Date:

DDA.

M T W T F S

Merge (L, R)

m = length (L) + length (R)

S = empty Array of size (m)

i = 1, j = 1, $\boxed{k = 1}$

for k = 1 to m

if L[i] ≤ R[j]

S[k] = L[i]

L++

else

S[k] = R[j]

j = j+1

Return S

while (i ≤ length (L) && j ≤ length (R))

{ if (L[i] ≤ R[j])

S[k] = L[i]

i++, k++

else

S[k] = R[j]

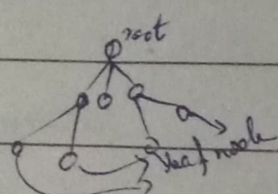
j++, k++

}

```
MergeSort (A)
    n = length (A)
        if (n ≤ 1)
            return A
    L = MergeSort (A [1.... floor(n/2)])
    R = MergeSort (A [floor(n/2) + 1 .... n])
return merge (L, R)
```

Tree

Root → no parent, ya 1 he.
last node → leaf node



(1) Binary Tree :-

Each node have max. 2 children.
min. 0

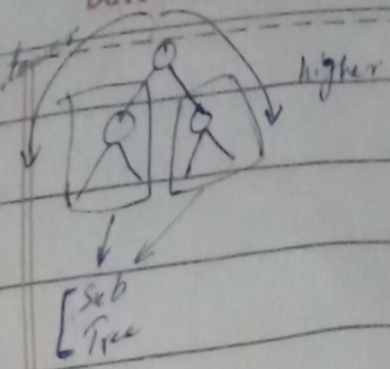
null Tree is also binary Tree.

(2). Binary Search Tree

if x is a node in binary search tree, then

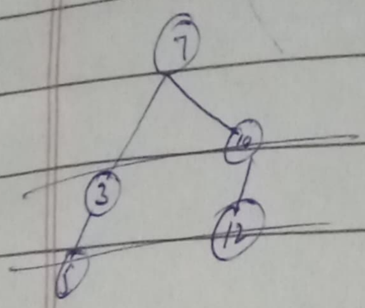
node y on left side of x will be y.key < x.key and node y on right-side of x will be y.key ≥ x.key

Date:



left-side value

```
struct node
{
    int key;
    node* left;
    node* right;
}
```



Traversing

سب ایلیمنٹس دو سے گزرنا

مثلاً سب کا اور ایلیمنٹس کا مربع نکالنا۔

i. InOrder — LNR
ii. PreOrder — NLR
iii. PostOrder — LRN

[Left sub tree
right " "
node

key ⇒ int, float (type)
left / right

InOrder (x) → root ka address, if not ⇒ null.
if(x ≠ null)

InOrder(x.left)

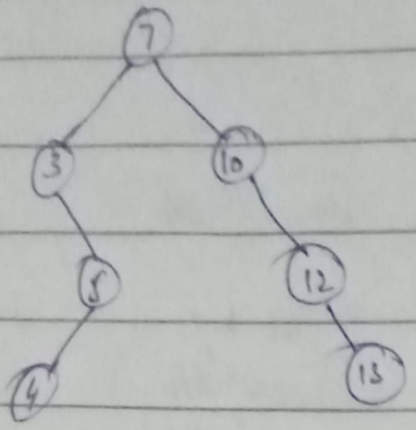
print x.key

InOrder(x.Right)

$\boxed{3, 4, 5, 7, 10, 12, 13}$

Date:

M T W T F S



PreOrder(x)

if (x ≠ null)

x.print

InOrder(x.left)

Inorder(x.right)

---

social media → semi structured data.

data mining → knowledge discovery

↓

prediction.

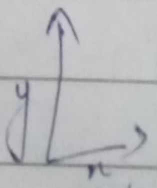
Youtube Recommandation.

on Home Page

Clustering, classification, Support Vector Machine

churn

dependent ↙ y



independent

Data Frame → df variable name.