

Veri Yapilari ve Algoritmaları

odev 5 · 10 Mayıs 2020

Hazırlayan :

[Onur Akbal](#)

[171802074](#)

Gazi Üniversitesi

Bilgisayar ve Elektrik-Elektronik Mühendisliği

Öğretim Görevlisi

Doç.Dr. Nursal Arıcı

PROGRAM CALISTIRILMASI

```
Run: Main
C:\Program Files\Java\jdk1.8.0_221\bin\java.exe ...
1.kez RR Rotate Uygulandi.
2.kez RR Rotate Uygulandi.
1.kez RL Rotate Uygulandi.
PreOrder dolasma ile AVL Agac yapisi :
33 21 12 27 45 56

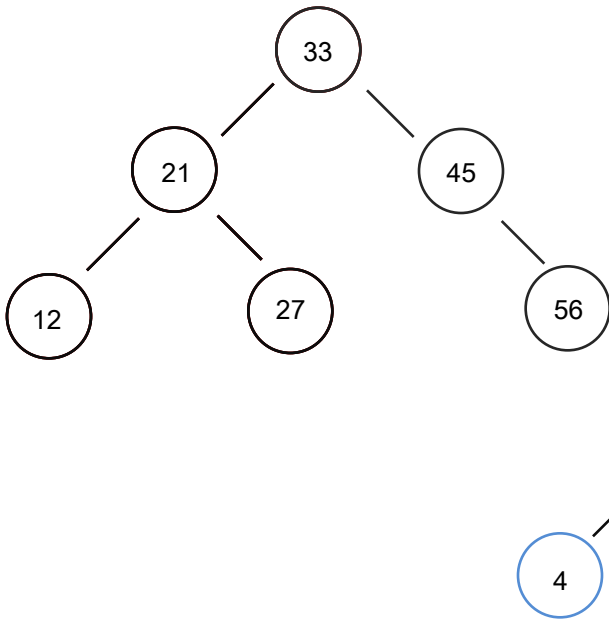
Agac Yapisina '5' ve '4' elemanlari eklendi

1.kez LL Rotate Uygulandi.

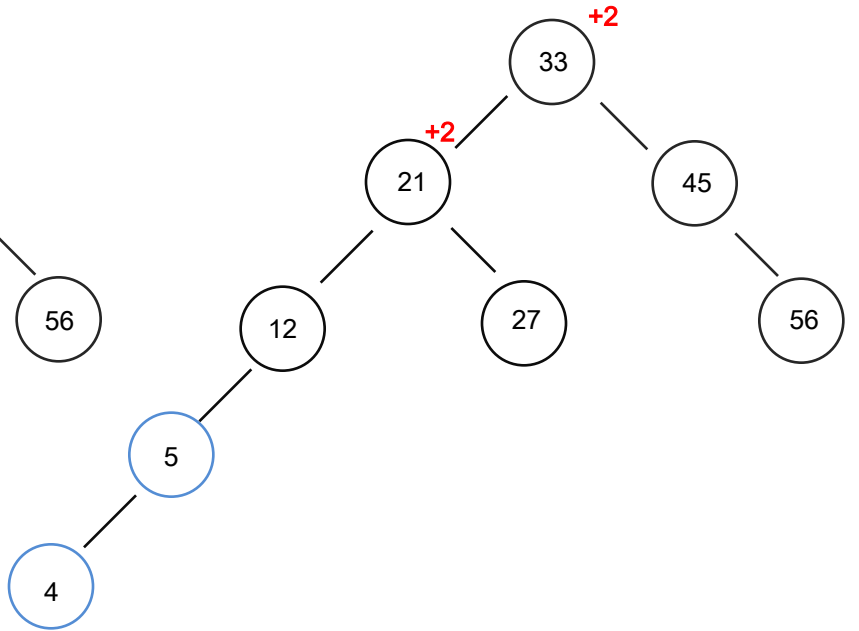
PreOrder dolasma ile AVL Agac yapisi :
33 21 5 4 12 27 45 56

Process finished with exit code 0
```

