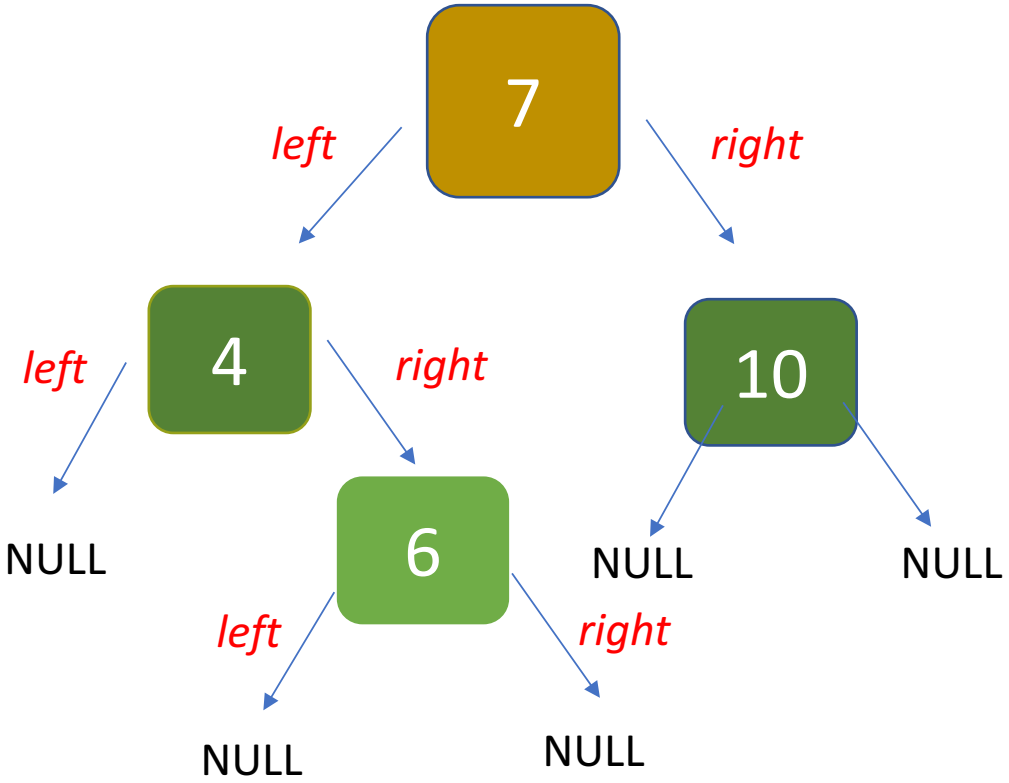
Functions

DELETE
prima defoglio poi stacco la root



Clear (root=7)

{ clear (7 --> left) che e' 4

{ clear (4 --> left) che e' NULL

clear (4--> right) che e' 6

return

{ clear (6 --> left) che e' NULL

clear (6--> right) che e' NULL

delete 6 }

delete 4}

clear (7-->right) che' 10

clear (10 --> left) che e' NULL

clear (10 --> right) che e' NULL

return

delete 10}

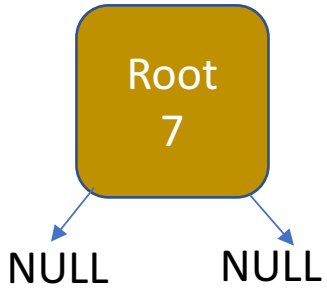
delete 7}

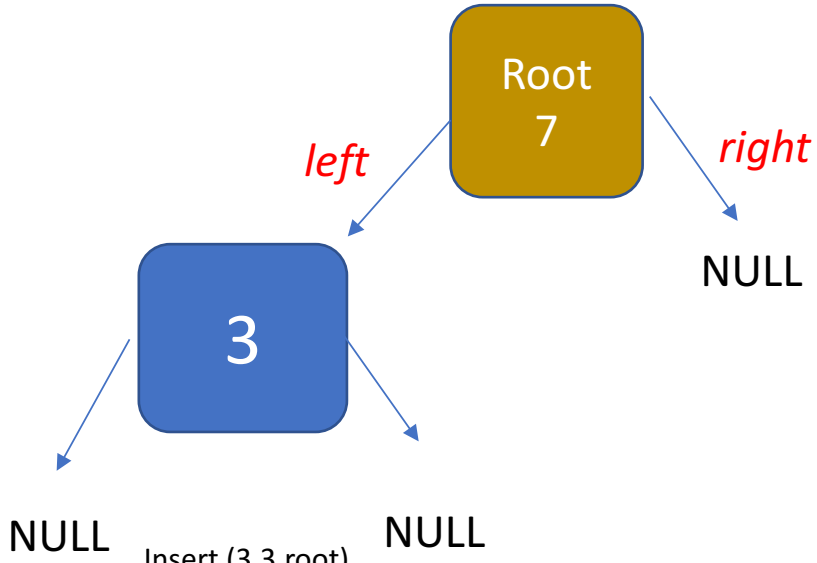
Root=t==nullptr

NULL

Insert (7,7,root)

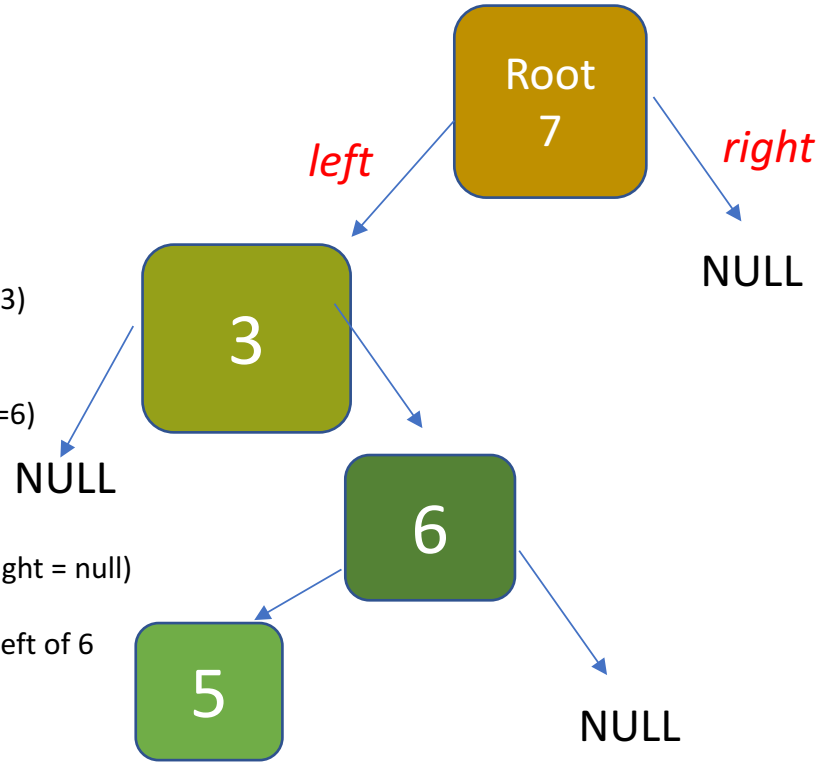
t = newnode





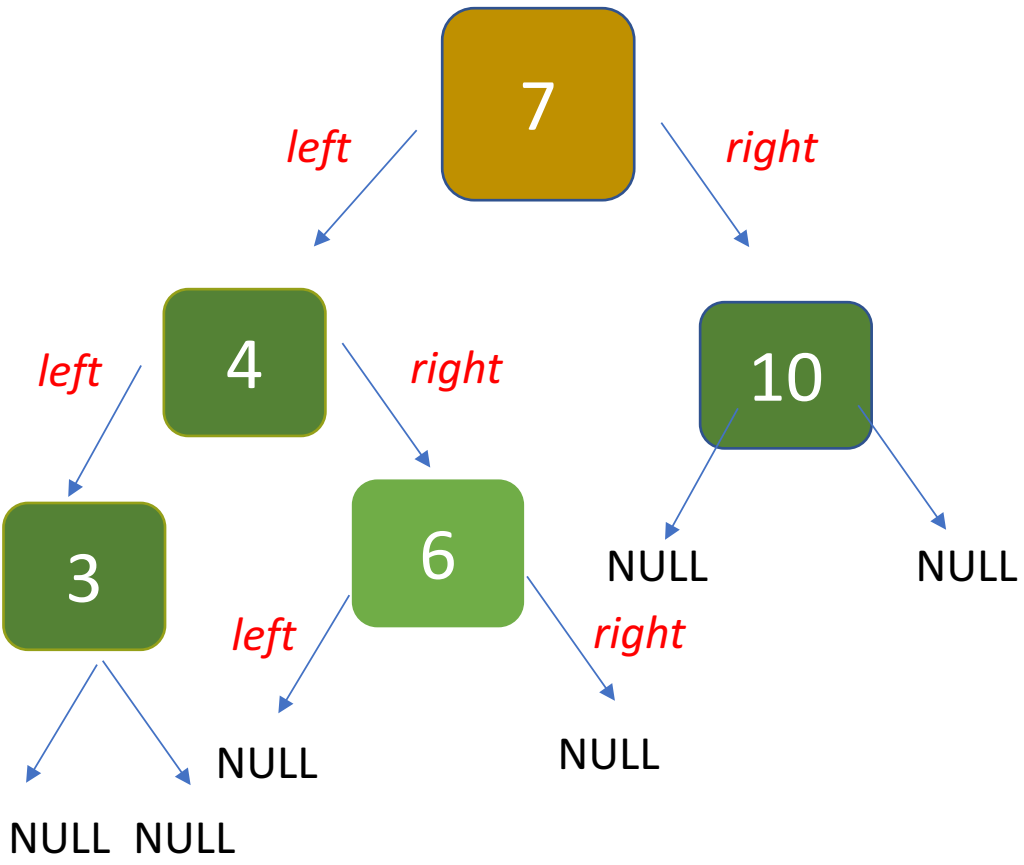
Insert (3,3,root)
t=root != nullptr
(perche' t e' 7)
3 < t-->key =7 ?
Insert (3,3, t→ left =null)
T—> left is newnode

