

Veri Yapıları Laboratuvarı Ödev 3

Öğr.Gör.Şevket Umut ÇAKIR
CENG 215 - Veri Yapıları

18 Kasım 2019

Ödevde Huffman ağaçlarını karşılaştırmak için kullanılacak, `Comparator` arayüzünü gerçekleştiren iki adet sınıf yazılması istenmektedir. `Comparator` arayüzünün gerçekleştirilmesi ile ilgili aşağıda bir örnek verilmiştir.

Huffman ağacı Huffman kodlama için oluşturulan bir ağaç yapısıdır. Ağacın bir kök düğümü bulunur, kökün sol ve sağ tarafında aynı türde düğümler bulunabilir. Düğüm yapısını temsil etmek için aşağıdaki `HuffmanNode` sınıfı verilmiştir. `value` değeri `null` ise o düğüm bir karakteri temsil etmez ve yaprak değildir. Yapraklardaki düğümlerin `value` değeri `null`'dan farklıdır. Karşılaştırmacı sınıflar `HuffmanNode` sınıflarını karşılaştırabilir nitelikte yazılmalıdır.

```
1  /**
2   * Huffman düğüm sınıfı. Bu sınıfın içeriğini değiştirmemeniz
   ↪ gerekmektedir. Değişiklik yaparsanız test ortamında kodunuz beklendiği
   ↪ gibi çalışmayacaktır.
3   */
4  public class HuffmanNode {
5      public Character value; // Karakter
6      public int frequency; // Frekansı
7      public HuffmanNode left; // sol çocuk
8      public HuffmanNode right; // sağ çocuk
9
10     public HuffmanNode(Character value, int frequency, HuffmanNode left,
   ↪ HuffmanNode right) {
11         this.value = value;
12         this.frequency = frequency;
13         this.left = left;
14         this.right = right;
15     }
16 }
```

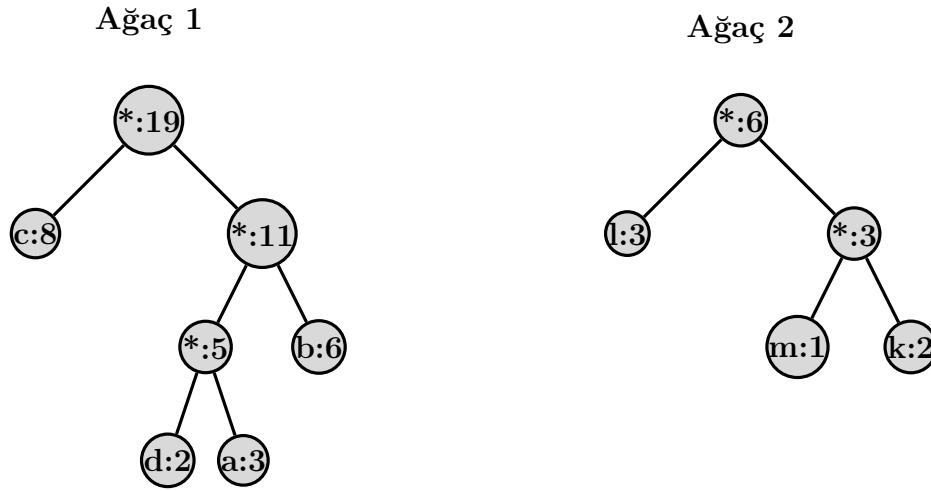
1 FrekansKarsilastirici Sınıfı

`FrekansKarsilastirici` sınıfı kök düğümleri verilen iki ağacın köklerini frekans değerlerine göre karşılaştırır. Kökteki frekansı değeri az olan ağaç daha küçüktür. Şekil 1'de Ağaç 2

frekansa göre Ağaç 1'den küçüktür. `FrekansKarsilastirici` sınıfının `compare` metodunu yazınız. `compare` metodundan dönen değer 1. ağaç küçükse `-1`, ağaçlar eşit ise `0` ve 1. ağaç büyükse `1` olmalıdır.

