Veri Yapıları Laboratuvar Föyü
Hafta 4
BinarySearchTree clone, successor, predecessor ve findParent Metodlarının Yazımı

Şevket Umut ÇAKIR

1 Giriş

Bu deneyde soyut olarak verilen BinarySearchTree sınıfının soyut olan clone, successor, predecessor ve findParent metodlarının yazımının LabBinarySearchTree sınıfı içinde gerçekleştirilmesi amaçlanmaktadır.

2 Konu Anlatımı ve Deney Hazırlığı

Derste anlatılan ikili arama ağaçları ile ilgili düğüm sınıfı olan BTNode ve ikili arama ağacını gerçekleştiren soyut BinarySearchTree sınıfları verilmektedir. BTNode sınıfının yapısı aşağıda verilmiştir.

```
public class BTNode<T> {
    public BTNode<T> left;
    public BTNode<T> right;
    public T value;

    public BTNode(T value, BTNode<T> left, BTNode<T> right) {
        this.left = left;
        this.right = right;
        this.value = value;
    }
}
```

`left` referansı düğümün sol çocuğunu, `right` referansı da sağ çocuğunu göstermektedir. `value` alanı ise o düğümün değerini gösterir. BTNode sınıfının üç parametreli bir yapıcı metodu da bulunmaktadır. İkili arama ağacını temsil eden BinarySearchTree sınıfının yapısı aşağıda verilmektedir.

```
1 public abstract class BinarySearchTree<T extends Comparable<T>> {
2     private BTNode<T> root;//Kök düğüm
```

```
3
4       public BinarySearchTree(BTNode<T> root) {
5           this.root = root;
6       }
7
8       public BinarySearchTree() { }
9
10      public BTNode<T> getRoot() {
11          return root;
12      }
13
14      public BTNode<T> find(BTNode<T>node, T value){
15          if(node==null || node.value==value)
16              return node;
17          else if(value.compareTo(node.value)<0)
18              return find(node.left, value);
19          else
20              return find(node.right, value);
21      }
22
23      public boolean contains(T value){
24          return find(root, value)!=null;
25      }
26
27      public void add(T value){
28          add(root, value);
29      }
30
31      private void add(BTNode<T> node, T value) {
32          if(root==null) {
33              root = new BTNode<>(value, null, null);
34              return;
35          }
36          if(value.compareTo(node.value)<0) {
37              if(node.left==null)
38                  node.left = new BTNode<>(value, null, null);
39              else
40                  add(node.left, value);
41          }
42          else if(value.compareTo(node.value)>0){
43              if(node.right==null)
44                  node.right=new BTNode<>(value, null, null);
45              else
46                  add(node.right, value);
47          }
48          else throw new RuntimeException("Eleman ağaçta mevcut!");
```

```
49      }
50
51      @Override
52      protected abstract Object clone() throws
        ↪  CloneNotSupportedException;
53      public abstract BTNode<T> successor(T value);
54      public abstract BTNode<T> predecessor(T value);
55      public abstract BTNode<T> findParent(BTNode<T> node);
56  }
```

`root` alanı kök düğümü gösterir. `BinarySearchTree` sınıfının iki adet yapıcı metodu bulunmaktadır. Bir tanesi boş bir ağaç oluştururken diğeri kök düğümünü parametre olarak ağaç oluşturmaya imkan verir. `getRoot` metodu kök düğümü verir. `find` metodu özyineli olarak bir düğümü bulur. `contains` metodu verilen bir değerin ağaçta olup olmadığını verir. `add` metodu ağaca eleman eklemek için kullanılır.

3 Deneyin Uygulanması

Deneyde soyut olarak verilen dört metodun gerçekleştirilmesi istenmektedir. İlk olarak Object sınıfından devralınan `clone` metodunun yazılması beklenmektedir. clone metodu ağacın bir kopyasını oluşturmaktadır. Yeni ağaçtaki düğümlerin referansları kopyalanan ağaçtaki düğümlerin referanslarından farklı olmalıdır. Yani düğümler aynı değerlerle yeniden oluşturulmalıdır.

`successor` metodu, varsa, ağaçtaki verilen değerden büyük, en küçük değeri döndürmelidir. `predecessor` metodu ise, varsa, ağaçtaki verilen değerden büyük, en küçük değeri döndürmelidir.

`findParent` metodu verilen düğümün ağaçtaki ebeveyn düğümünü bulan metoddur. Bu dört metod soyut olarak verilen `BinarySearchTree` sınıfında soyut olarak bulunmaktadır. Size verilen `LabBinarySearchTree` sınıfında bu dört metodun gerçekleştirilmesi istenmektedir.