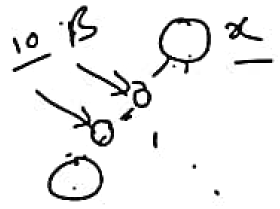
Red Black Trees

Sunday, December 4, 2022   2:10 PM

MachineLearning > MTechClasses

○ R   ○ B



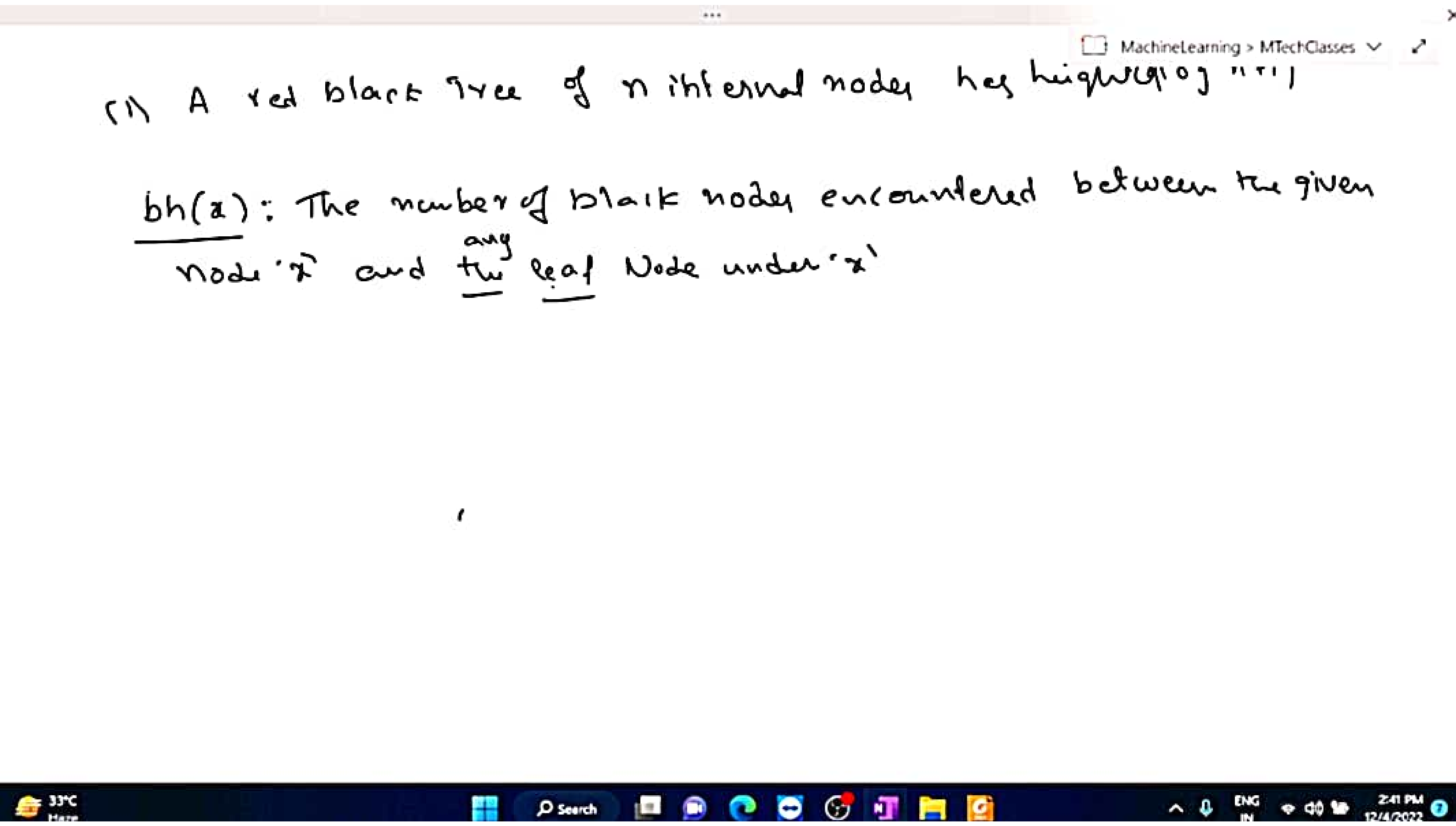
(1) Every node is either Red or Black

(2) The root is Black

(3) Every <u>leaf</u> is Black

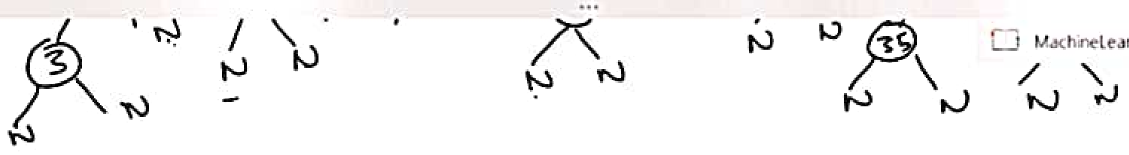
(4) If a node is Red both its children are black

(5) for each node all simple paths from that node to descendant leaves contain the same number of black node



(1) A red black Tree of n internal nodes has heig[illegible]log (n+1)

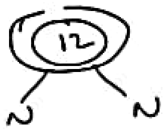
bh(x): The number of black nodes encountered between the given node 'x' and $\overset{\text{any}}{\underline{\text{the}}}$ <u>leaf</u> Node under 'x'



(1) A red black Tree of n internal nodes has height $O(\log n+1)$

bh(x): The number of black nodes encountered between the given node 'x' and ~~the~~ any leaf Node under 'x'

If $bh(x)$ is h then there are atleast $2^{bh(h)}-1$ internal nodes in such R-B Tree

 $\rightarrow 2^{bh(x)}-1$

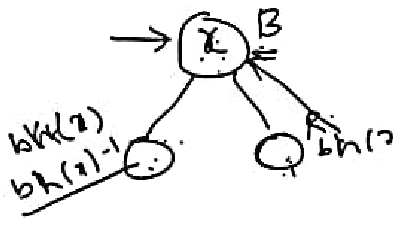
If $bh(x)$ is h then there are atleast $\underline{2^{bh(h)}-1}$ internal nodes in such R-B Tree

