

Assignment - 6

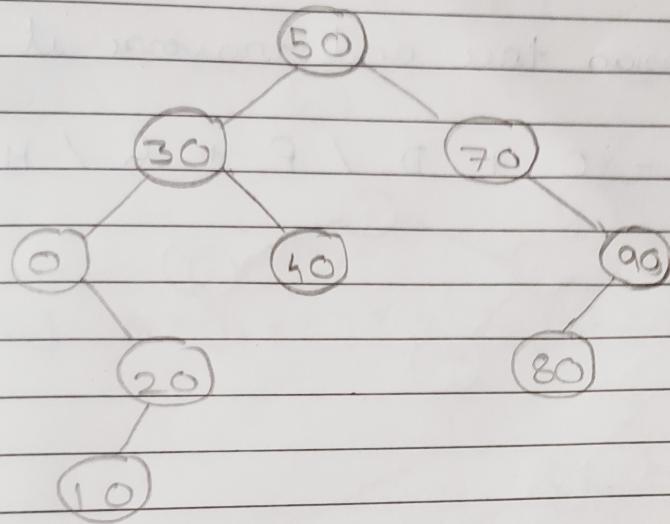
Q1. Algorithm for evaluating an expression using a tree.

```
Let t be the expression tree  
If t is not null then  
  If t.value is operand then  
    Return t.value  
  A = solve(t.left)  
  B = solve(t.right)  
  // calculate applies operator 't.value'  
  // on A and B, and returns value  
  Return calculate(A, B, t.value)
```

Q

Draw a binary search tree and traverse in-order, pre-order and post-order

i) 50, 30, 0, 20, 70, 90, 80, 10, 40

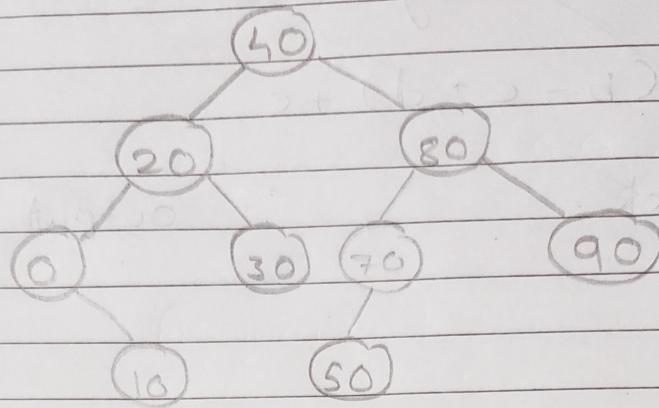


inorder : 0, 10, 20, 30, 40, 50, 70, 80, 90

preorder : 50, 30, 0, 20, 10, 40, 70, 90, 80

postorder: 10, 20, 0, 40, 30, 80, 90, 70, 50

2 40, 20, 0, 30, 80, 90, 70, 10, 50



inorder : 0, 10, 20, 30, 40, 50, 70, 80, 90

pre : 40, 20, 0, 10, 30, 80, 70, 50, 90

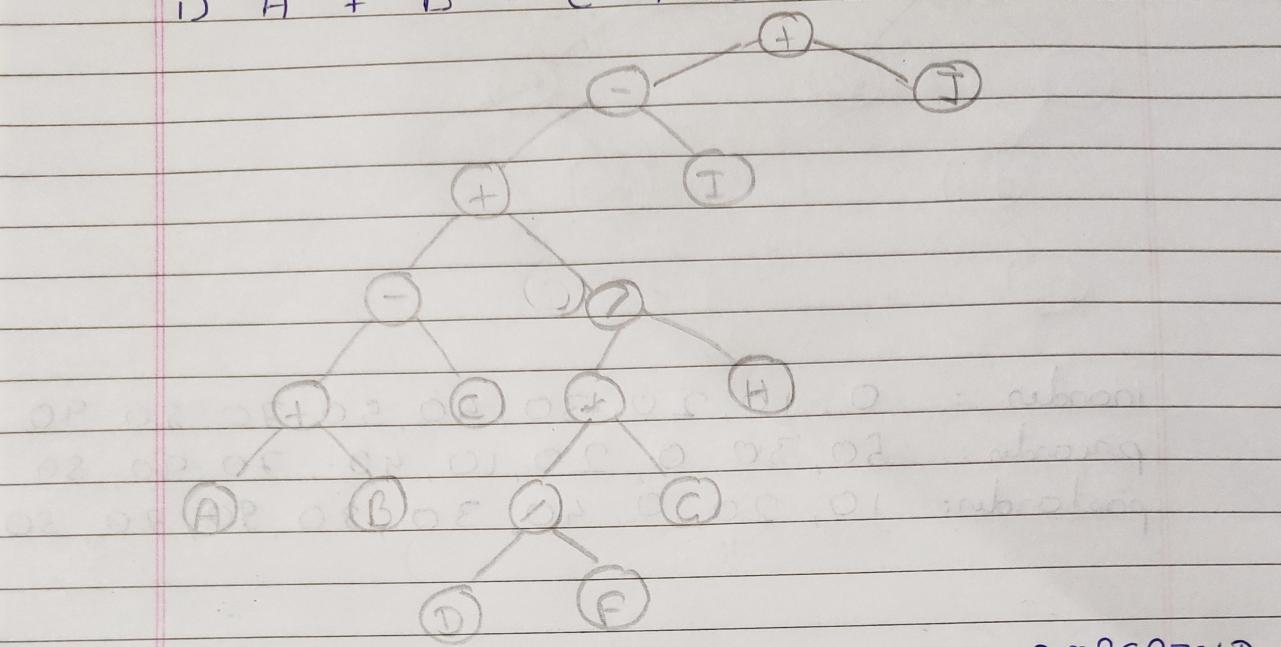
post : 10, 0, 30, 20, 50, 70, 90, 80, 40