Insert (5,5,root)
t=root != nullptr
(perche' t e' 7)
5< t-->key =7 ?
Insert (5,5, t→ left =3)
t=3 and t!=null
Check 5> t ? (t=3)
Insert (5,5, t→ right=6)
t=6 and t !=null
Check 5< t ? (t=6)
YESSSS
New node (5,5,t →right = null)
T=null finally
I can insert 5 at the left of 6



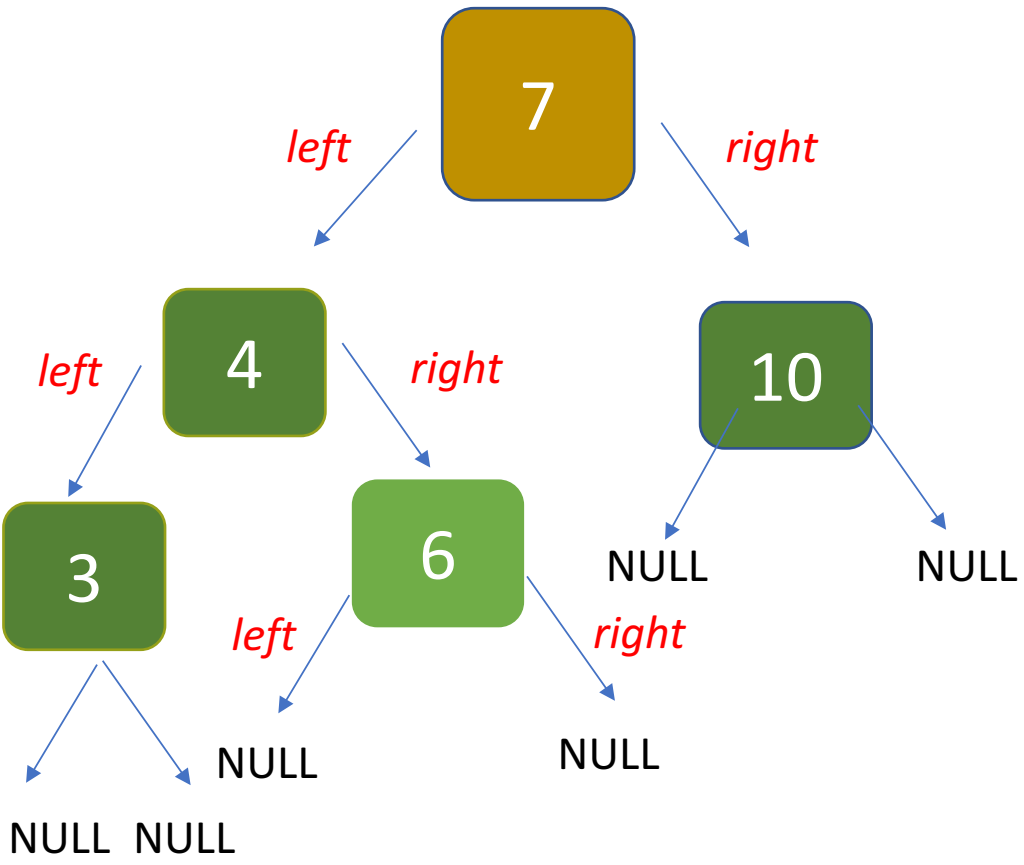
INSERT

Print in order



Inorder (root)=inorder(7)
inorder (7)
{ inorder(7 --> left) che e' 4
{ inorder (4 --> left) che e' 3
t=3
inorder (3--> left) che e' NULL
t=NULL
return;
cout << 3 → 3
inorder (3-->right)
t=NULL
return;
cout << 4 → 4
inorder (4→ right) che e' 6
t=6
inorder(6--> left) che e' NULL return
cout << 6 → 6
inorder(6--> right) che e' NULL return
cout << 7 → 7
inorder (7--> right) che e' 10
t = 10
inorder (10 →left) che e' NULL return
cout << 10 ecc… → 10

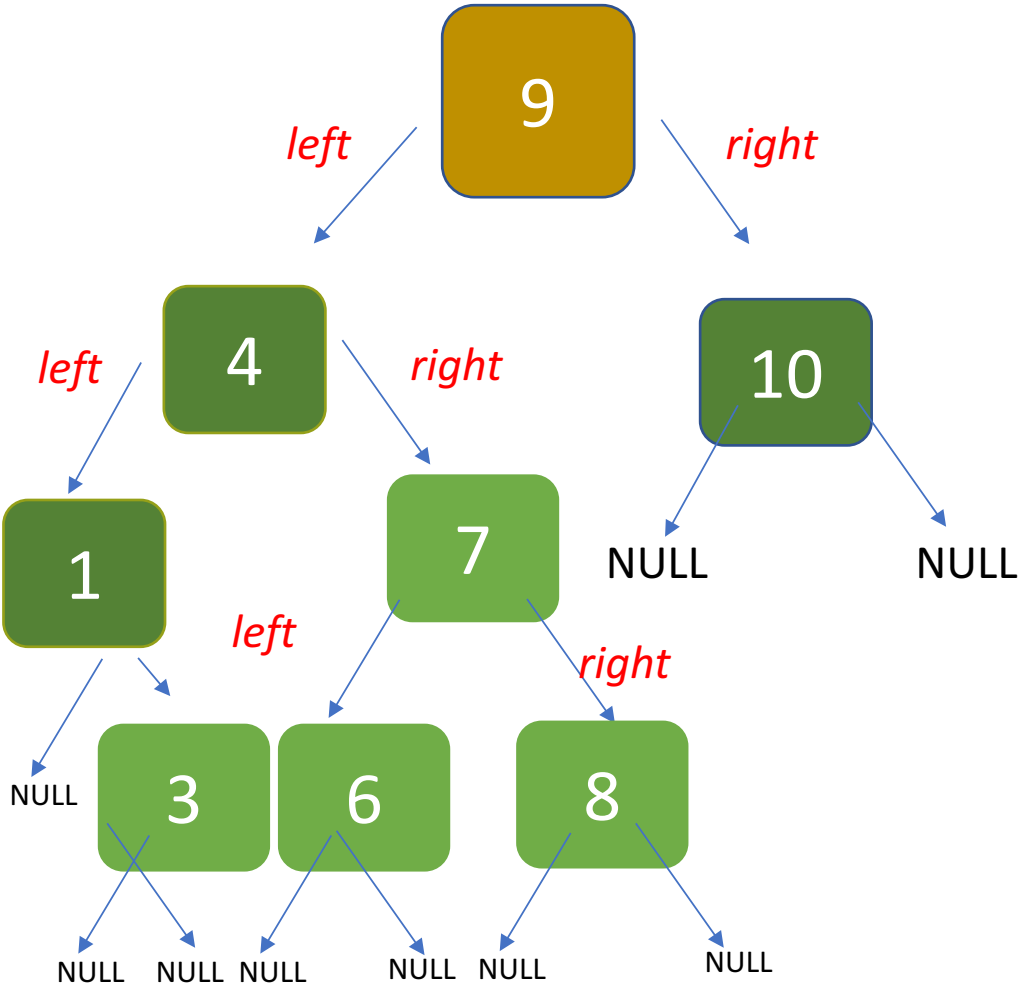
Balance

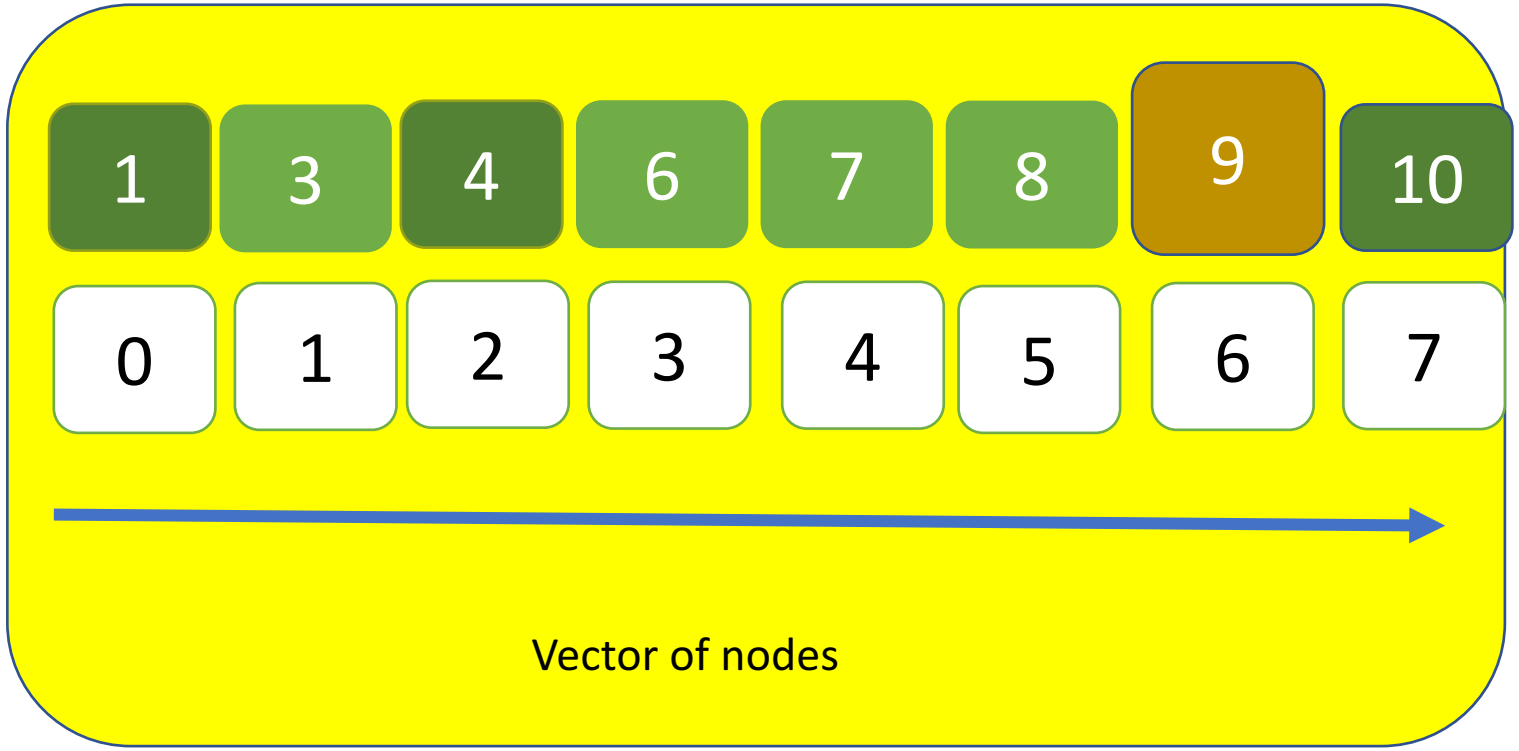


Inorder (root)=inorder(7)
inorder (7)
{ inorder(7 --> left) che e' 4
{ inorder (4 --> left) che e' 3
t=3
inorder (3--> left) che e' NULL
t=NULL
return;
cout << 3 → 3
inorder (3-->right)
t=NULL
return;
cout << 4 → 4
inorder (4→ right) che e' 6
t=6
inorder(6--> left) che e' NULL return
cout << 6 → 6
inorder(6--> right) che e' NULL return
cout << 7 → 7
inorder (7--> right) che e' 10
t = 10
inorder (10 →left) che e' NULL return
cout << 10 ecc… → 10