$\rightarrow 2^{bh(x)} - 1$

$\underline{2^0 - 1} : \underline{0}$

$\underline{1} > 0$

$bh(x) = \underline{h}$

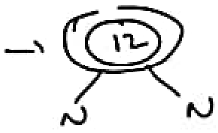


$\geq 2(\underline{2^{bh(x)-1} - 1}) - 1$

induction step

MachineLearning > MTechClasses

$\nearrow$ in such R-B Tree

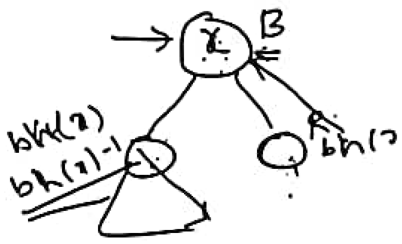


$\rightarrow 2^{bh(x)} - 1$

$\underline{2^0 - 1} : \underline{0}$

$\underline{1 > 0}$

$bh(\underline{x}) = \underline{h}$

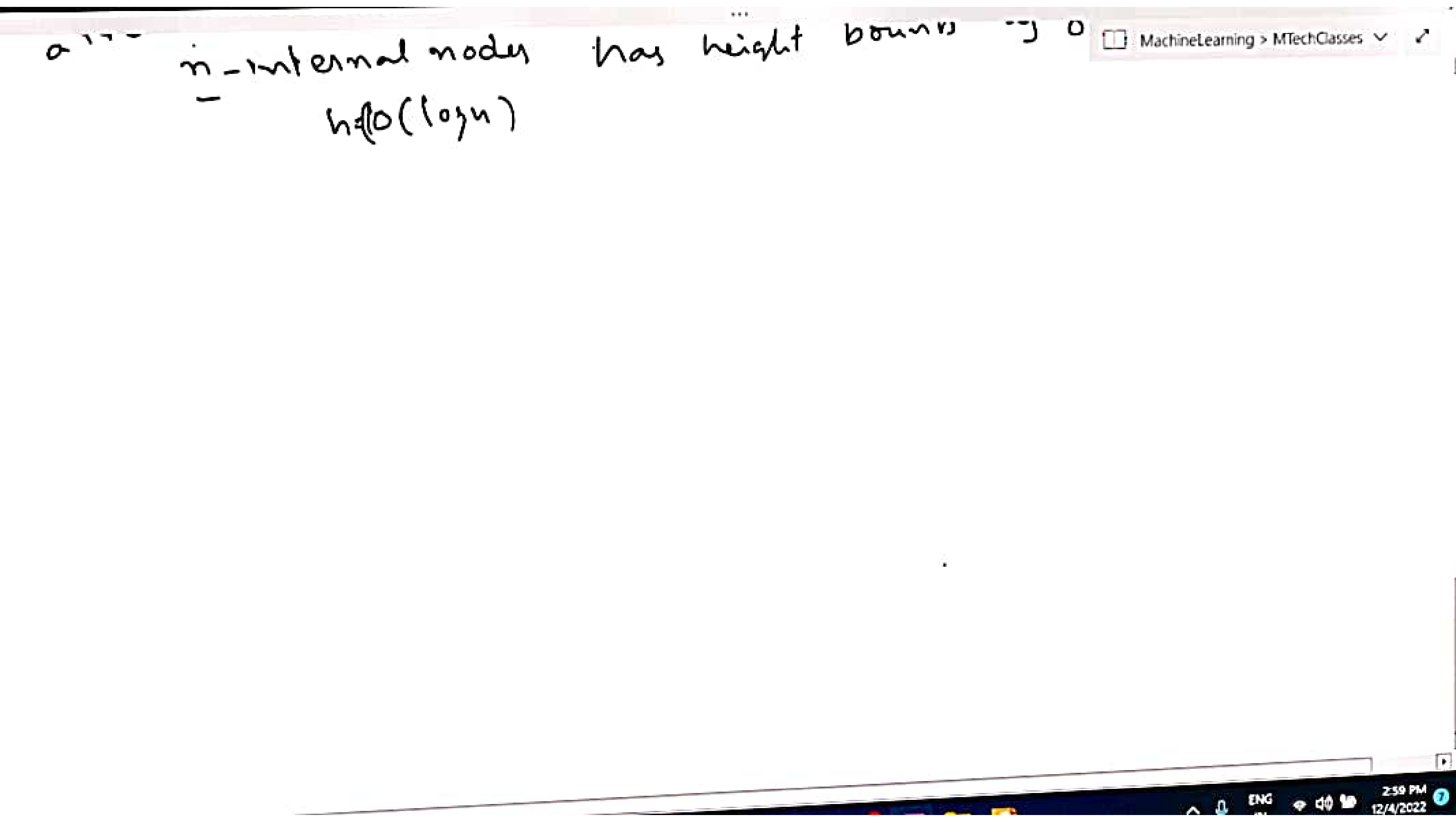


$b \geq 2(\underbrace{2^{bh(x)-1} - 1}_{\text{induction step}}) + 1 : \underline{2^{bh(x)} - 1}$

MachineLearning > MTechClasses

The ~~b~~number of internal nodes of a R.B.T with bh(T) is atleast $\geq 2^{bh(T)} - 1$

a tree with n-internal nodes has height upper bound of $O(\log n)$



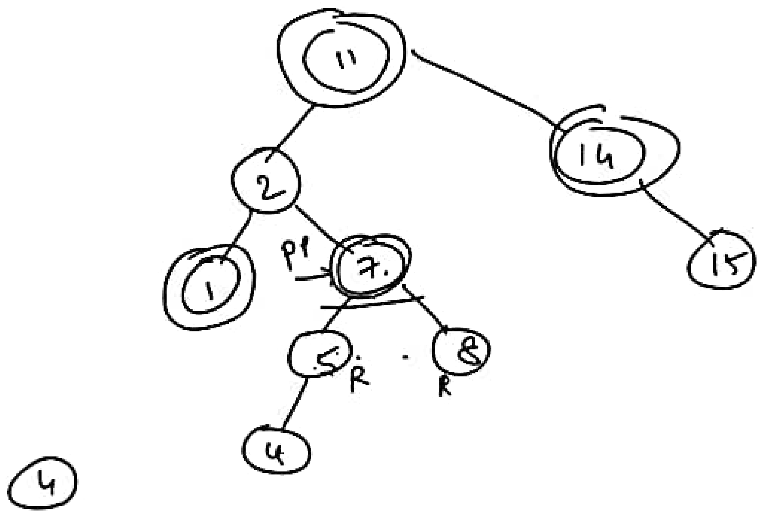
a tre n-internal nodes has height bounds of O

