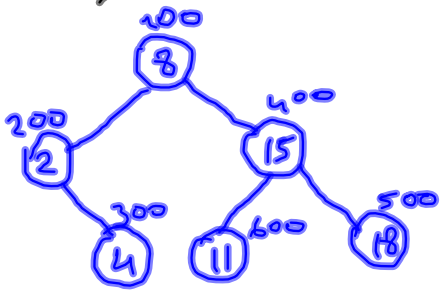
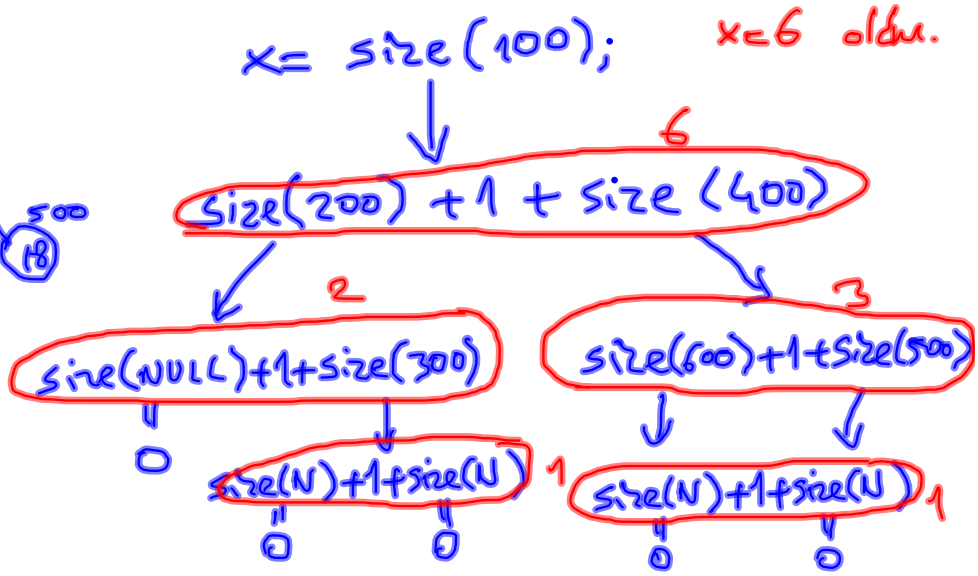
Bir Ağacın Düğümleri Sayısını Bulma

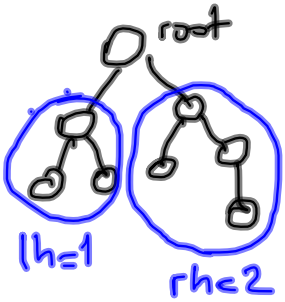
```
int size(BTREE root){
    if(root==NULL)
        return 0;
    else
        return size(root->left)+1+size(root->right);
}
```





Bir Ağacın Yüksekliği

```
int height(BTREE root){
    if(root==NULL)
        return 0;
    else{
        int lheight, rheight;
        lheight= height(root->left);
        rheight= height(root->right);
        if(rheight > lheight)
            return rheight+1;
        else
            return lheight+1;
    }
}
```



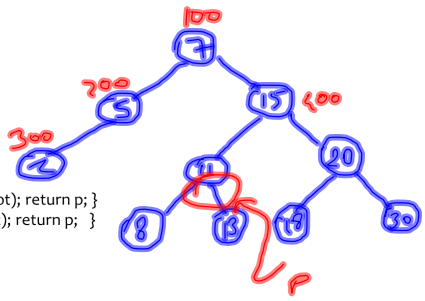
İkili Arama Ağacından bir Düğüm Silme

```
BTREE delete_node(BTREE root,int x)
{   BTREE p,q;

    if(root==NULL) return NULL; // no tree

    if(root->data==x)                       // find x in root
    {       if(root->left==root->right)     // root is a  leaf node
                    { free(root); return NULL; }
            else    {
                     if(root->left==NULL)           { p=root->right; free(root); return p; }
                    else if(root->right==NULL)      { p=root->left; free(root); return p;  }
                    else {      p=q=root->right;
                                while(p->left!=NULL) p=p->left;
                                p->left=root->left;
                                free(root);
                                return q;  }
                    }
    }
    if(root->data<x)                { root->right=delete_node(root->right,x); }
    else                            { root->left=delete_node(root->left,x); }
    return root;

}
```