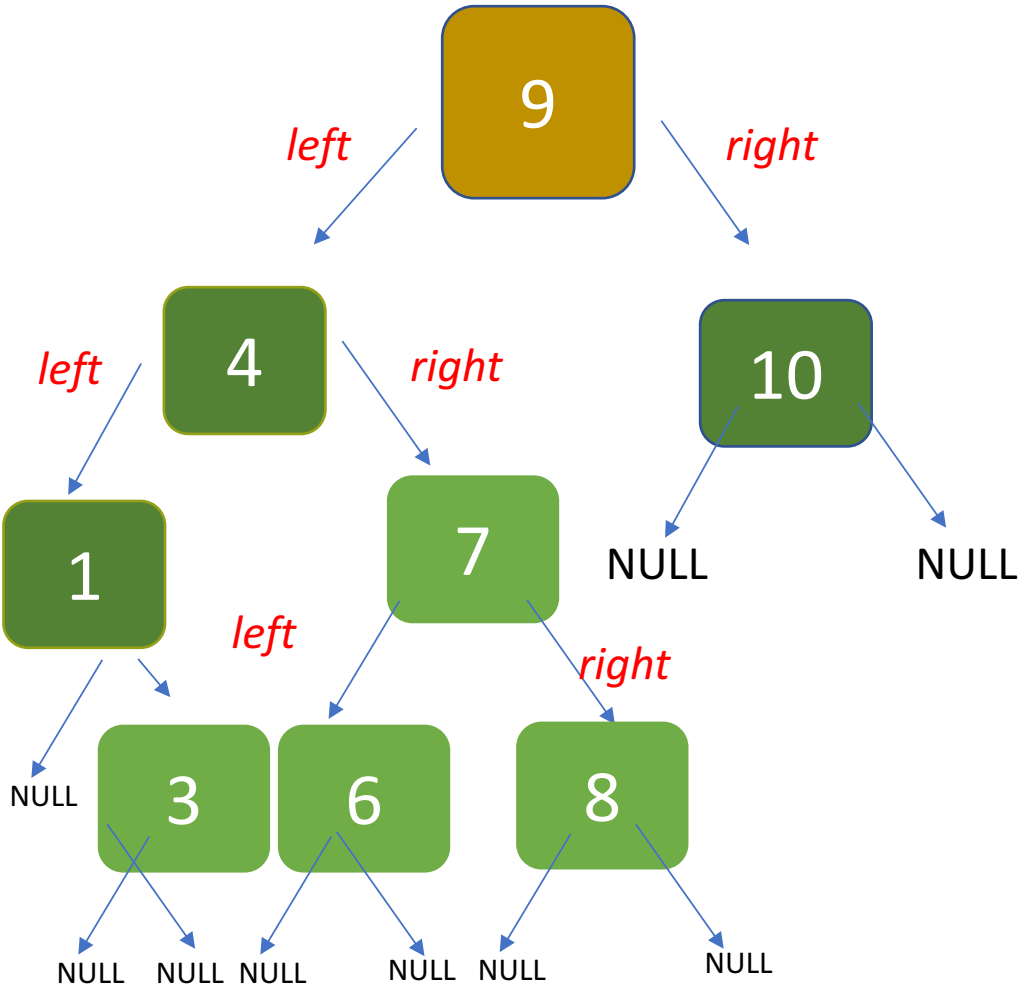
Balance

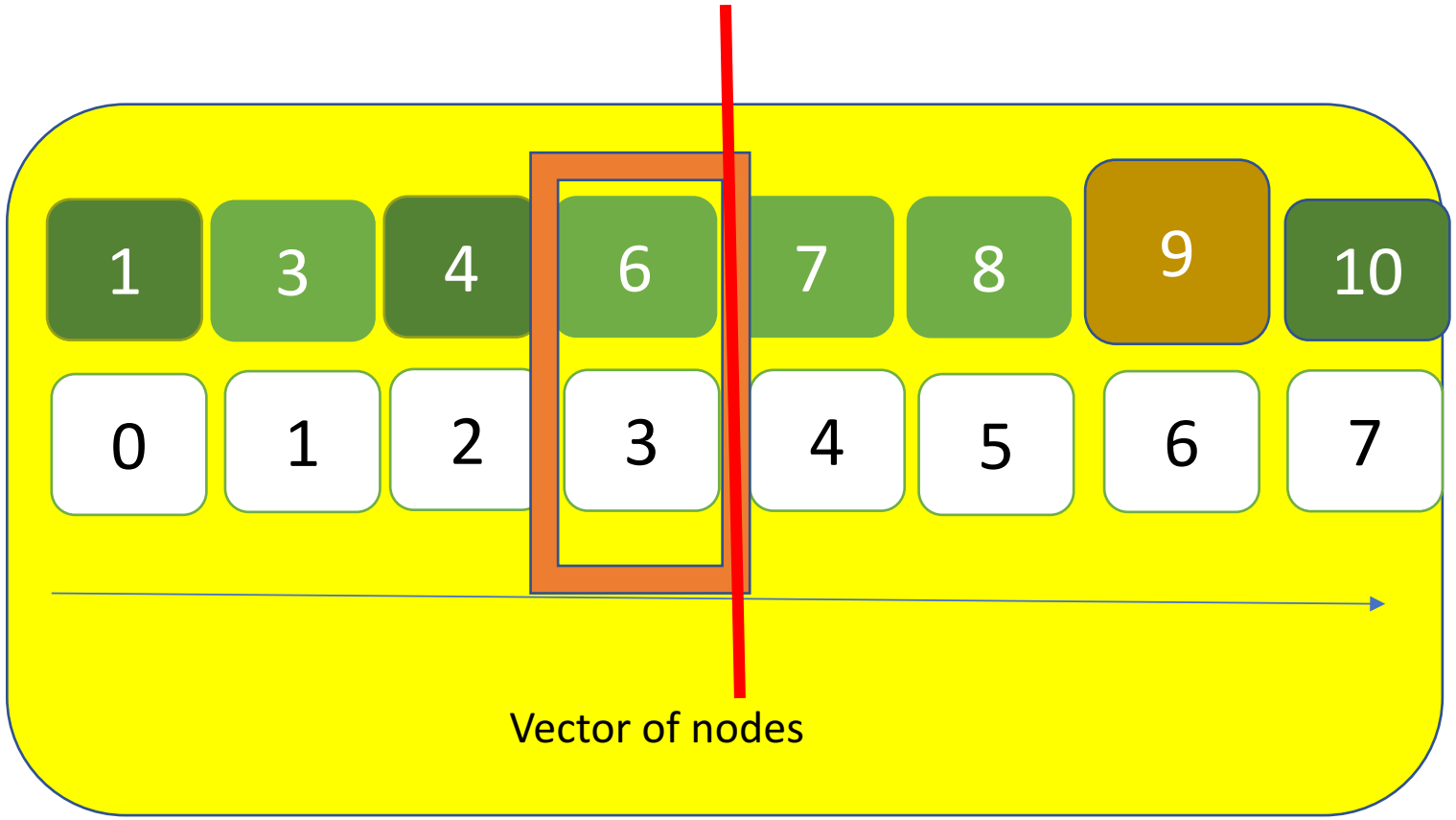
Not balanced tree





Balance

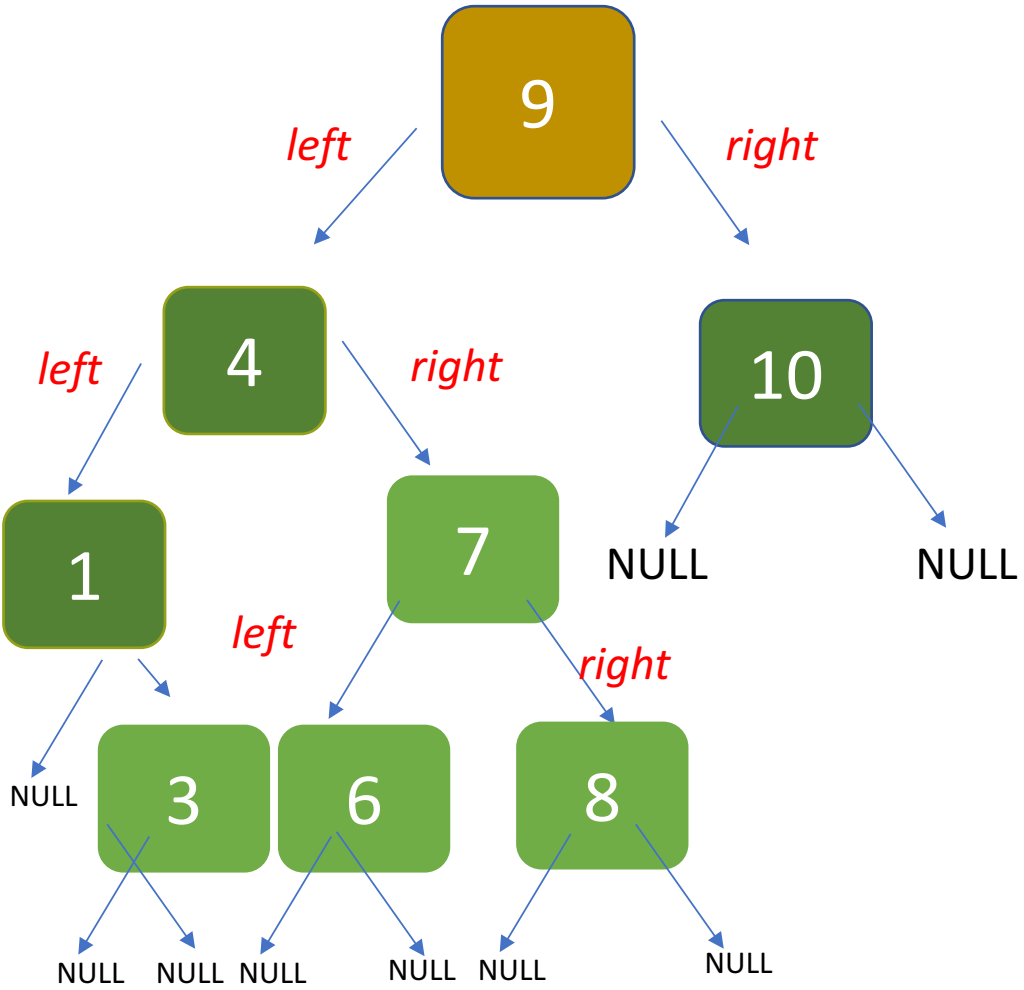


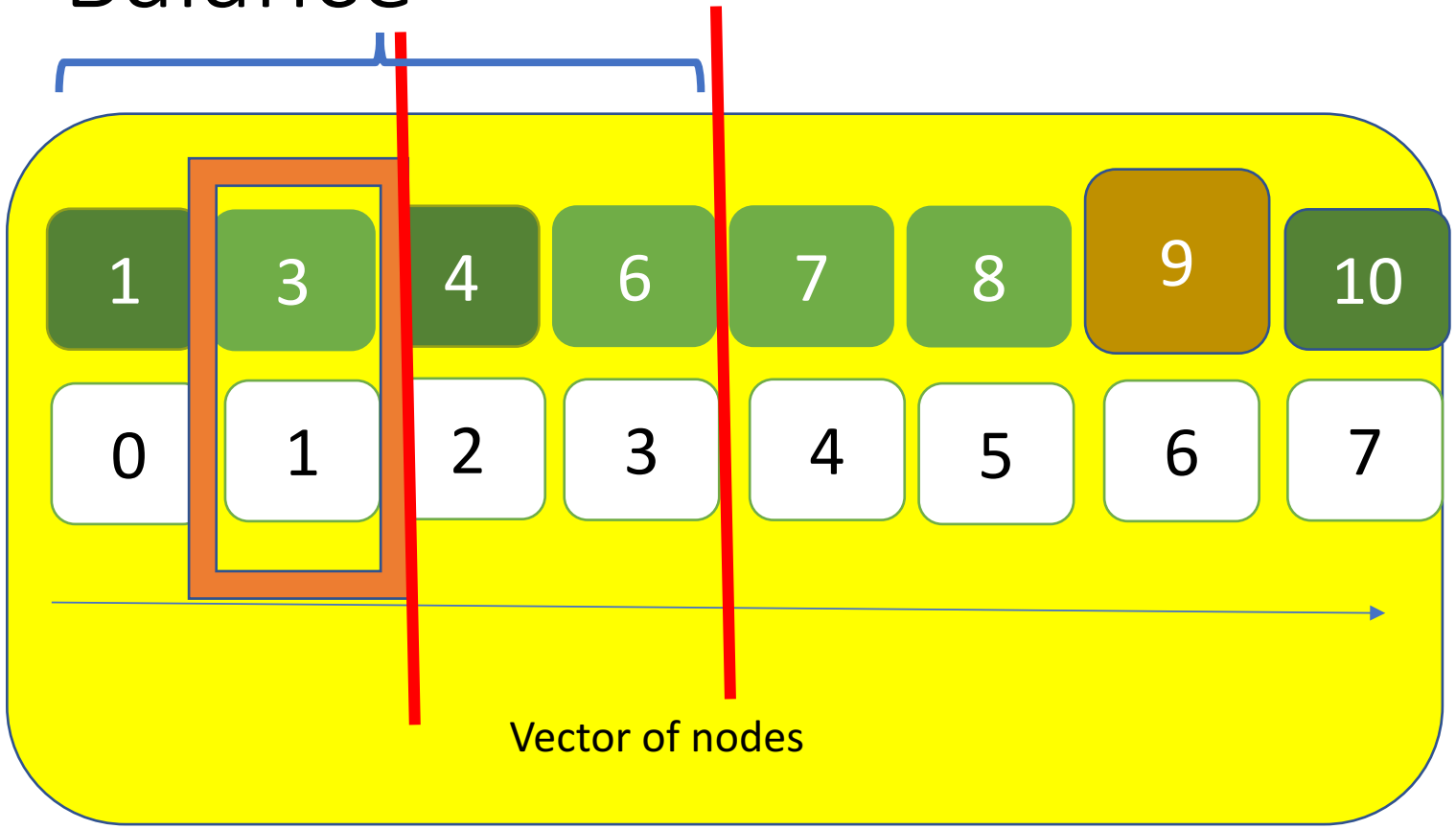


Root < - insert (v, 0, 7)
Start = 0