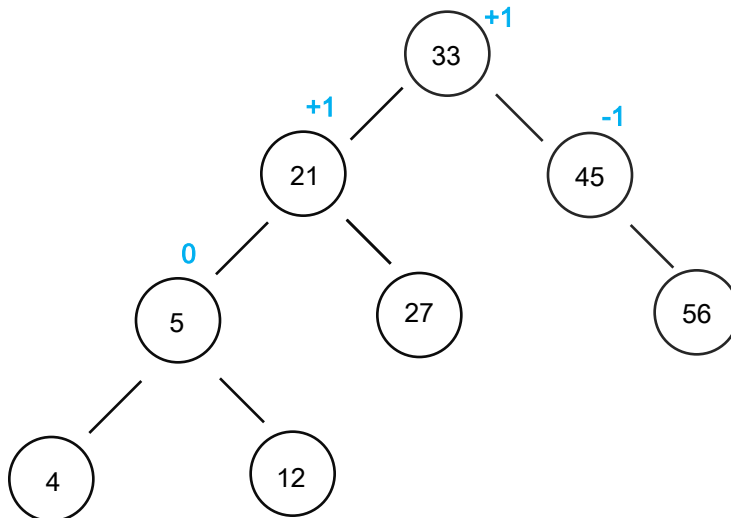
ILK DURUMDA AVL AGACIMIZ



ELEMANLAR EKLENINCE AGAC YAPISI



ROTATE ISLEMI SONUNDAKI AVL AGAC YAPISI



Main Sınıfı

```
1 package com.company;
2
3 // @author Onur Akbal
4
5 public class Main {
6
7     public static void main(String[] args) {
8         AVLTree agac = new AVLTree();
9
10        agac.kok = agac.ekle(agac.kok, icerik: 12);
11        agac.kok = agac.ekle(agac.kok, icerik: 21);
12        agac.kok = agac.ekle(agac.kok, icerik: 33);
13        agac.kok = agac.ekle(agac.kok, icerik: 45);
14        agac.kok = agac.ekle(agac.kok, icerik: 56);
15        agac.kok = agac.ekle(agac.kok, icerik: 27);
16
17        /* ilk bastaki AVL Agaci
18
19        33
20        /  \
21       21  45
22      /  \  \
23     12  27  56
24
25        */
26
27        System.out.println("PreOrder dolasma ile AVL Agac yapisi :");
28        agac.preOrder(agac.kok);
29        System.out.println("\n");
30
31        System.out.println("Agac Yapısına `5` ve `4` elemanlari eklendi\n");
32
33
34        agac.kok = agac.ekle(agac.kok, icerik: 5);
35
36        agac.kok = agac.ekle(agac.kok, icerik: 4); // EN SOL TARAFI İKİ ELEMAN EKLEYEREK Dengeyi BOZUYORUZ.
37
38
39        System.out.println("\nPreOrder dolasma ile AVL Agac yapisi :");
40        agac.preOrder(agac.kok);
41        System.out.println("\n");
42        /* Sonraki Rotate isleminde sonra bu hali alacaktır
43        Dengeyi hallededir.
44
45        33
46        /  \
47       21  45
48      /  \  \
49     5   27  56
50    /  \
51   4   12
52
53        */
54
55    }
56 }
```

Dugum Sınıfı

```
1 package com.company;
2
3 public class Dugum {
4     int icerik, boy;
5     Dugum sol, sag;
6
7     Dugum(int d) {
8         icerik = d;
9         boy = 1;
10    }
11 }
12
```

AVLTree Sınıfı

```
1 package com.company;
2
3 public class AVLTree {
4     Dugum kok;
5
6     int boy(Dugum N) {
7         if (N == null)
8             return 0;
9
10        return N.boy;
11    }
12
13    // Maksimum olani dondurme
14    int max(int a, int b) {
15        return (a > b) ? a : b;
16    }
17    int sayac=0;
18    int sayac1=0;
19    int sayac2=0;
20    int sayac3=0;
21
22    @ Dugum rightRotate(Dugum y) {
23        Dugum x = y.sol;
24        Dugum T2 = x.sag;
25
26        // Dondurme islemi
27        x.sag = y;
28        y.sol = T2;
29
30        // boyu guncellemek icin
31        y.boy = max(boy(y.sol), boy(y.sag)) + 1;
32        x.boy = max(boy(x.sol), boy(x.sag)) + 1;
33
34        return x;
35    }
36
37    @ Dugum leftRotate(Dugum x) {
38        Dugum y = x.sag;
39        Dugum T2 = y.sol;
40
41        // Dondurme Islemi
42        y.sol = x;
43        x.sag = T2;
44
45        // boy guncelleme
46        x.boy = max(boy(x.sol), boy(x.sag)) + 1;
47        y.boy = max(boy(y.sol), boy(y.sag)) + 1;
48
49        // dugum tipinde y yi dondur
50        return y;
51    }
52
53    // Denge faktoru hesaplanmasi
54    int getBalance(Dugum N) {
55        if (N == null)
56            return 0;
57
58        return boy(N.sol) - boy(N.sag);
59    }
60
61    Dugum ekle(Dugum dugum, int icerik) {
62
63        if (dugum == null)
64            return (new Dugum(icerik));
65
66        if (icerik < dugum.icerik)
67            dugum.sol = ekle(dugum.sol, icerik);
68        else if (icerik > dugum.icerik)
69            dugum.sag = ekle(dugum.sag, icerik);
70        else
71            return dugum;
72
73        dugum.boy = 1 + max(boy(dugum.sol),
74            boy(dugum.sag));
75
76        // denge kontrolu icin balance hesaplama
77        int balance = getBalance(dugum);
78
79        // Sol Sol Rotasyon
80        if (balance > 1 && icerik < dugum.sol.icerik) {
81            sayac++;
82            System.out.println(sayac+ ".kez LL Rotate Uygulandi.");
83            return rightRotate(dugum);
84        }
85
86        // Sag Sag Rotasyon
87        if (balance < -1 && icerik > dugum.sag.icerik) {
88            sayac3++;
89            System.out.println(sayac3+ ".kez RR Rotate Uygulandi.");
90            return leftRotate(dugum);
91        }
92    }
```

```
92         return leftRotate(dugum);
93     }
94     // Sol Sag Rotasyon
95     if (balance > 1 && icerik > dugum.sol.icerik) {
96         sayac2++;
97         System.out.println(sayac2+ " .kez LR Rotate Uygulandi.");
98         dugum.sol = leftRotate(dugum.sol);
99         return rightRotate(dugum);
100     }
101
102     // Sag Sol Rotasyon
103     if (balance < -1 && icerik < dugum.sag.icerik) {
104         sayac1++;
105         System.out.println(sayac1+ " .kez RL Rotate Uygulandi.");
106         dugum.sag = rightRotate(dugum.sag);
107         return leftRotate(dugum);
108     }
109     return dugum;
110 }
111
112 // pre order dolasma
113 void preOrder(Dugum dugum) {
114     if (dugum != null) {
115         System.out.print(dugum.icerik + " ");
116         preOrder(dugum.sol);
117         preOrder(dugum.sag);
118     }
119 }
120 }
```