

**Gebze Technical University
Computer Engineering**

CSE 222 - 2018 Spring

HOMEWORK 6 REPORT

**Ramazan Guvenc
161044037**

Course Assistant: Fatma Nur Esirci

1 Worst RedBlack Tree

This part about Question1 in HW6

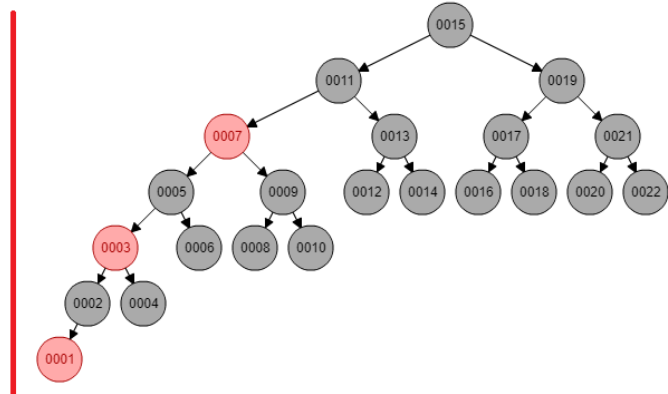
1.1 Problem Solution Approach

Galiba her search tree de yani sorted treelerde worst case ornegin bst de olduđu gibi sorted veya tersten sorted bir liste inputu vermek. Bende heighti 6 olana kadar for dongusu içinde tersten sorted bir input dizisi verdim. Sonuc olarak RedBlackTree için worst case i elde etmiş oldum.

1.2 Running Commands and Results

```
Black: 15
Black: 11
Red : 7
Black: 5
Red : 3
Black: 2
Red : 1
null
null
null
Black: 4
null
null
null
Black: 6
null
null
null
Black: 9
Black: 8
null
null
Black: 10
null
null
null
Black: 13
Black: 12
null
null
Black: 14
null
null
null
Black: 19
Black: 17
Black: 16
null
null
null
Black: 18
null
null
null
Black: 21
Black: 20
null
null
Black: 22
null
null
null
```

HEIGHT = 6



2 binarySearch method

This part about Question2 in HW6

2.1 Problem Solution Approach

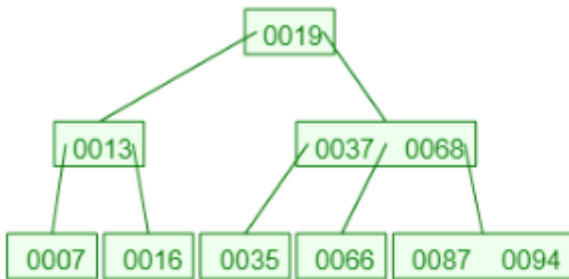
Bizden istene binarySearch aslında binarySearch denilince aklımıza gelen şey değil. Kitaptaki

soruda bu binarySearchten beklentilerinin itemin hangi indexe koyulması gerektiğini veya itemin aynısından varsa o itemin indexinin return edilmesi gerektiğini yazmış. Bu indexide compareTo ile karşılaştırıp eğer suanki data[i] de bulunan değerden küçükse child'e inip aynı i'ye doldurması yoksa for döngüsü ile küçük olan bir değer bulunana kadar aynı şeyi yapıyorum. Eğer hiç ufak değer bulamıyorsak data.length-1 donuyor.

2.2 Test Cases

Zaten insert'in içinde bu binarySearch kullanıldığı için add methodu çağırıldığında otomatik olarak kullanılıyor binarySearch. Hiç bir problemle karşılaşılmadı.

2.3 Running Commands and Results



Verdiğiniz siteden alınmış inputlara göre çıkması gereken ağac.

```
19
 13
  7
    null
    null

 16
    null
    null

 37, 68
  35
    null
    null

 66
  null
  null

 87, 94
  null
  null
  null
```

Cıkan ağac.

3 Project 9.5 in book

```
// AVLTree kodu eksik methodlari tarafimdan YAZILMAMISTIR.  
// AVLTree yi kitabın verdiği 2. kaynaktan kopyaladım çünkü yoksa add methodu düzgün çalışmıyordu  
// Bu yüzden AVLTree yi kontrol edecektim.  
// Bu arada constructora boş Tree gönderince method çalışmayabilir garantisini veremiyorum.
```

3.1 Problem Solution Approach

Burada belirtmeliyim ki sadece constructor görevini kendim tamamladım. Diğer remove, rebalance vb kodları zaten kitabın temin ettiği kodlardan aldım. Constructora dönecek olursak verdiğiniz binaryTree nin AVL olup olmadığını kontrol etmemizi istemistidiniz. Bunun için iki şart aradım. Birincisi verilen Binary Tree nin sıralı olması yani binary search tree veya diğer search tree ler gibi olması ve ağacın dengeli olması.

3.2 Test Cases

Bir tane AVL tree ve bir tane bst verildi. Constructor verilen tree ye göre konsola print yapıyor.

3.3 Running Commands and Results

```
public static void main(String[] args) {  
    AVLTree<Integer> av = new AVLTree<>();  
    for(int i =0; i < 10 ; i ++)  
        av.add(i);  
    AVLTree<Integer> k = new AVLTree<>(av);  
    BinarySearchTree<Integer> bst = new BinarySearchTree<>();  
    for(int i =0 ; i < 10 ; i ++)  
        bst.add(i);  
    AVLTree<Integer> t = new AVLTree<>(bst);  
}
```

```
Its AVL  
Its not an AVL
```