End = 7
Middle = (0 + 7)/ 2 = 3
t<- v[3]=6

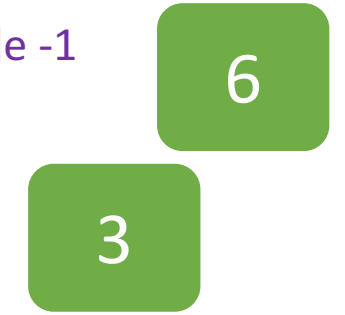
6

Balance

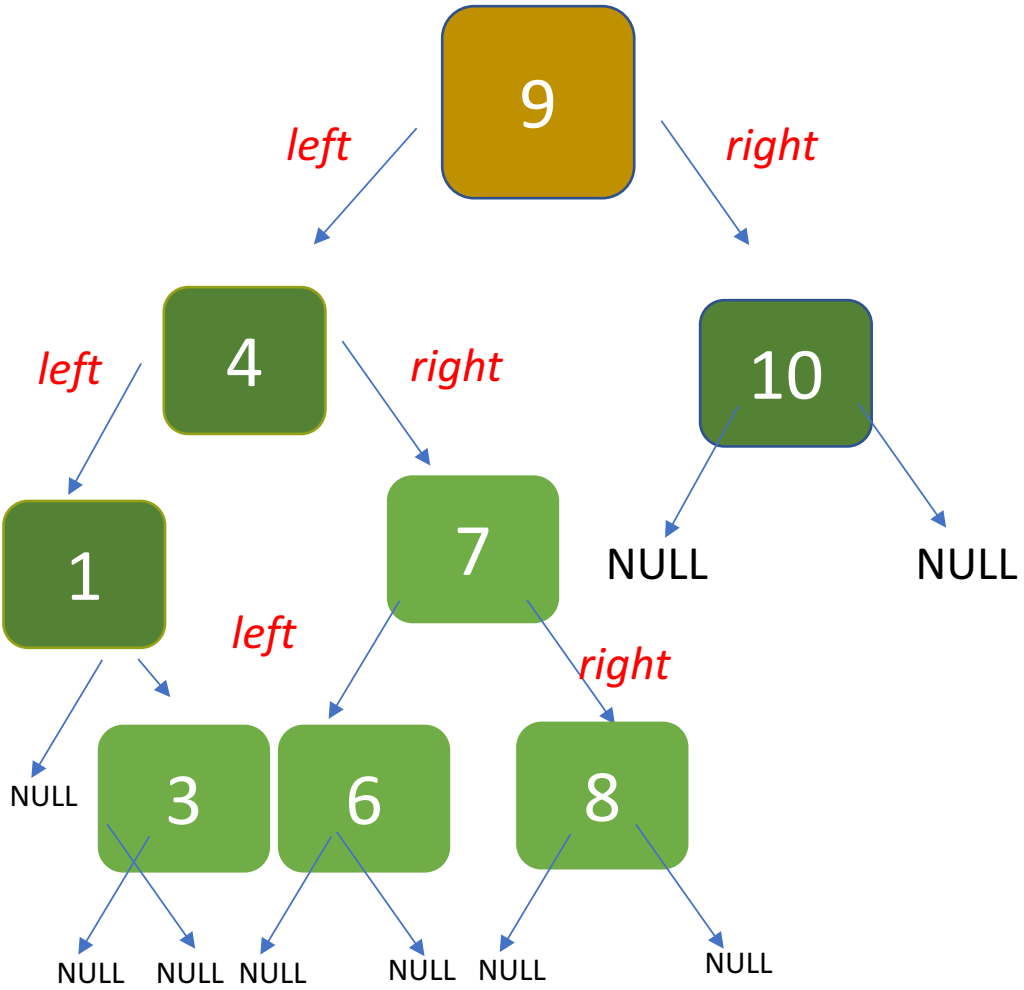


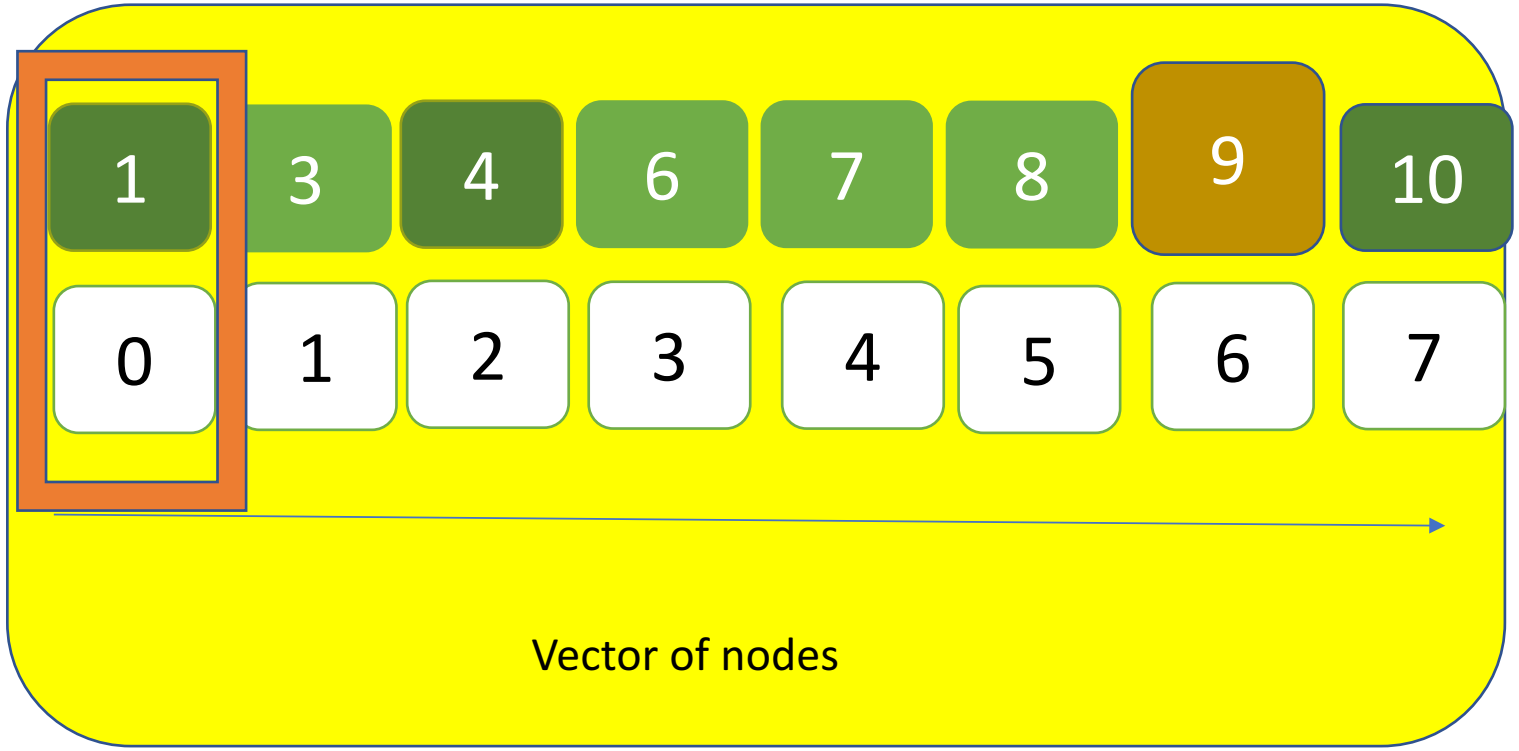


t6 <- left <- insert (v, 0,2) dove 0=start, 2=middle -1
Start = 0