1. DURUM

2.DURUM

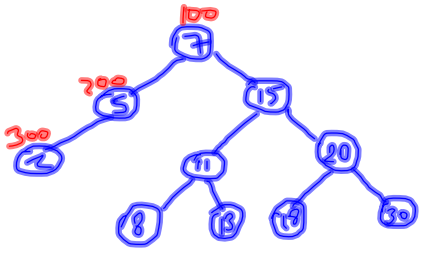
3. DURUM



Üç durum

1. Silinecek olan düğümün çocuğu yok ise.
2. sadece 1 "i var ise
3. 2 çocuğu var ise.

Örnek:



1.) x=2 silinsin

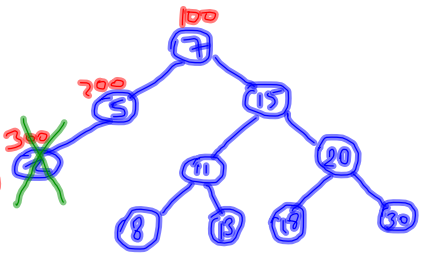
delete_node(100,2)

↓

delete_node(200,2)

↓

delete_node(300,2)



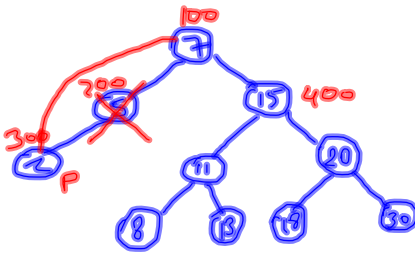
2.) x=5 silinsin

delete_node(100,5)

↓

root->left=delete(200,5)

p



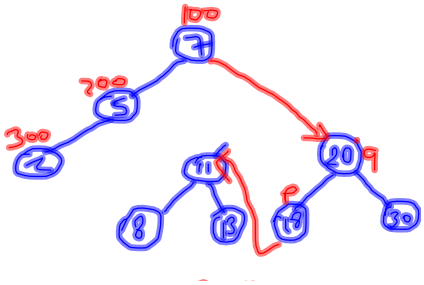
3.) x=15 silinsin

delete_node(100,15)

↓

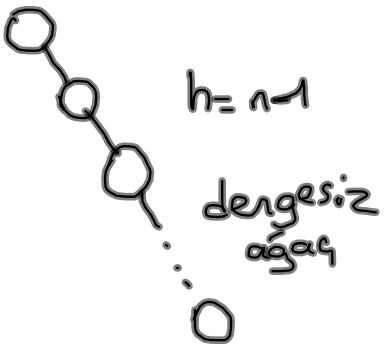
root->right=delete(400,15)

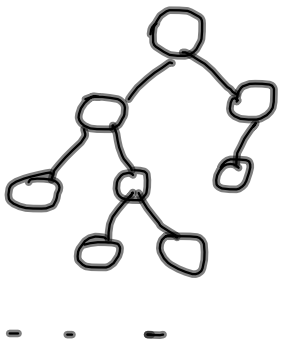
9



AVL Ağaçları

Dengeli ağaçlardır.

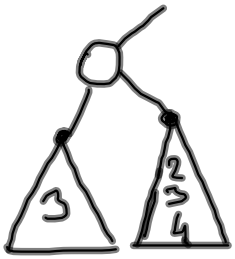




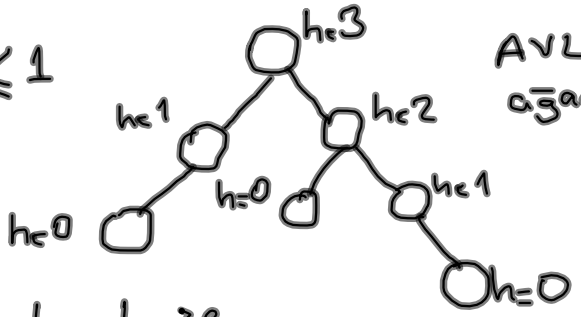
dengeli ağaç

AVL ağaçları BST'dir.

AVL ağaçların herig düğümünün sol ve sağ alt ağaçları arasındaki yükseklik farkı en fazla 1'dir.

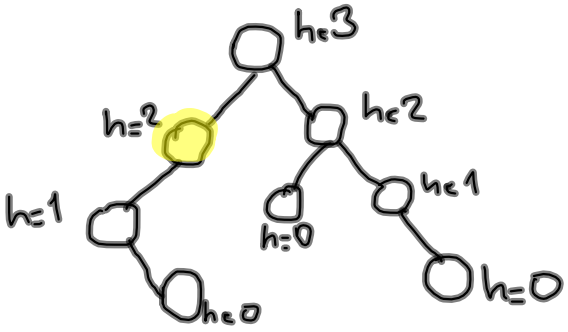


$|h_l - h_r| \leq 1$



AVL ağacıdır.

NOT: Sağ veya sol çocuk yok ise yada NULL değerlerin yüksekliği -1 dir.