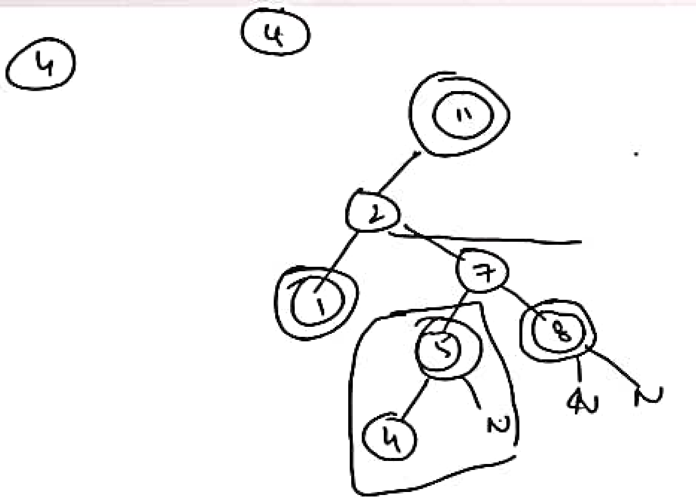
$h = O(\log n)$

-

$h = O(\log n)$

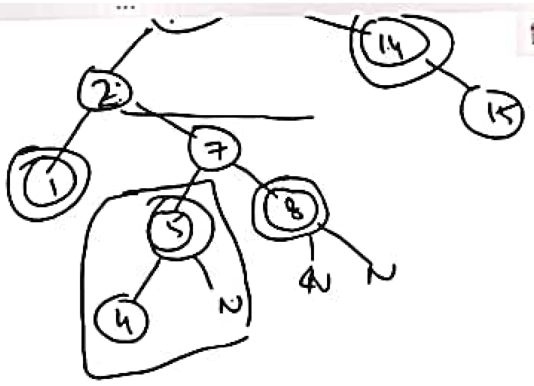
MachineLearning > MTechClasses

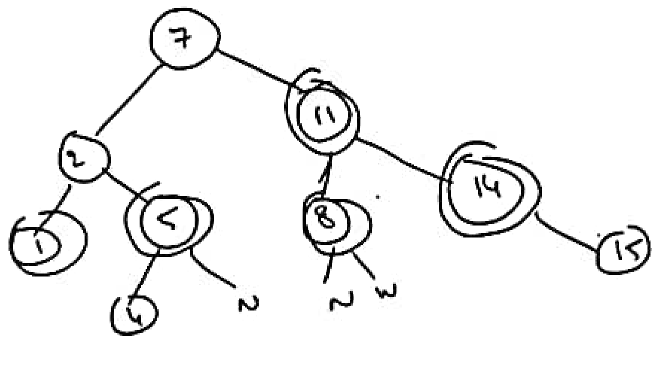


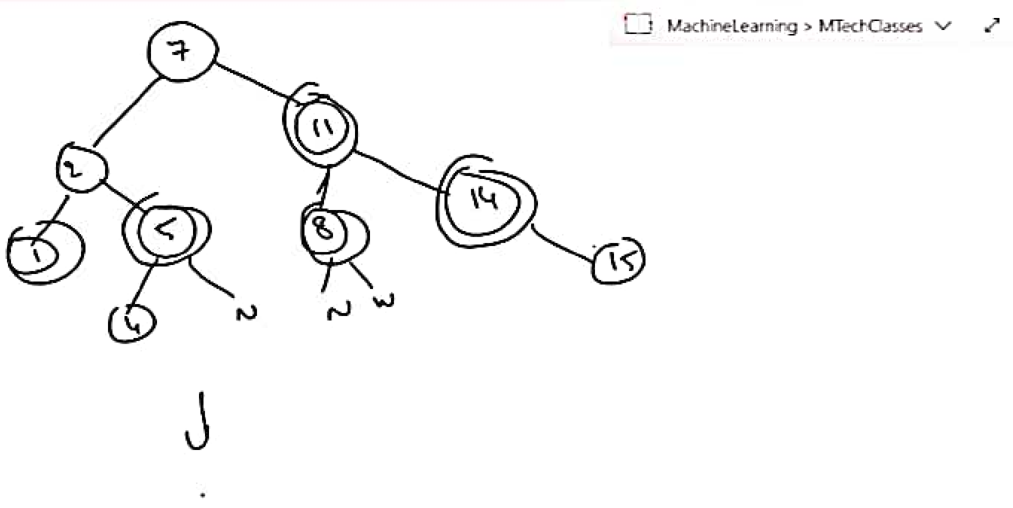


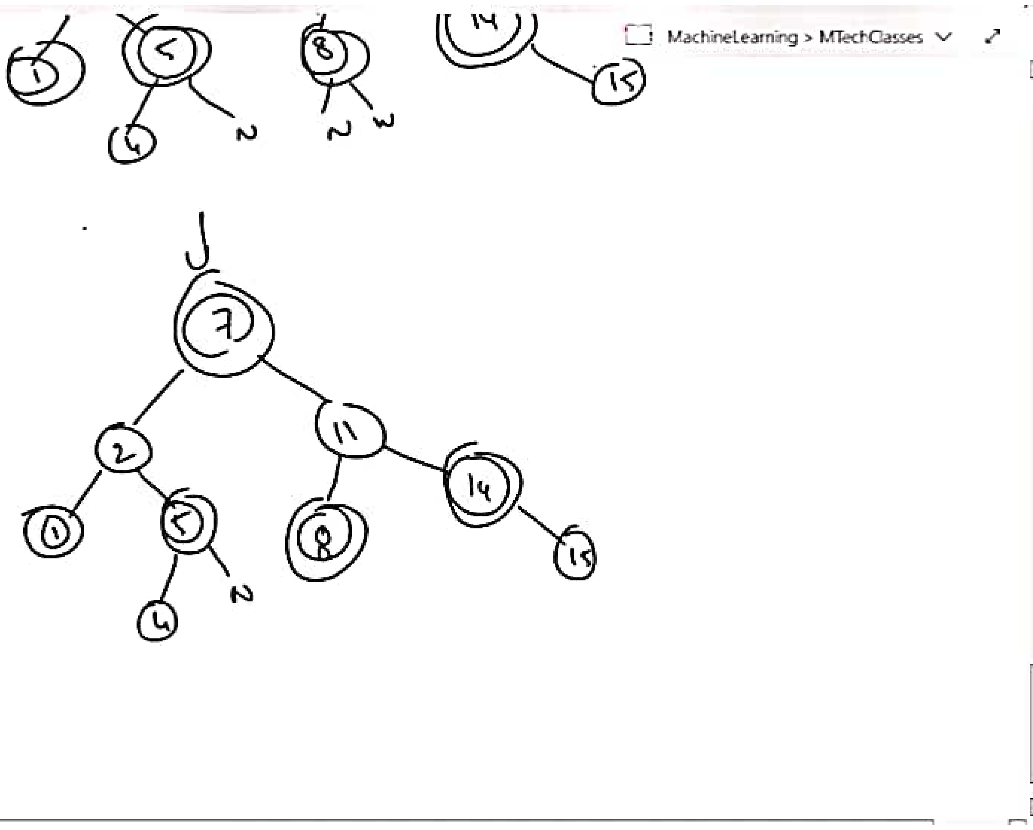
MachineLearning > MTechClasses

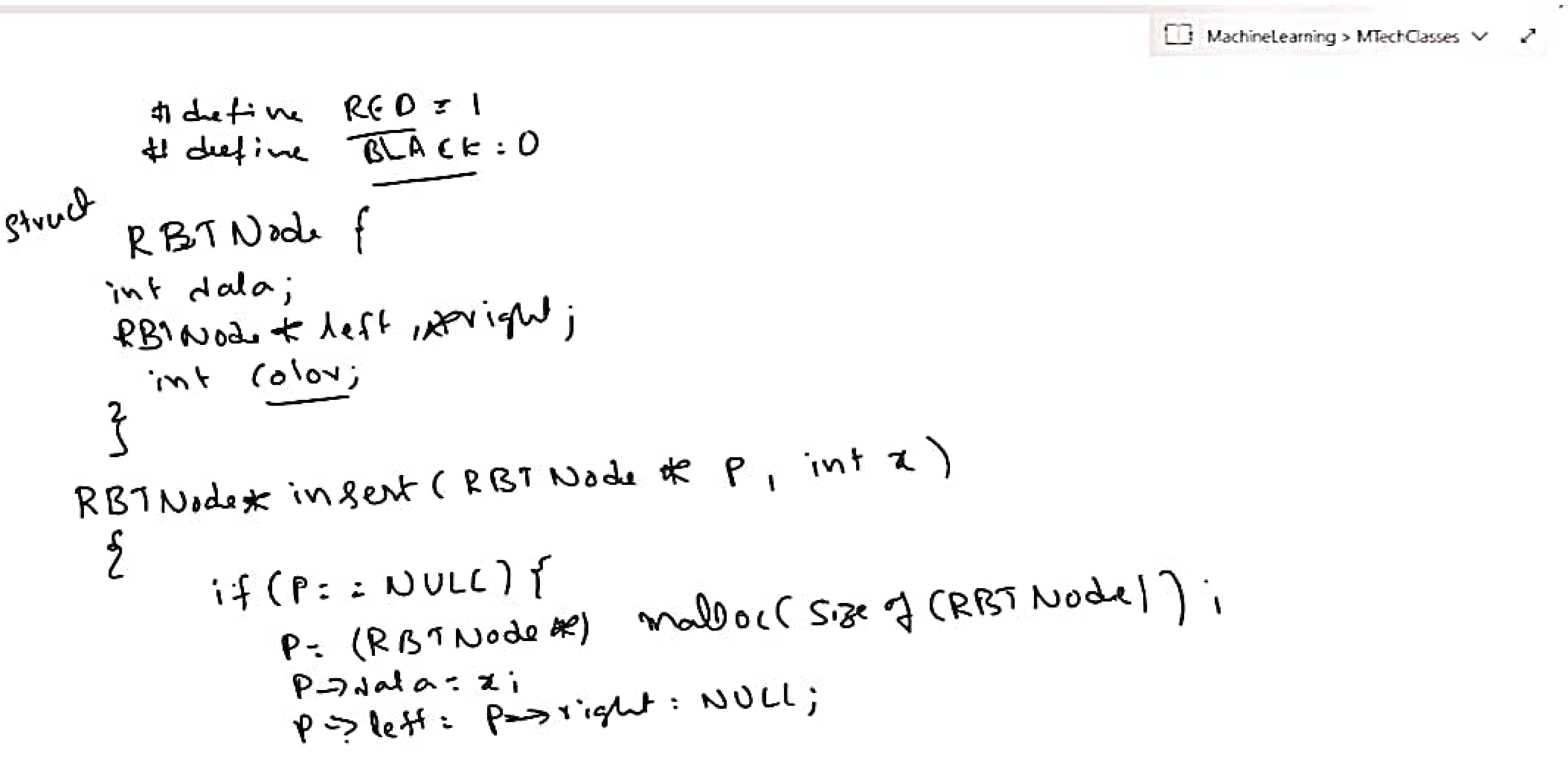
MachineLearning > MTechClasses





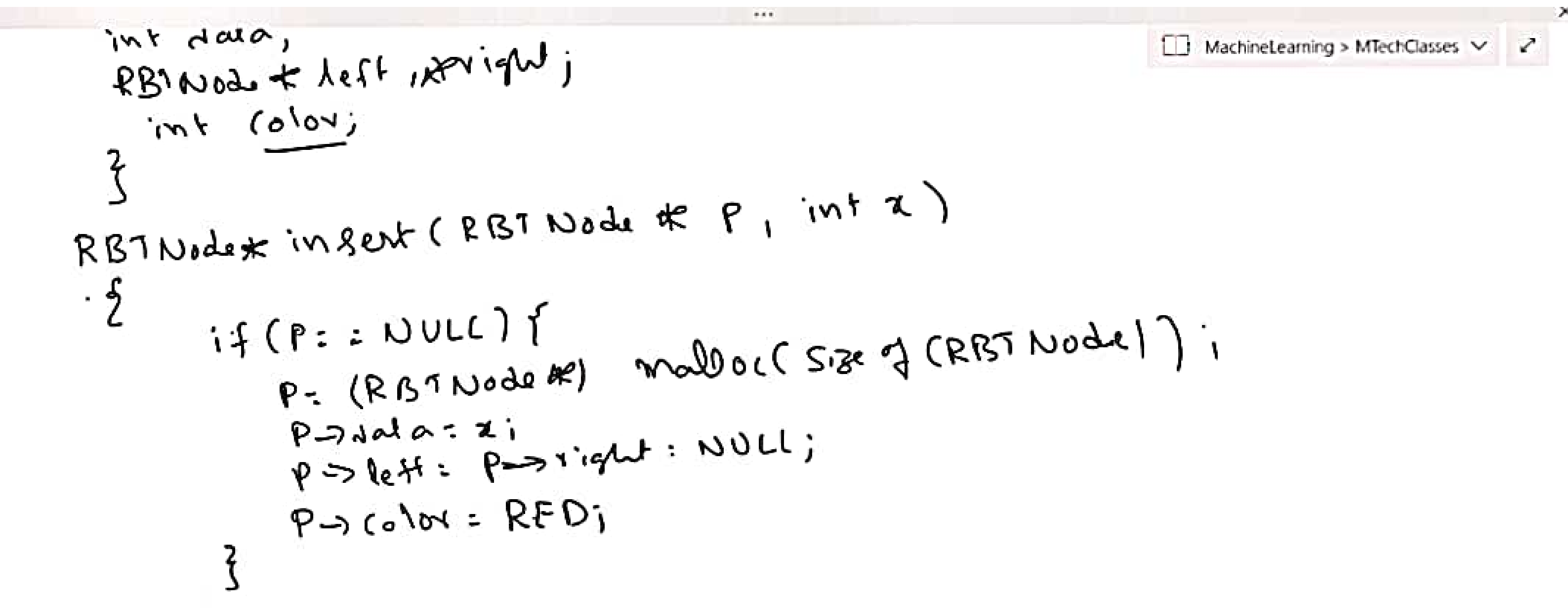