End = 2
Middle = (0 + 2)/ 2 = 1
t<- v[1]=3



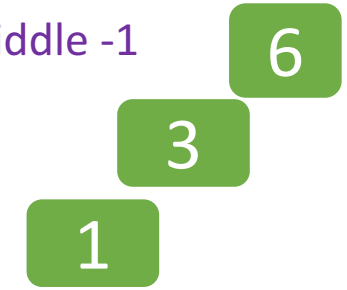
Balance



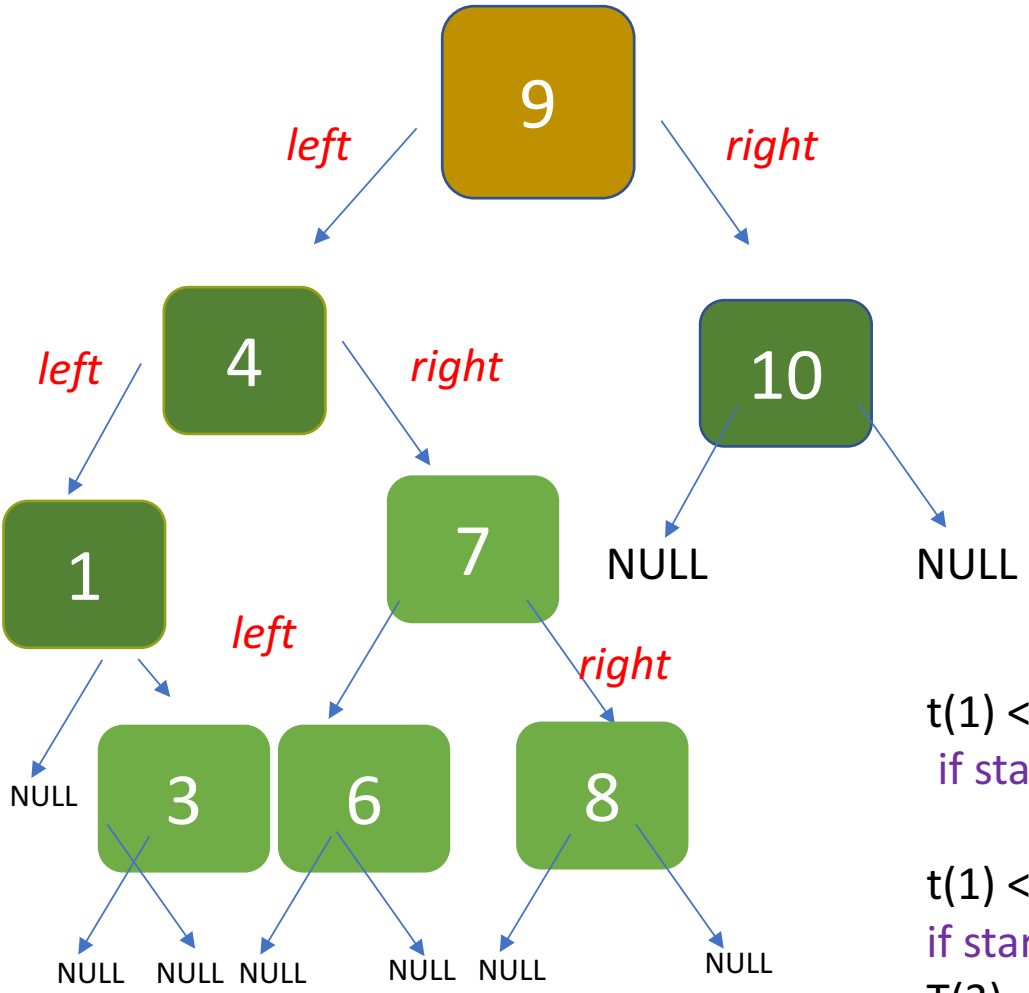


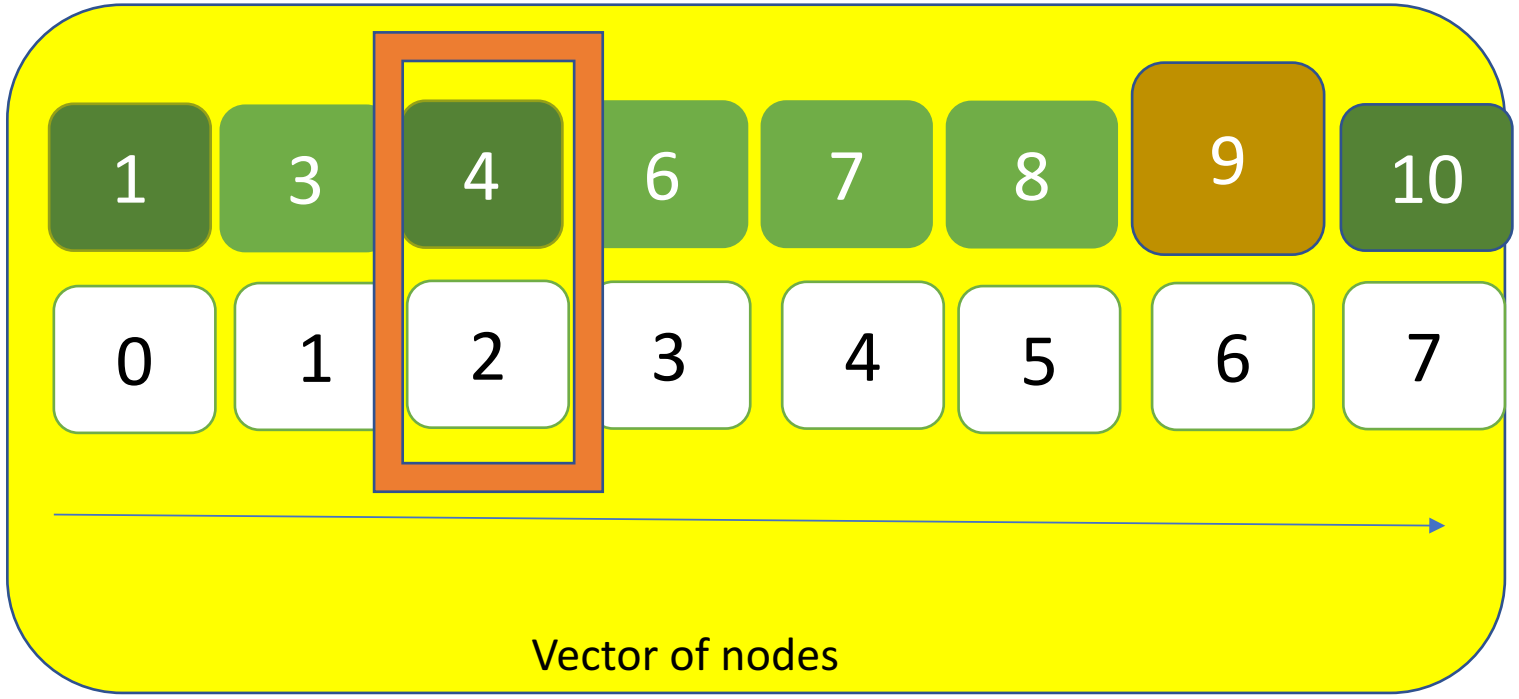
T(3) <- left <- insert (v, 0,0) dove 0=start, 0=middle -1