2 AlfabetikKarsilastirici Sınıfı

`AlfabetikKarsilastirici` sınıfı Huffman ağaçlarını içlerinde bulunan en küçük karakterlere göre karşılaştırır. Şekil 1'de Ağaç 1 alfabetik sıraya göre Ağaç 2'den daha küçüktür. `AlfabetikKarsilastirici` sınıfının `compare` metodunu yazınız. `compare` metodundan dönen değer 1. ağaç küçükse `-1`, ağaçlar eşit ise `0` ve 1. ağaç büyükse `1` olmalıdır.



Şekil 1: Örnek ağaçlar

```
1 /**
2  * Comparator arayüzünün kullanımına örnek
3  */
4 import java.util.ArrayList;
5 import java.util.Comparator;
6
7 class Sanatci {
8     int dogumYili; //Doğum yılı
9     String adi; // Sanatçı adı
10    // Constructor
11    public Sanatci(int dogumYili, String adi) {
12        this.dogumYili = dogumYili;
13        this.adi = adi;
14    }
15    // Geçersiz kılınmış toString metodu: sanatçının doğum yılını ve adını
16    // ↪ verir.
17    @Override
```

```

17     public String toString() {
18         return adi+" (" +dogumYili+" )";
19     }
20 }
21
22 class IsimKarsilastirici implements Comparator<Sanatci>{
23     // Sanatçı adlarını karşılaştırır
24     @Override
25     public int compare(Sanatci o1, Sanatci o2) {
26         return o1.adi.compareTo(o2.adi);
27     }
28 }
29
30 class DogumYiliKarsilastirici implements Comparator<Sanatci>{
31     // Sanatçıların doğum yıllarını karşılaştırır(büyükten küçüğe)
32     @Override
33     public int compare(Sanatci o1, Sanatci o2) {
34         return o2.dogumYili - o1.dogumYili;
35     }
36 }
37
38 class Test {
39     public static void main(String[] args) {
40         ArrayList<Sanatci> sanatcilar=new ArrayList<>();
41         sanatcilar.add(new Sanatci(1946, "Musa Eroğlu"));
42         sanatcilar.add(new Sanatci(1945, "Arif Sağ"));
43         sanatcilar.add(new Sanatci(1955, "Sabahat Akkiraz"));
44         sanatcilar.add(new Sanatci(1952, "Belkıs Akkale"));
45         sanatcilar.sort(new DogumYiliKarsilastirici()); // doğum yılına
46         ↪ göre sırala
47         yazdir(sanatcilar);
48         sanatcilar.sort(new IsimKarsilastirici()); // ada göre sırala
49         yazdir(sanatcilar);
50     }
51     public static<T> void yazdir(ArrayList<T> liste){
52         for(T eleman:liste)
53             System.out.println(eleman);
54         System.out.println();
55     }
56 }

```

Önemli Tarihler :

Tablo 1: Önemli Tarihler

Olay	Tarih	Konum	Biçim
Ödev Teslimi	27.11.2019	bilmoodle.pau.edu.tr	Odev3Ogrenci.java

Ödev Teslimi ile İlgili Açıklamalar

- Ödevler Programlama Ödevleri Moodle Sistemi(<http://bilmoodle.pau.edu.tr/>) üzerine kaynak kod yüklenecektir.
- Ödevde girdiler rastgele olarak her değerlendirmede üretilmektedir. Kodunuzda hata varsa bazı girdilerde çalışıp, bazılarında çalışmayabilir. Dolayısıyla kodunuzu bir kaç defa değerlendirmeye göndermeniz önerilir.
- Ödevler teslim süresi bittikten sonra otomatik olarak değerlendirilecektir. Otomatik değerlendirme sonucu notunuzu belirleyecektir. Ödev teslim süresinden önce almış olduğunuz notlar yanıltıcı olabilir.
- Kaynak kod dosyasının en üstüne öğrenci numarası ve ad soyad açıklama satırı olarak eklenmek zorundadır.
- Ödevler bireysel olarak cevaplanacaktır. Kopya olduğu anlaşılan ödevlerin hepsine 0 puan verilecektir.