MachineLearning > MTechClasses
7
2
11
1
5
8
14
4
2
2
3
15


MachineLearning > MTechClasses

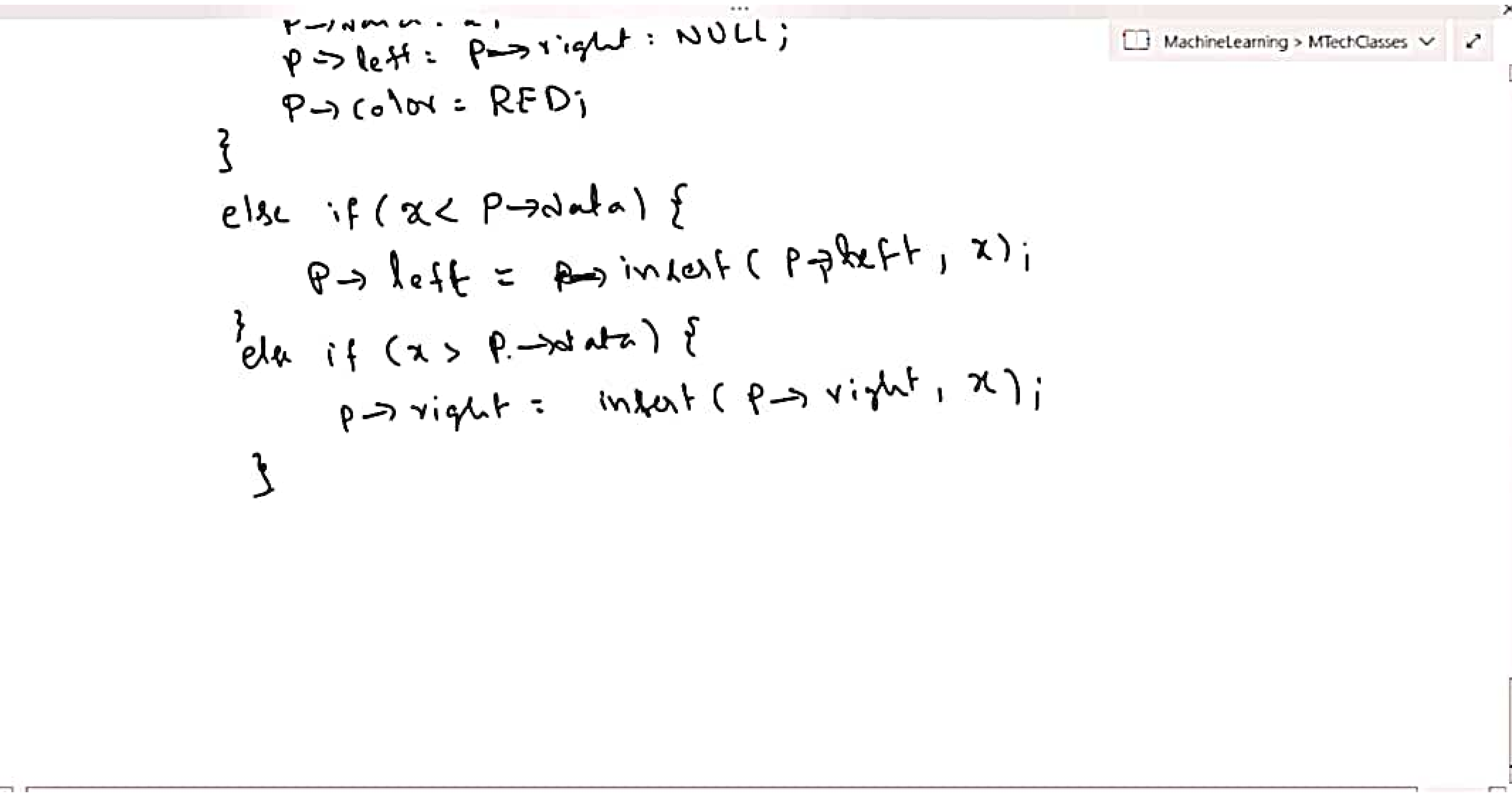


MachineLearning > MTechClasses

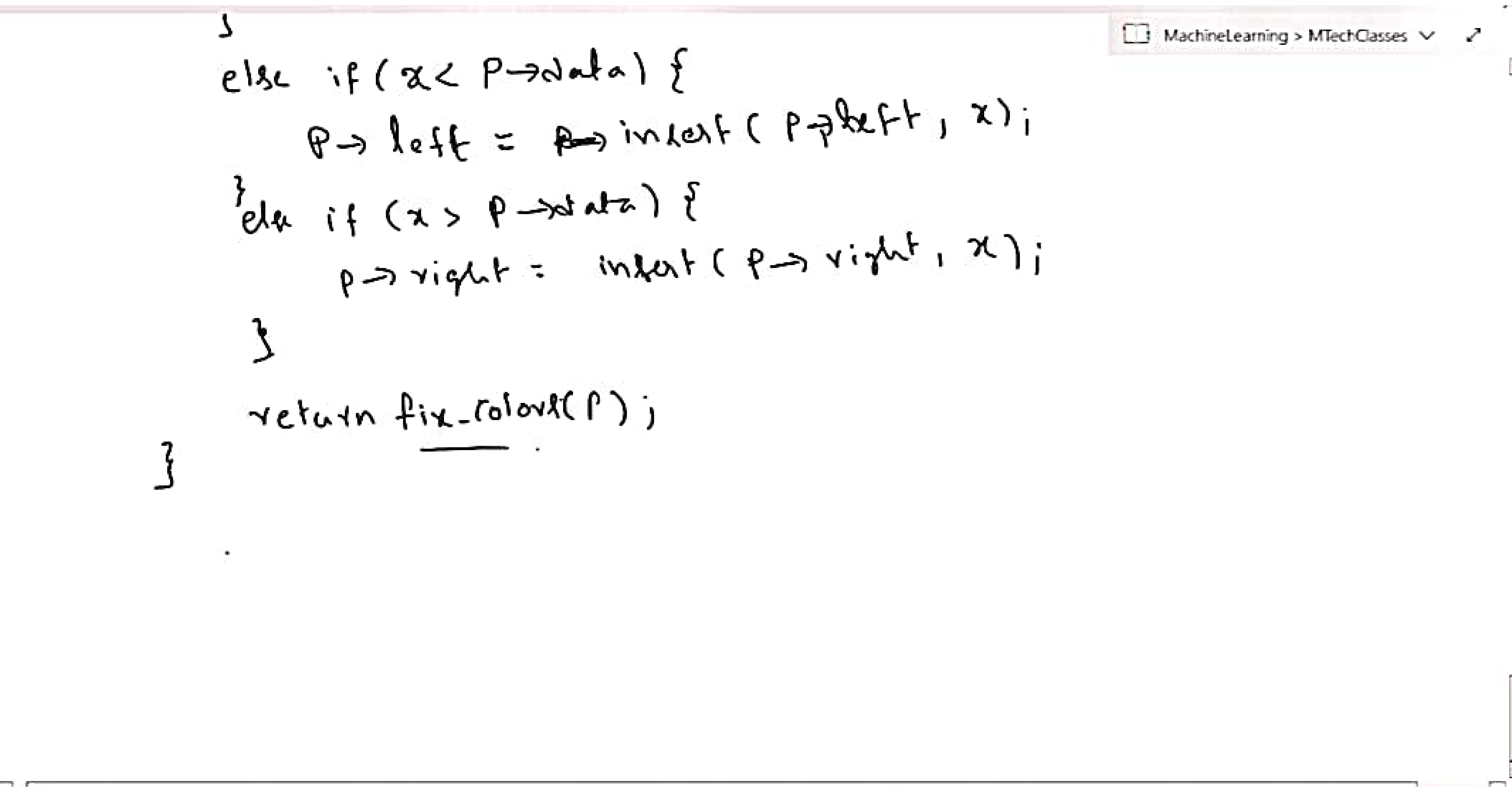
```
   # define  RED = 1
   # define  BLACK = 0
struct
   RBTNode {
  int data;
  RBTNode * left, *right;
    int color;
  }
 RBTNode* insert ( RBTNode * P, int x )
  {
        if (P = = NULL) {
           P = (RBTNode *)  malloc( size of (RBTNode) ) ;
           P->data = x;
           P -> left = P->right = NULL;
```



```
  int data,
  RBTNode * left, *right;
   int color;
  }
RBTNode* insert(RBTNode * P, int x)
 {
       if(P==NULL){
          P= (RBTNode *)  malloc(sizeof(RBTNode));
          P->data= x;
          P->left= P->right= NULL;
          P->color= RED;
       }
```