Start = 0
End = 0
Middle = (0 + 0)/ 2 = 0
t<- v[0]=1



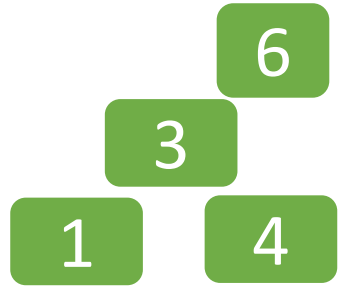
Balance



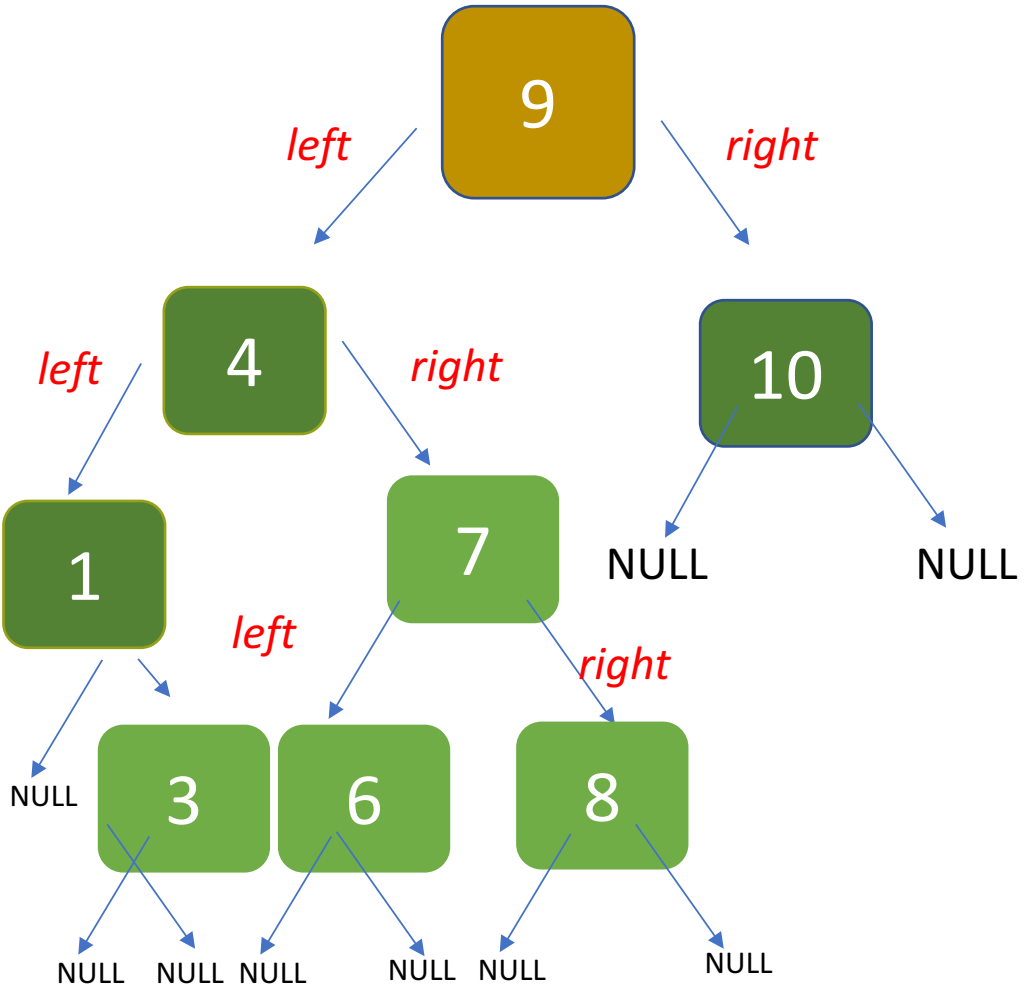


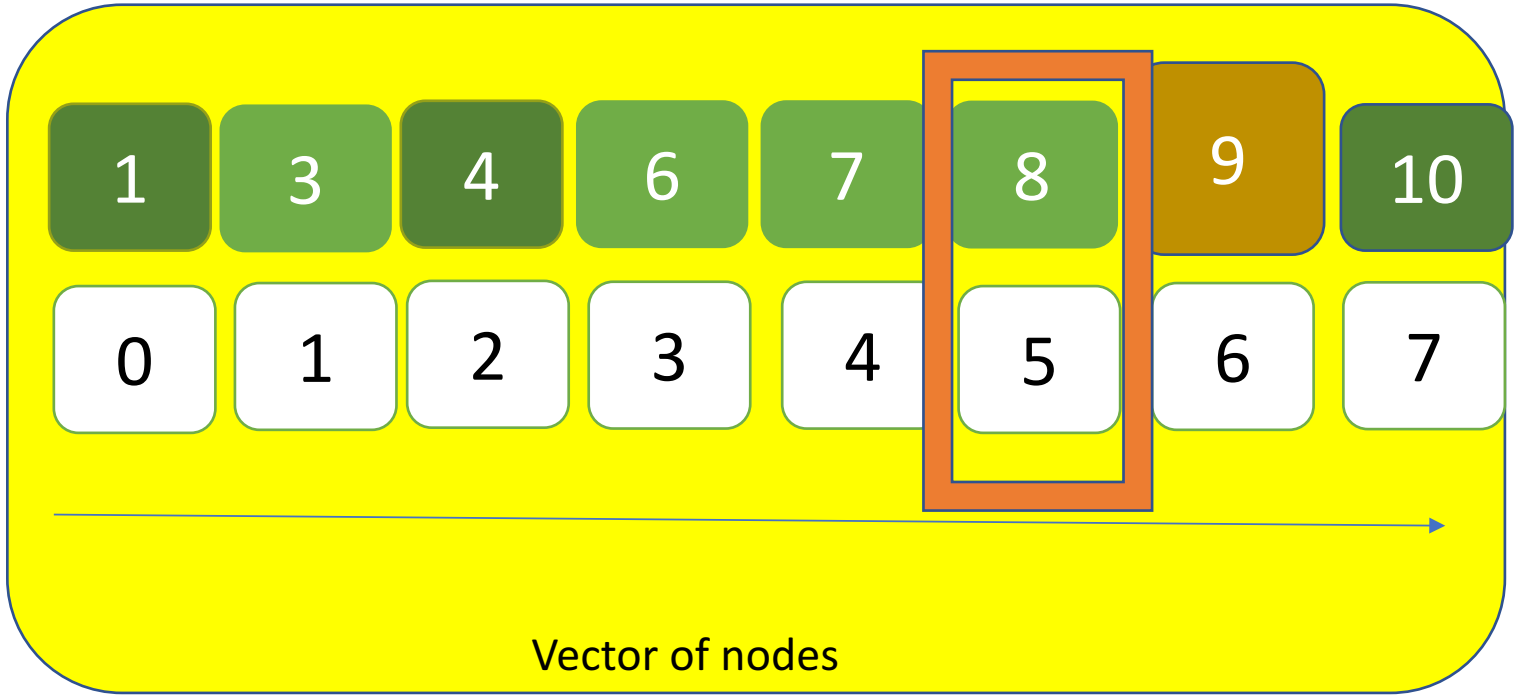
t(1) <- left <- insert (v, 0,-1) = null dove 0=start ,-1=middle -1=0 -1
if start > end RETURN return t

t(1) <-right<-insert(v,1,0) =null dove 1=middle +1, 0 =end
if start > end RETURN return t
T(3) <-right<-insert(v,2,3) Start=2 End=3 middle=2
T v[2]=4

