Assignment 2

1) Pseudo code to delete all elements $b < a$ in a given BST T and given value a:

```
del_less (T, a)
{  if (T == NULL)
   { return ; }
   else if ( T->value > a)
   { del_less (T->left, a); }
   else if ( T->value == a)
   { T-left = NULL; }
                                  // T->value < a
   else
   { T->left = NULL;
     if ( T->parent != NULL)
     { T-> parent -> left = T->right;
       T = T->right;
       T->parent = T->parent -> parent;
       del_less (T, a); }
     else
     { T = T->right;
       T->parent = NULL;
       del_less (T, a); }
   }
}
```

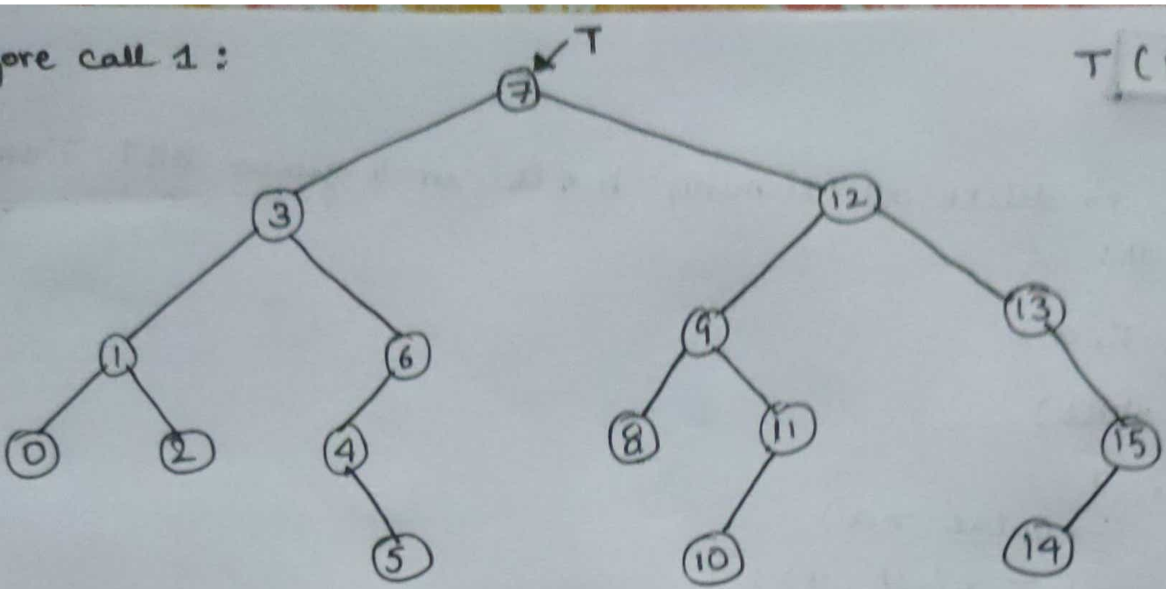
del_less (T, a) = O(height (T)) [Time Complexity of Algorithm]

∵ we are traversing down the tree and at each step the processes take constant (O(1)) time. ∴ $O(ch) = O(h)$

c → some +ve constant

Before call 1 :

T (pointer to node)



a = 11   T = 7 (root)

del_less (7, 11) ⟶ del_less (12, 11) ⟶ del_less (9, 11)

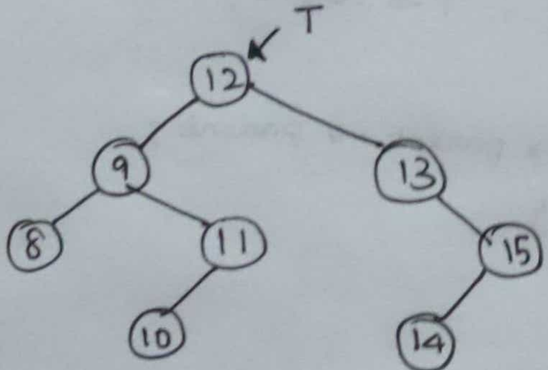
call 1   call 2   call 3

↓

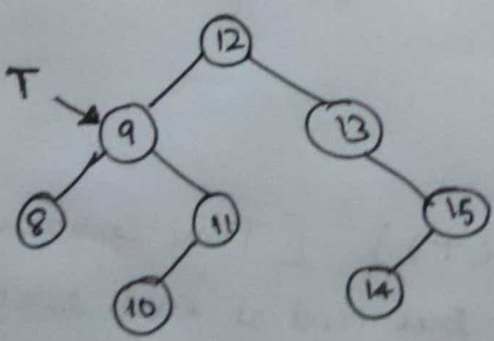
del_less (11, 11)

call 4

After call 1 :



