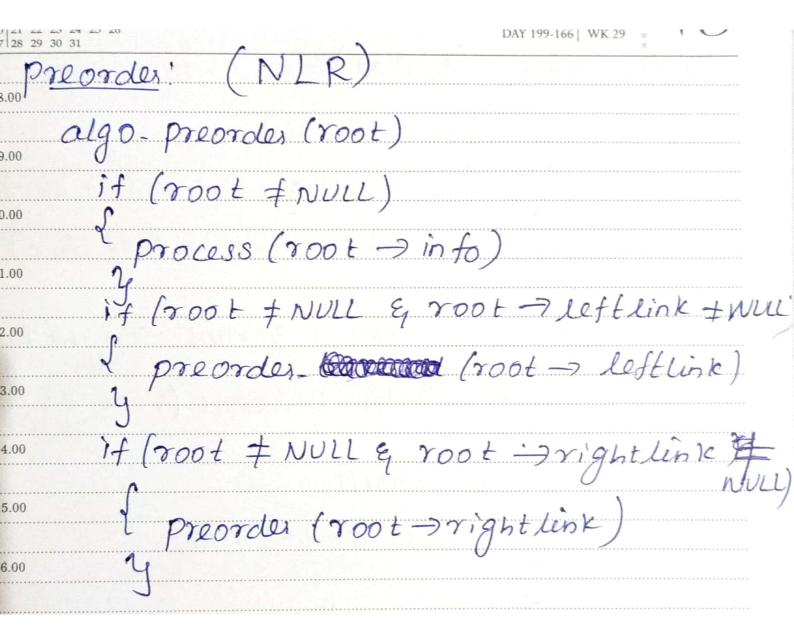
08.00	norder algorithm! (LNR)
09.00	algo inorder (root)
10.00	if (root + null & root -) leflink + NULL
11.00	inorder to constant (root -) lett link
12.00	Y
***************************************	if (root = null)
13.00	
14.00	Process (root → info)
15.00	if (root frull & root rightlink fw
16.00	In order toot - right link
17.00	· · · · · · · · · · · · · · · · · · ·



09.00	algo postordes (root)
10.00	if (root + null & root->lest link 7 NU
10.00	· · · · · · · · · · · · · · · · · · ·
11.00	postorder + (root -) left link)
12.00	if (root #NULL & root Dright link +
13.00	Postorder-traversal (root > rightlin
4.00	Y
5.00	if troot frull
6.00	process (root -) info)

```
Slarching in Binary Slarch Tree.
    algo-bst search (root, item, par, Loc)
          y return:
(root -> info 7item) ptr=root -> left
      flue ptr = root -> right

Bomp: root

while (Ptr + null)

if (Ptr->into==item)
            1 loc = Ptr, Pas = tomp &
             2 return
                       APPOINTMENTS
NOTES
```

ALT I MAN MY NO. TO	
08.00 stemp = ptr if (ptr -) info Titem) ptr = p	br-) left
09.00 yelseptr = ptr -> right	
10.00 loc=ncell, pas=temp	Ж.:
11.00	
Insertion in BST	
14.00 algo_bst însext (root, newinfo)	
Call bst Sparth (root, newinfo, Par,	LOC)
16.00 Pf (loc fnull) return	
17.00 new node = get node ()	
new node -> info = newinfo new node -> rink = null new node -> rink = null	SUNDAY 23

if (par = = null)
{ root = newnode (Pas→info 7 newinfo)

par → left = newnode -> right = new node

Deleting a node from a BST:
algo del-bst (root, item)
Cau find (root, item, Pax, 10C)
if (loc = = null) return if (loc > right = Null and loc > left = Null)
Call del. two. bst (root, loc, par)
olse
call del-one-bst (root, loc, par)
free node (LOC)
return.

	23120 2/ 28 29 30
De1	ete a node with one or Tero Child!
.00	190-del-Onebst (root, Loc, par)
.00	if (10c -) left == null & coc -> right == nu
.00	Child = = null
2.00	else if Coc > deft frull
3.00	Child=loc-> deft
1.00	erge
5.00	$ \begin{array}{c} \text{Child} = \text{Loc} \rightarrow \text{right} \\ \text{Sight} \end{array} $
3.00	if (pas ≠ null)
7.00	if (loc == par -> left) par -> left= Chilo
3.00	eige
***************************************	d root = Child
NOTES	y return Appon