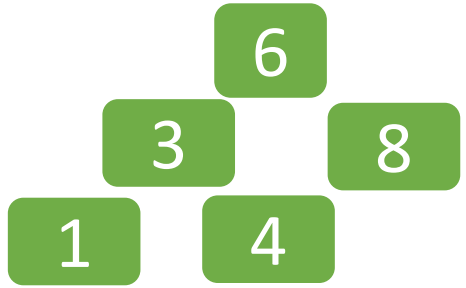


Balance

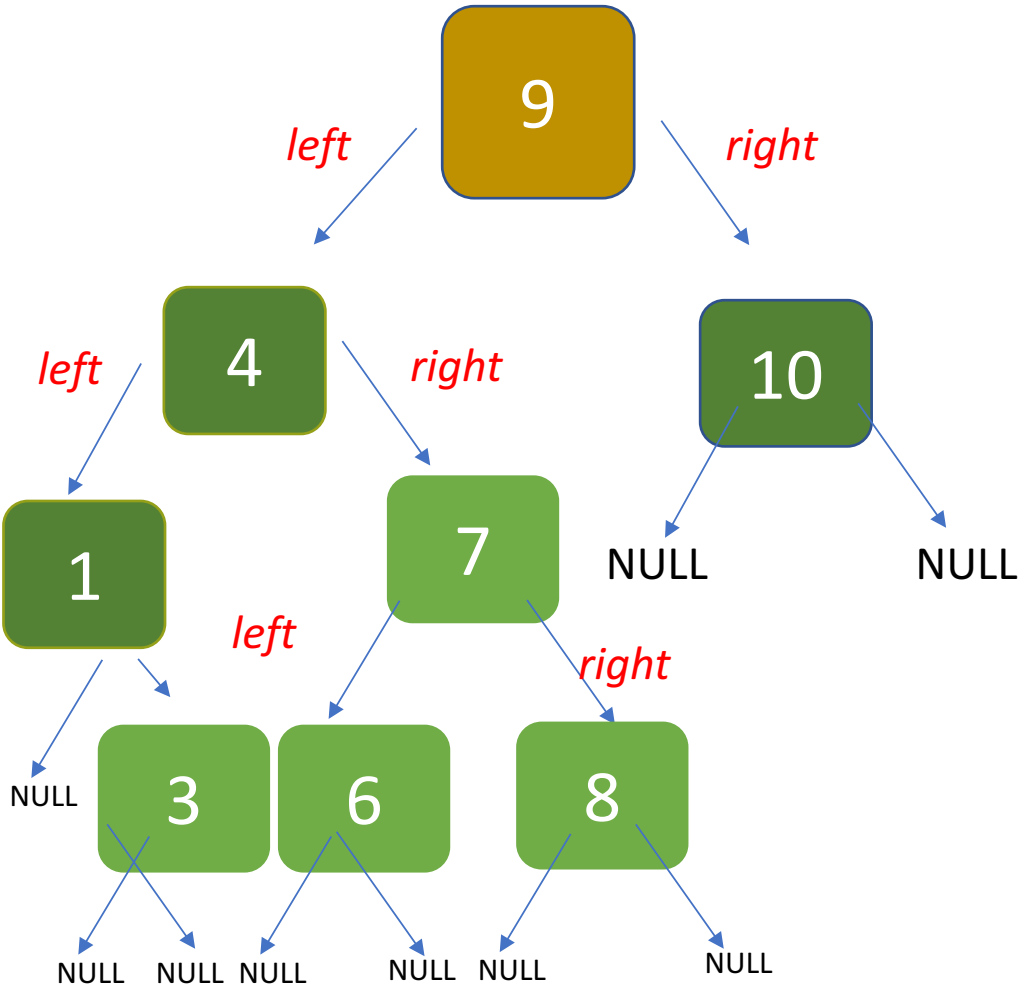


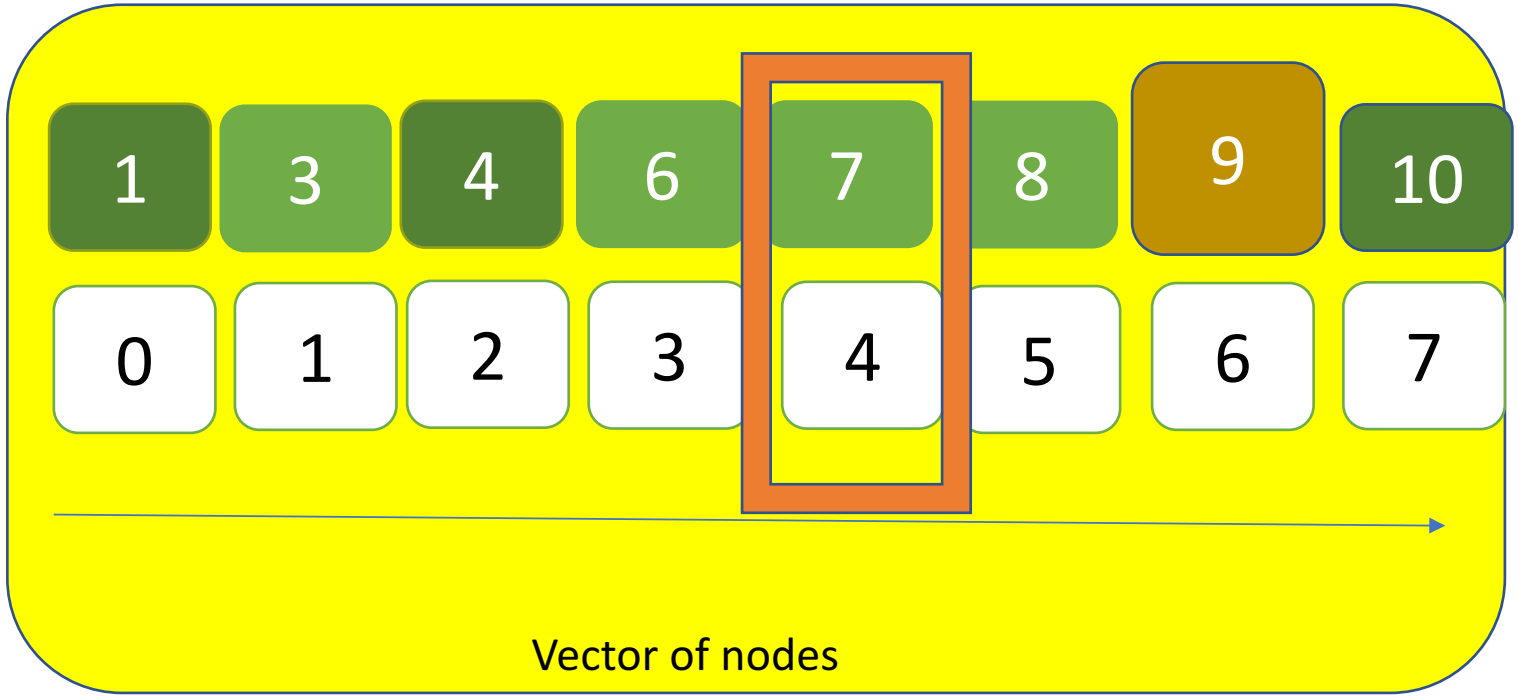


t(4) <- left = null
t(4) <-right=null
return t
T(6) <-right (v,4,7)
Start=4 end=7 middle=5
Tv[5]=8