20BCP304D

Q

Draw expression tree and traverse it

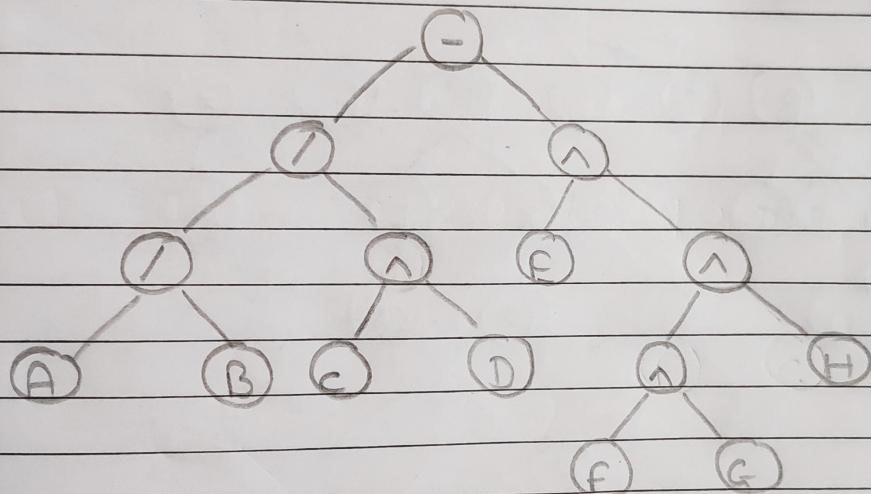
i) $A + B - C + D / F + G / H - I + J$



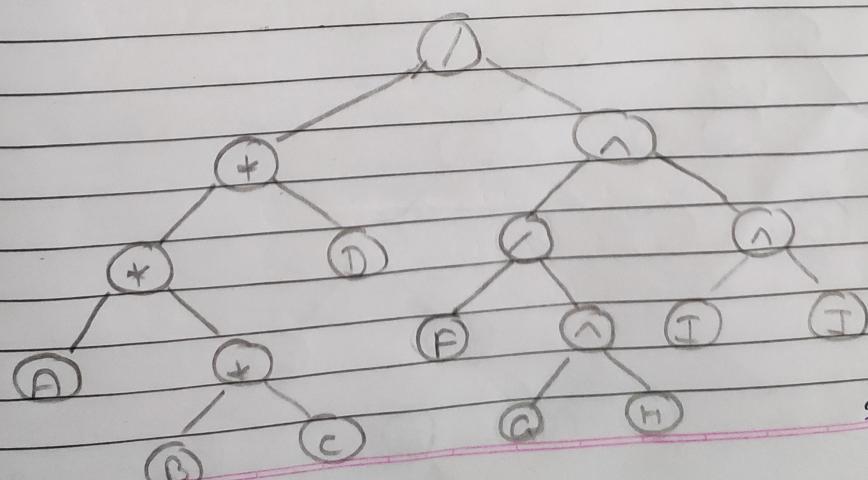
pre : $+ - + + ABC / + / D F G H I J$
 post : $A B + C - D F / G * H / + I - J +$

ii) $A / B / C \wedge D - E \wedge (F \wedge G) \wedge H$

pre : $- / / A B \wedge C D \wedge E \wedge F G H$
 post : $A B / C D \wedge / E F G \wedge H \wedge -$



iii) $A * (B + C) * D / (F / G \wedge H) \wedge I \wedge J$



pne order:

$/ * * A * B C D \wedge / F \wedge G H \wedge I J$

post order

$A B C * * D * F G H \wedge \wedge I J \wedge \wedge /$

20BCP304D

Q4.