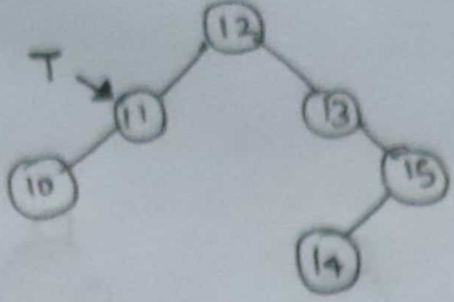
After call 2 :



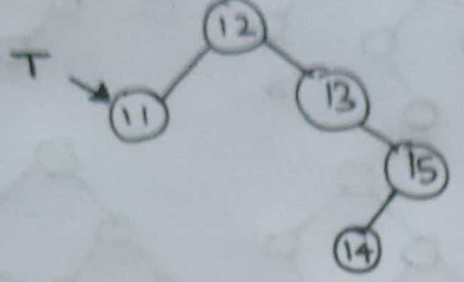
~~After call 3 :~~

P.T.O

After call 3:



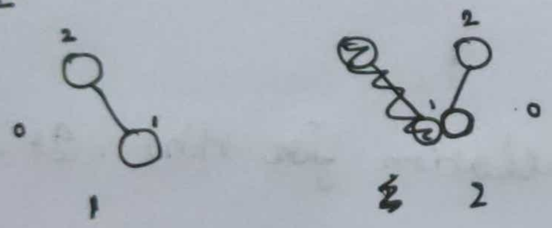
After call 4 :



2)

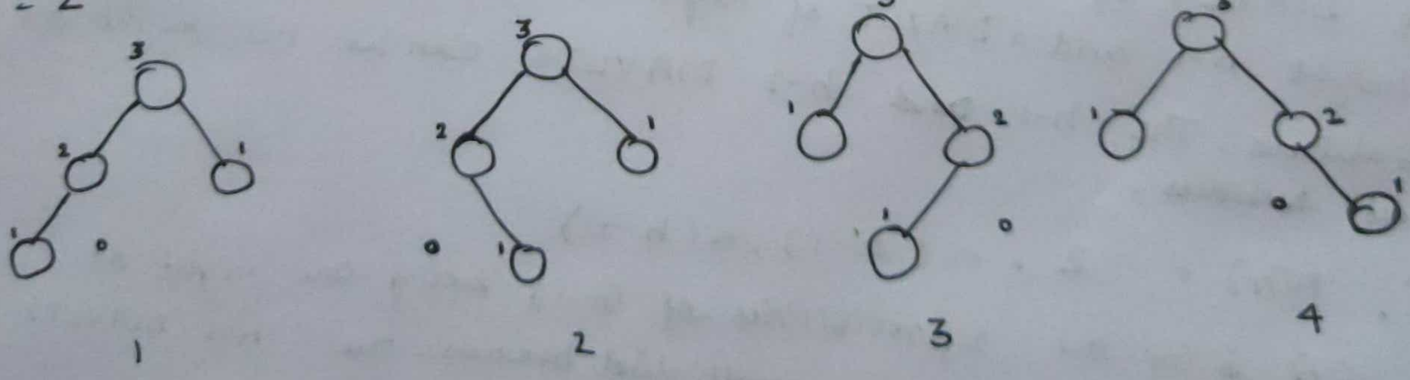
a) no. $\underbrace{\text{deeply imbalanced AVL trees}}_{\text{DIAVLT (for convenience)}}$ of height (h) = n(h)