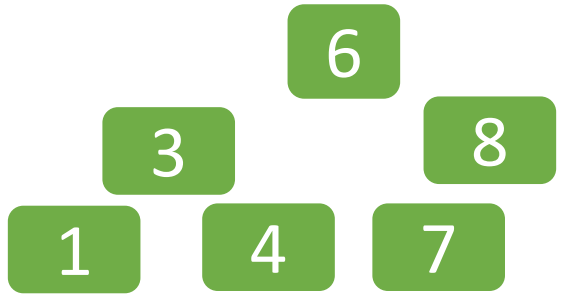


Balance

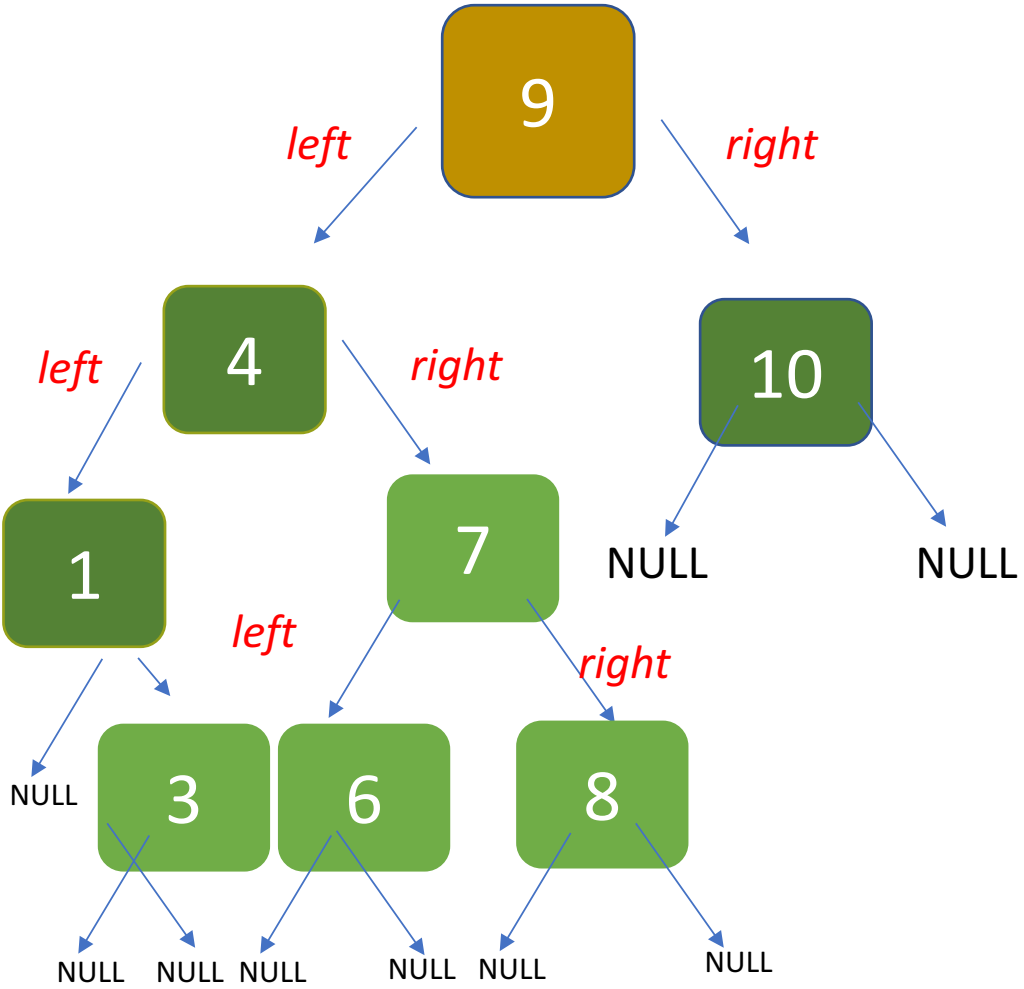


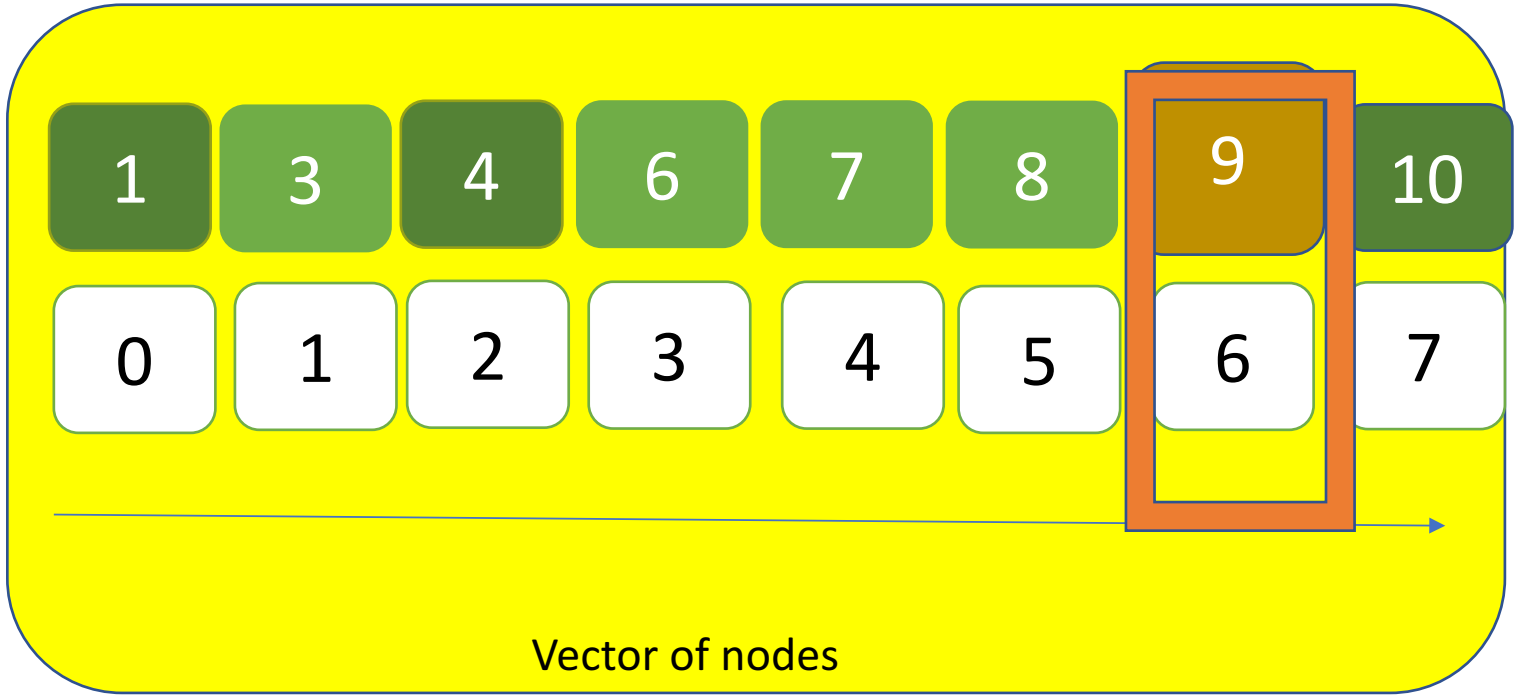


T(8) <-left (v,4,4)
Tv[4]=7
T(7) <-left null
T(7) <-right null
T(8) <-right(v,6,7)
Middle=6
tv[6]=9

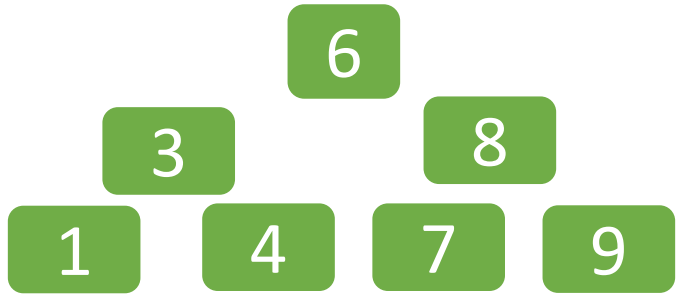


Balance

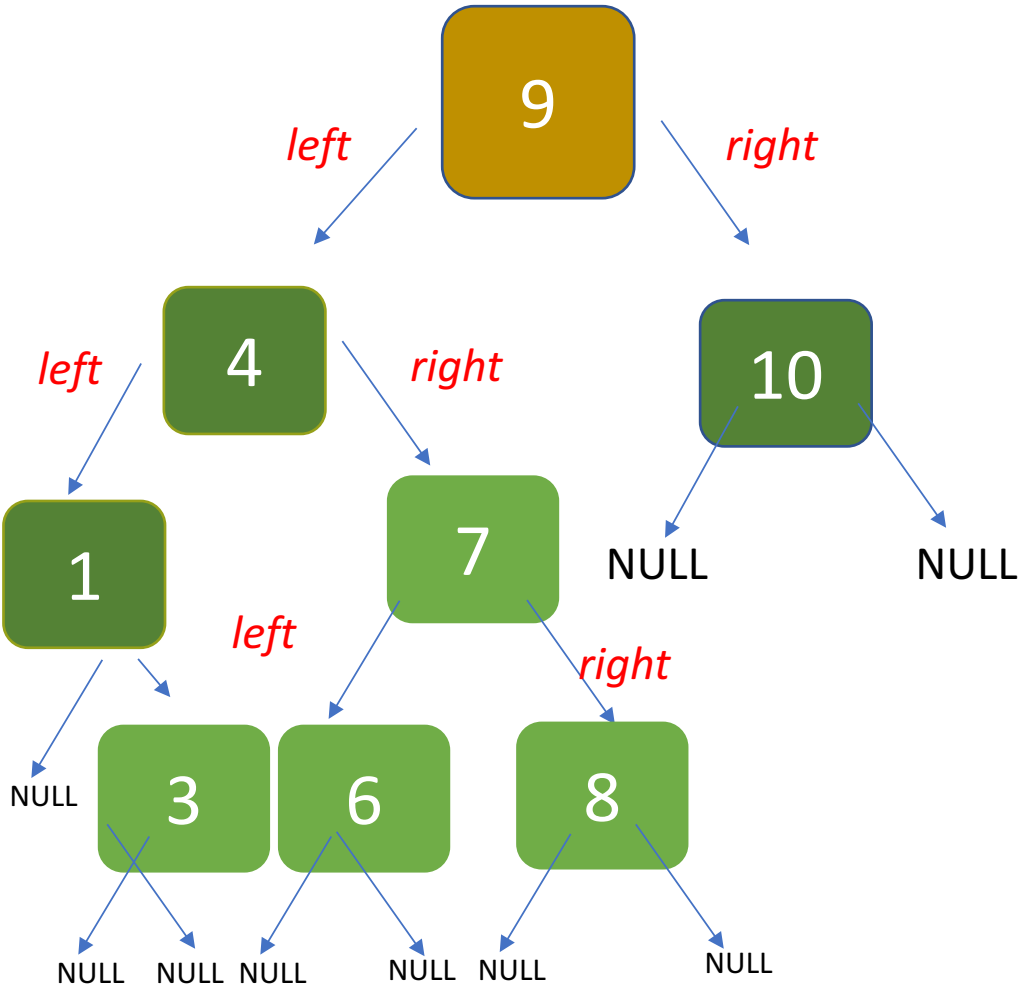


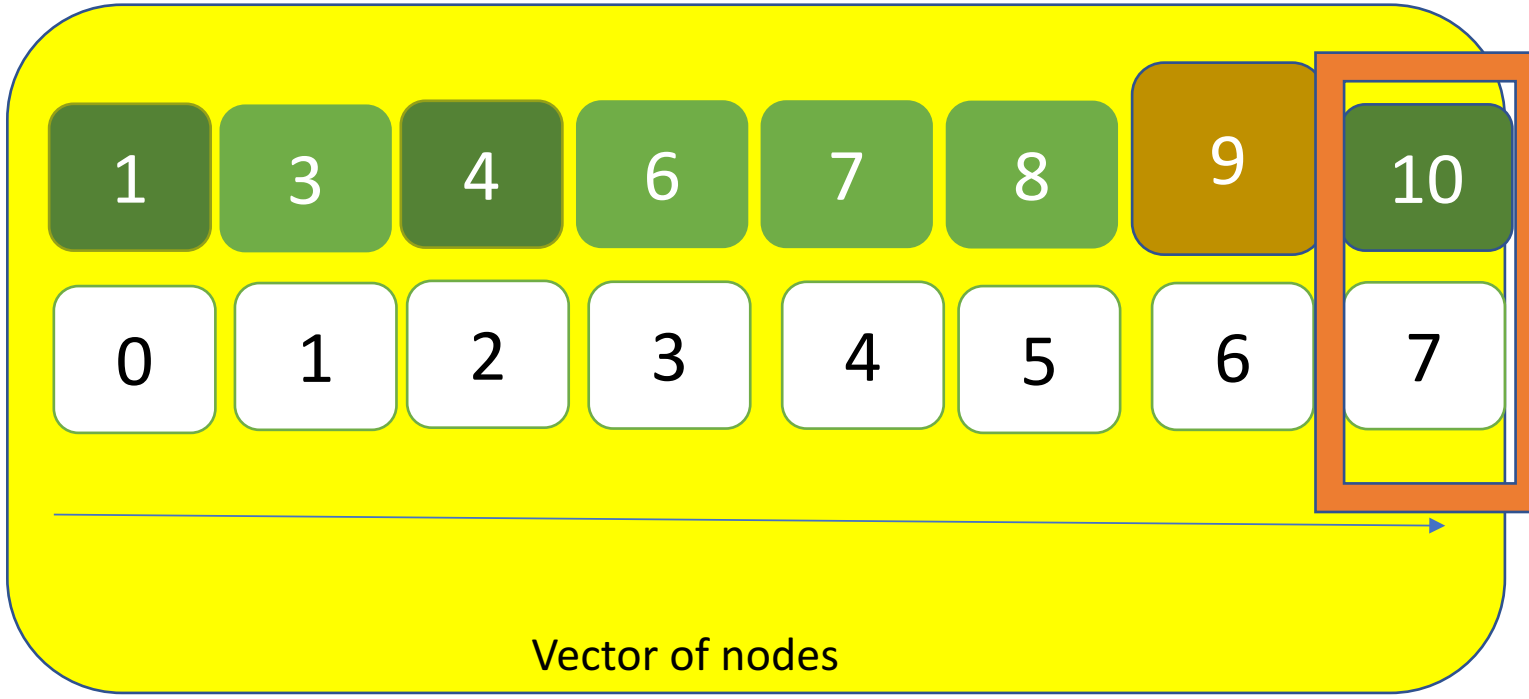


T(8) <-left (v,4,4)
Tv[4]=7
T(7) <-left null
T(7) <-right null
T(8) <-right(v,6,7)
Middle=6
tv[6]=9



Balance





T(9) <-left null
T(9) <-right(v,7,7)
Middle=7
tv[7]=10