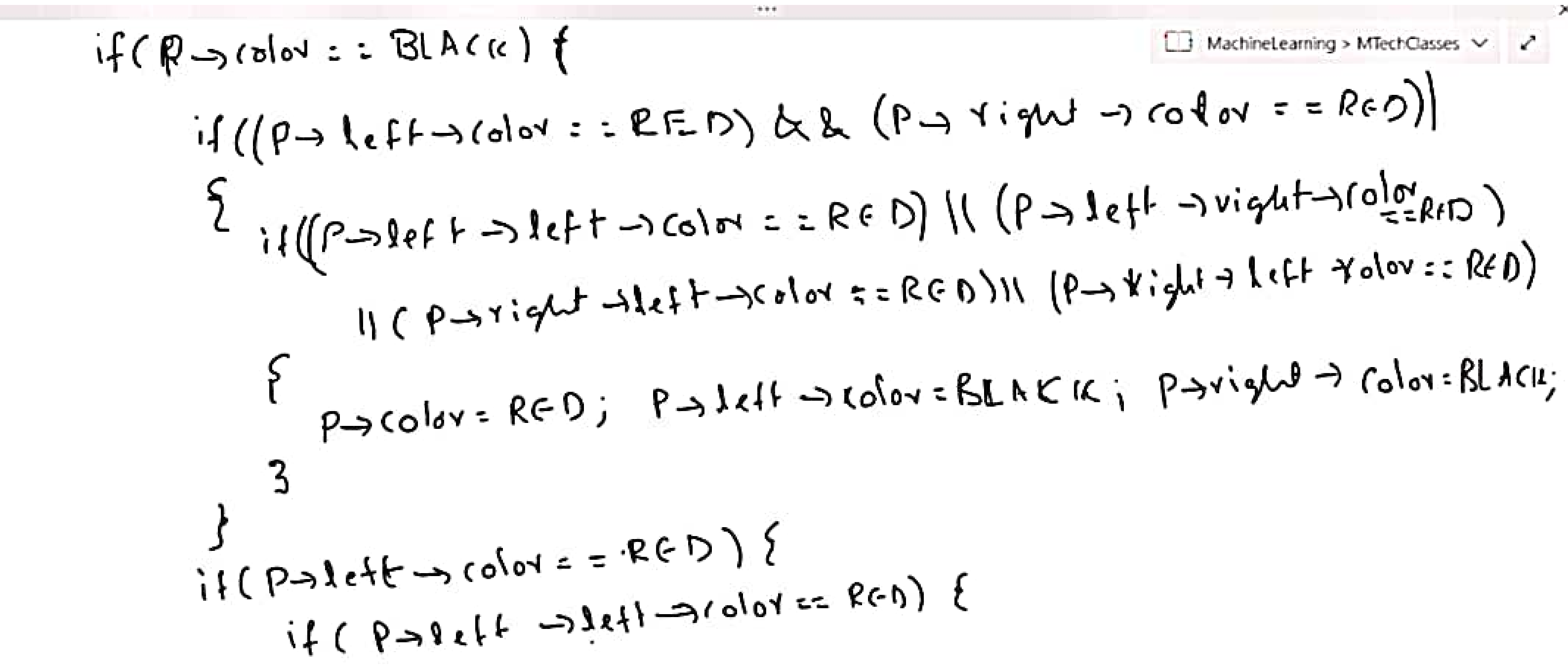
```
    [illegible]
    P->left = P->right = NULL;
    P->color = RED;
}
else if (x < P->data) {
     P->left = ~~P->~~ insert ( P->left, x);
}
else if (x > P->data) {
      P->right = insert ( P->right, x);

  }
```



```
}
else if(x< P->data) {
    P-> left = ~~P->~~ insert ( P->left , x);
}else if (x > P->data) {
    P-> right = insert ( P-> right , x);
}
return fix_colour(P);
}
```

MachineLearning > MTechClasses

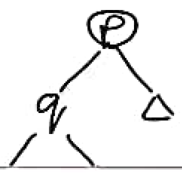
```
}
RBTNode*   fix_colors(RBTNode *p) {
    if(p->color == BLACK) {
        if((p->left->color == RED) && (p->right->color == RED))
        {
            if((p->left->left->color == RED) || (p->left->right->color == RED)
                || (p->right->left->color == RED)) || (p->right->left->color == RED))
            {
                p->color = RED;  p->left->color = BLACK;  p->right->color = BLACK;
            }
        }.
```