h = 1



n(1) = 2

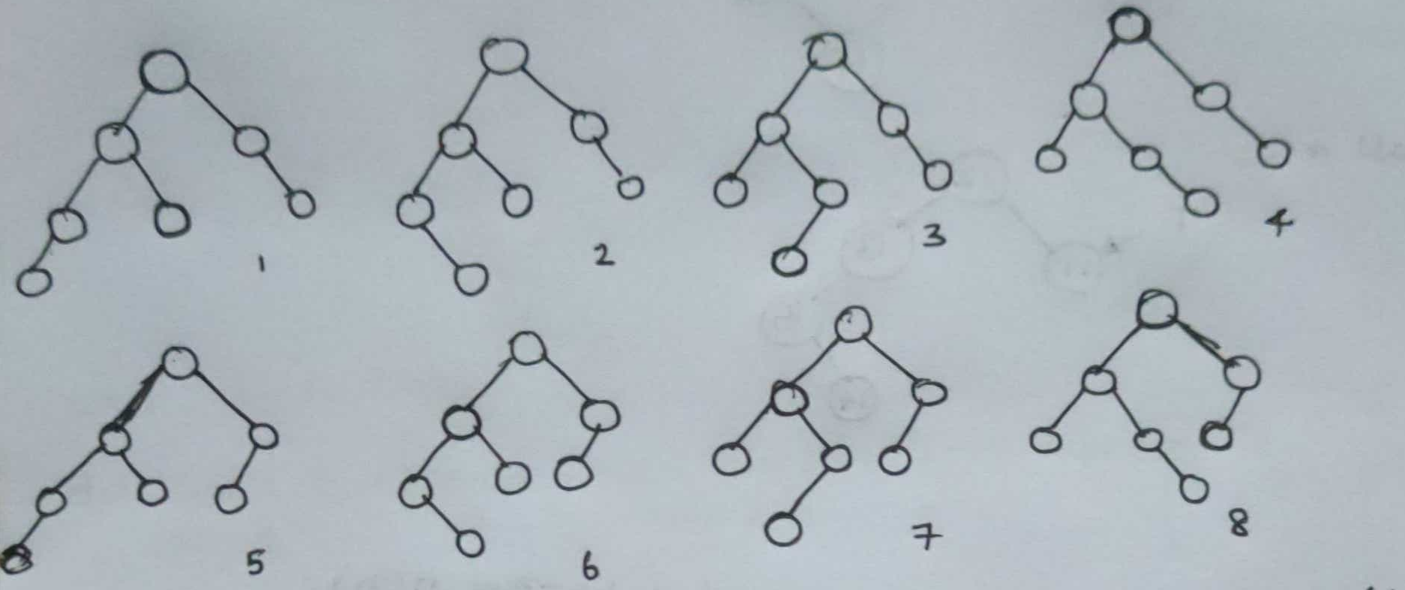
h = 2



n(2) = 4

h = 3

~~n(3) = 16~~



another 8 deeply imbalanced AVL's with the left and right subtrees of the root in these trees switched.

$n(3) = 8 + 8 = 16$

b) Yes there is a recurrence relation for n(h). It is:

$$n(h) = 2 \cdot n(h-1) \cdot n(h-2)$$

A DIAVLT of height h must have a DIAVLT of height h-1 and a DIAVLT of height h-2 as it's root's 2 subtrees. The h-1 and h-2 DIAVLT's can be the right or left subtrees.

$\therefore$ $n(h) = 2 \cdot n(h-1) \cdot n(h-2)$

(2 is for the 2 possibilities of (h-1) being the right or left subtrees) (they are multiplied because the h-1 DIAVLT's arrangements and h-2 DIAVLT's arrangements are independent of eachother)

$n(1) = 2 \cdot n(0) \cdot n(-1)$

$n(0) = 1$ (only root node)

$n(-1) = 1$ (defn/set value cause tree with $h = -1$ is no tree at all)

$n(1) = 2 \cdot 1 \cdot 1 = 2$ True

1 1
2 2
3 4
4 7

$n(2) = 2 \cdot n(1) \cdot n(0)$

$= 2 \cdot 2 \cdot 1 = 4$ True

$n(3) = 2 \cdot n(2) \cdot n(1)$

$= 2 \cdot 4 \cdot 2 = 16$ True

$\therefore n(4) = 2 \cdot n(3) \cdot n(2)$

$= 2 \cdot 16 \cdot 4 = 128$

Ans: $n(4) = 128$