AVL değildir.

"Önerme: n düğümlü bir AVL ağacının yüksekliği $O(\lg n)$ dir.

Soru: denge ne sağlar? Performans

Önermeyi ispatlamak için $f(h)$ değerini tanımlayalım.

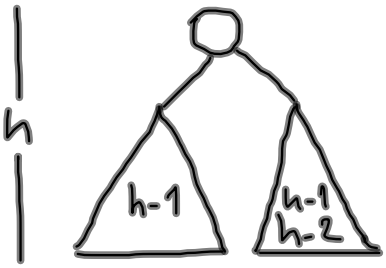
$f(h)$: yüksekliği h olan bir AVL ağacındaki minimum düğüm sayısı.

$f(0) = 1$ $\quad f(1) = 2$

$h \geq 2$ için, h yüksekliğindeki bir AVL ağacı

kök içerir + $h-1$ yüksekliğinde bir alt ağaç

+ $h-1$ veya $h-2$ " bir diğer alt ağaç



$f(h) = 1 + f(h-1) + f(h-2)$

Biliyoruz ki $f(h-1) \geq f(h-2)$

O zaman

$f(h) \geq 1 + 2f(h-2)$

$f(h) > 2f(h-2)$

$> 2 \cdot \underbrace{2 \cdot f(h-4)}_{f(h)} = 4f(h-4)$

$> 8f(h-6)$

$> 16f(h-8)$

$\vdots$

$f(h) > 2^i f(h-2i)$

$i = \frac{h}{2}$ iken $\quad f(h) > 2^{h/2} \cdot f(0) = 2^{h/2}$

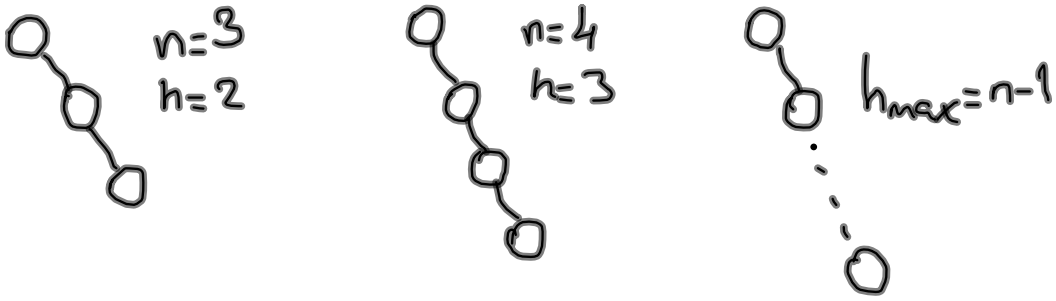
İki tarafın log'unu alalım $\quad \lg f(h) > \frac{h}{2} \Rightarrow h < 2 \lg f(h)$

Böylece bir AVL ağacının yüksekliği $O(\lg n)$ dir. □

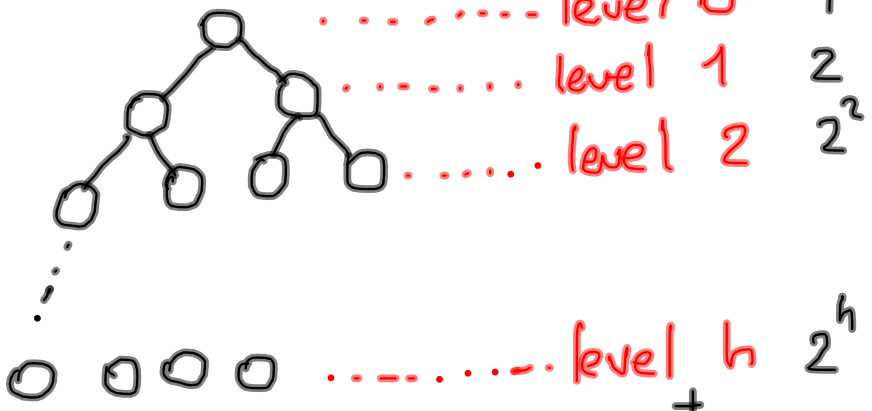
Ödev: n düğümlü bir ikili ağacın maksimum ve minimum yüksekliği nedir?

$\lg n = \log_2 n$

Çözüm:



min durumu:



$$1+2+2^2+\dots+2^h = 2^{h+1}-1 \geq n$$

Geometrik seri

$$\Rightarrow 2^{h+1} \geq n+1 \Rightarrow h+1 \geq \lg(n+1) \Rightarrow h \geq \lg(n+1)-1$$

$$h_{min} = \lg(n+1)-1$$