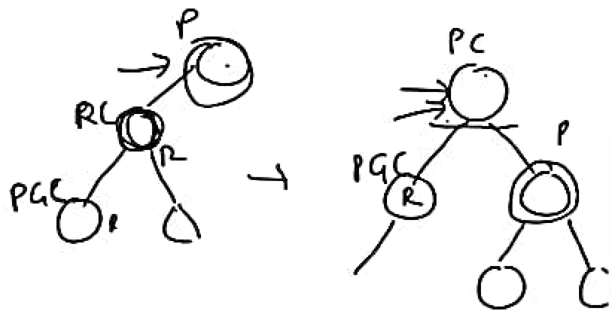


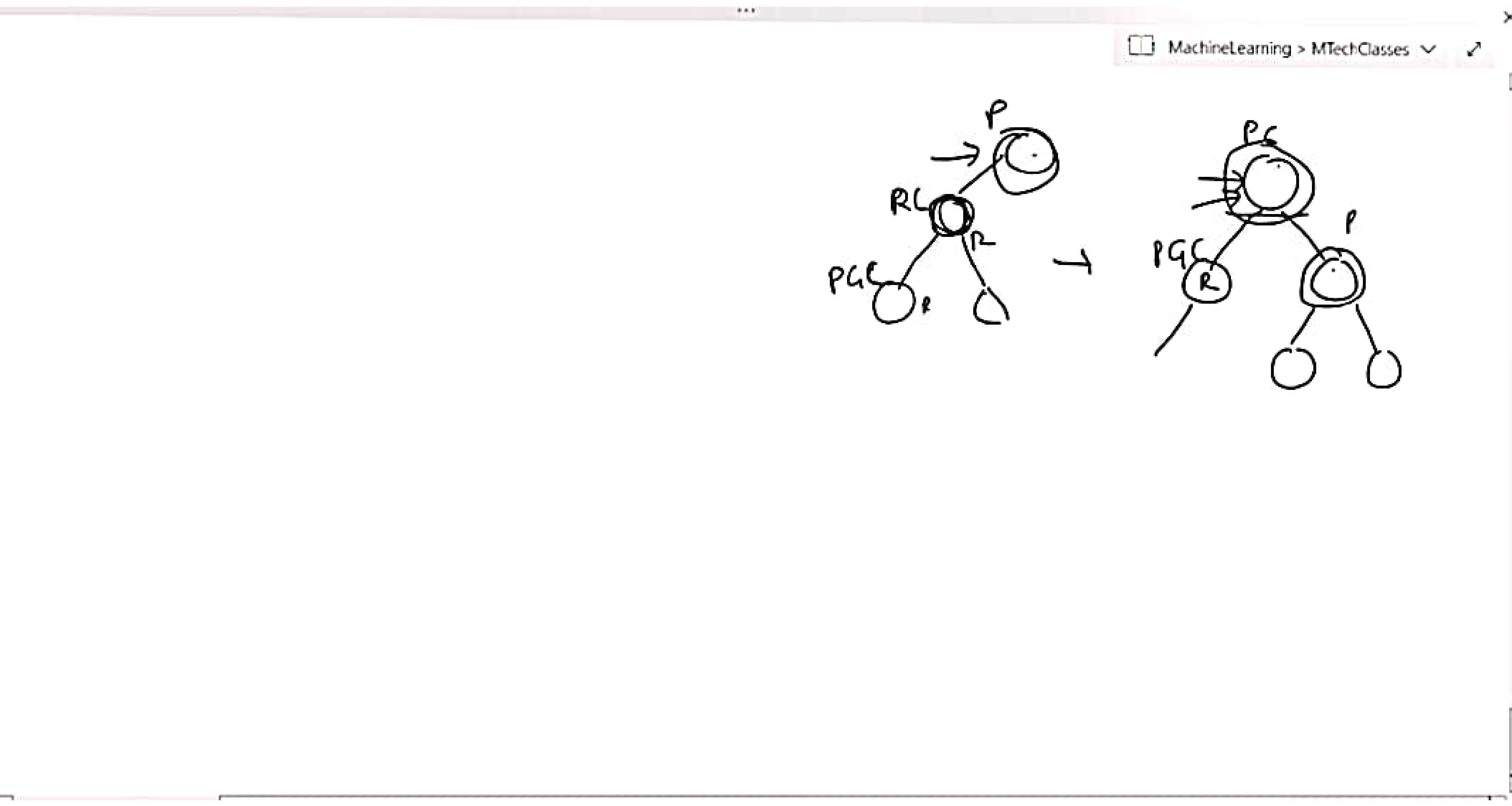
```
if(P→color == BLACK) {
    if((P→left→color == RED) && (P→right→color == RED))
    {
       if((P→left→left→color == RED) || (P→left→right→color == RED)
            || (P→right→left→color == RED) || (P→right→left→color == RED))
       {
         P→color = RED;  P→left→color = BLACK;  P→right→color = BLACK;
       }
    }
    if(P→left→color == RED) {
       if( P→left→left→color == RED) {
```

MachineLearning > MTechClasses

```
if((P→left→color = = RED) && (P→ right →
{
  if((P→left→left→color = = RED) || (P→left →right→color == RED)
        || (P→right→left→color == RED) || (P→right→left →color == RED)
    {
      P→color = RED;  P→left→color = BLACK;  P→right → color = BLACK;
    }
}
if(P→left→color = = RED) {
    if( P→left →left→color == RED) {
    P  =   rotate right(P);
    P.
```

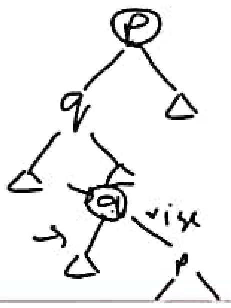



P
RC
R
PGC
R
PC
P
PGC
R

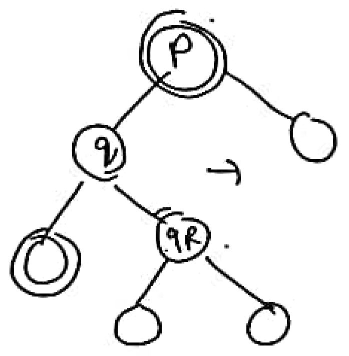

MachineLearning > MTechClasses

MachineLearning > MTechClasses

```
    if(q->left ...
        || ( P->right ->left ->color == RED)|| (P->right -> l...
    {
      P->color = RED;   P->left ->color = BLACK;   P->right -> color = BLACK;
    }
}
if( P->left -> color == RED ) {
   if ( P->left ->left ->color == RED) {
   P  =  rotate right( P ) ;
    P->color = BLACK ;
    P-> right -> Color = RED;
```

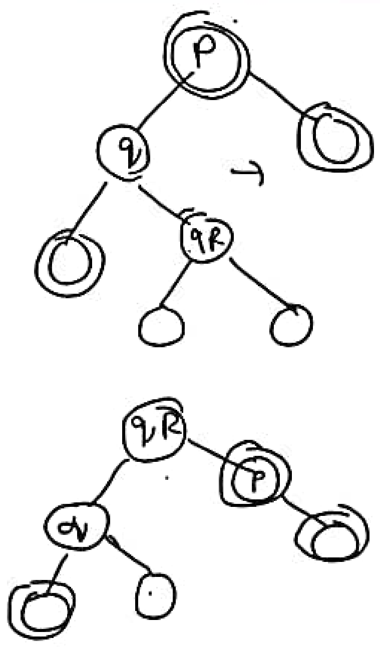


```
}
if( P->left -> color = = RED ) {
    if ( P->left ->left ->color == RED) {
    P  =  rotate right( P );
     P->color = BLACK ;
     P-> right -> Color = RED;
    }
     if ( P-> left -> right ->color == RED){
          rotate left Right (P) ;
```



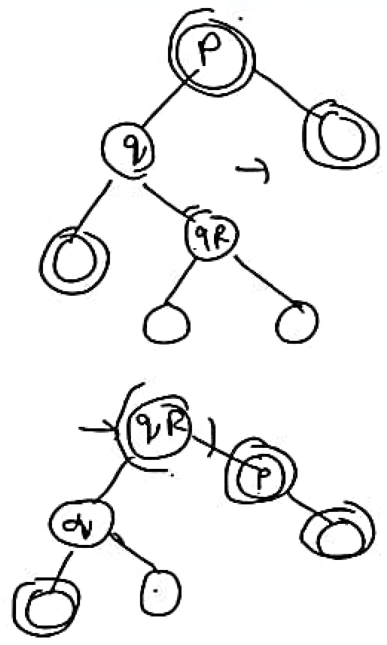
MachineLearning > MTechClasses

```
P→ right → color =
}
if (P→ left → right → color == RED){
        rotate left Right (P);
```

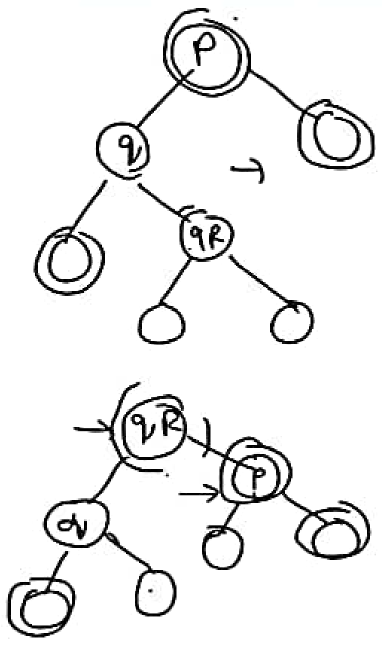


```
P→ right → color =
}
if ( P→ left → right → color == RED)){
 P = rotate left Right (P);
```

