("Hello");
```

Ex: Find the frequency count for the following piece of code

Sum = 0;

for (i=1 ; i < n ; i++)

 Sum = Sum + a[i];

Ex: Calculate the runtime complexity of following program segment:

i = 1

loop (i <= n)

 print - i

i = i + 1

Ex: Find O, Ω , Θ for the given function $f(n) = 10n^2 + 3n + 5$.

Ex: Find time complexity

for i := 1 to n do

 for j := i+1 to n do

 for k := j+1 to n do

$$2 = 2 - 1$$