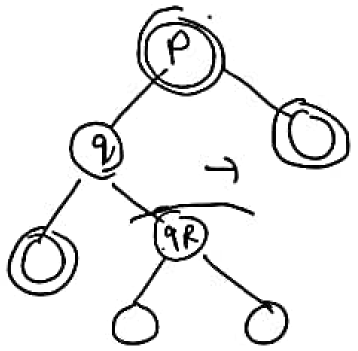
MachineLearning > MTechClasses



```
    P→ right → Color = 
}
 if ( P→ left → right → color == RED){
   P = rotate left Right (P) ;
   P→color = BLACK
   P→ right → Color = RED
```

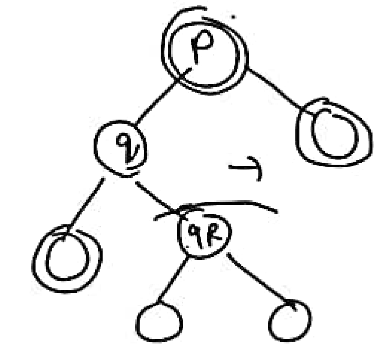


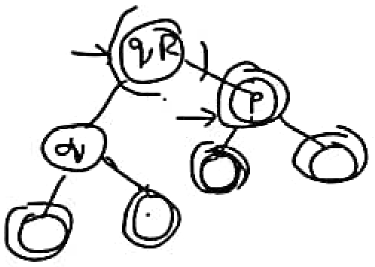
```
}
if( P→left → color = = RED ) {
    if( P→left → left → color == RED) {
    P =  rotate right( P );
     P→color = BLACK ;
     P→ right → Color = RED;
    }
     if( P→ left → right → color == RED){
      P = rotate left Right (P) ;
      P→color = BLACK
      P→ right → Color = RED
     }
}
```



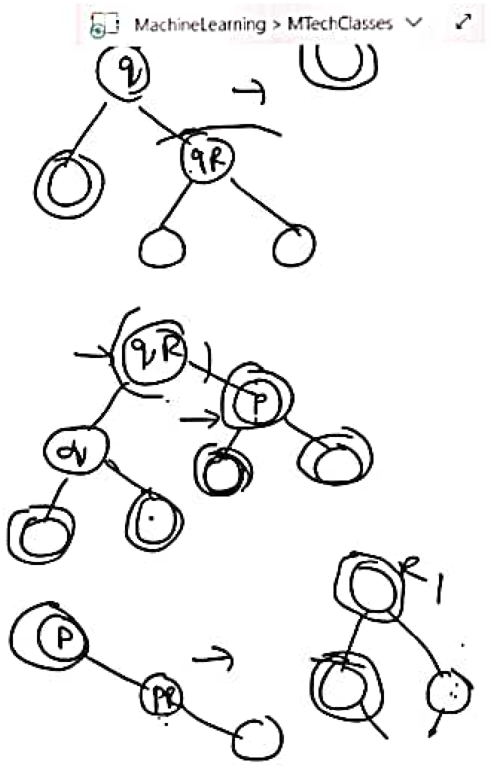
MachineLearning > MTechClasses

```
    P→ right → Color = RE
  }
  if ( P→left→right→color == RED){
     P = rotate left Right (P);
     P→color = BLACK
     P→right → Color = RED
  }
}
if ( P→ right →color == RED) {
  if(P→right →right →color==RED)) {
```



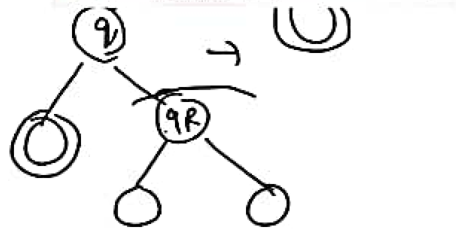


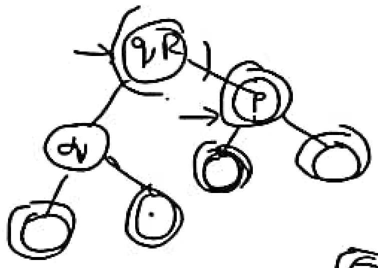
```
        P = rotate left Right (P);
        P->color = BLACK
        P->right -> Color = RED
      }
}
if ( P -> right ->color == RED) {
   if(P->right ->right ->color == RED) {
        P = rotate left(P))
        P->color = BLACK
        P->left->color = RED.
   }
```

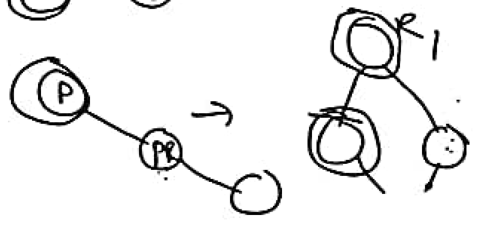


MachineLearning > MTechClasses

```
        P = rotate left Right (P);
        P->color = BLACK
        P->right->Color = RED
      }
}
if ( P-> right ->color == RED) {
  if(P->right->right->color == RED) {
        P = rotate left(P))
        P->color = BLACK
        P->left->color = RED
  }
  elif ( P-> right->left->color == RED) {
        P = rotate right Left (P);
          P->color = BLACK;
```

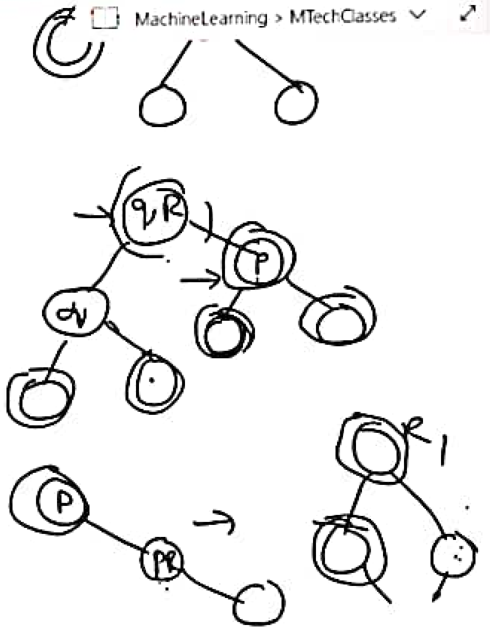




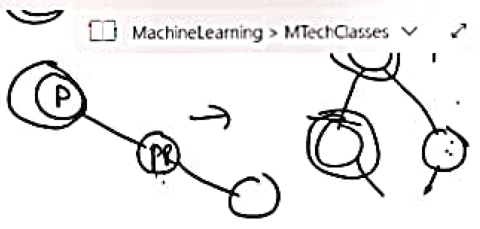


MachineLearning > MTechClasses

```
    P → right → colo
  }
}
if ( P → right → color = = RED ) {
  if( P → right → right → color = = RED ) {
       P = rotateleft(P))
       P → color = BLACK
       P → left → color = RED
  }
  elif ( P → right → left → color = = RED) {
       P ~~→~~ = rotaterightLeft ( P );
          P → color : BLACK ;
          P → left → color : BLACK ;
       }
```



```
        P->left->color = RED.
}
Elif (  P->right->left->color = = RED) {
        P = rotate right Left (P);
            P->color : BLACK;
            P->left->color : BLACK;
        }
}
```


MachineLearning > MTechClasses
P
PR