

PROJE RAPORU

Bu projede, ikili arama ağacı (Binary Search Tree-BST) veri yapısı üzerine yoğunlaştım. Öncesinde ikili arama ağaçlarının nasıl çalıştığını derinlemesine kavramam gerekti. Bunun için çeşit okumalar yaptım, ders notlarını kontrol ettim, çeşitli videolar izledim. İkili arama ağaçlarında eleman ekleme, eleman silme postorder, preorder, inorder gezinme fonksiyonları üzerine çalıştım. Bu aşamaları yeterince kavradıktan sonra ödevde başladım.

Ödevde bizde iki farklı ikili arama ağacı yaratarak bu ağaçlar üzerinde gezinmemiz, soysayılarını bulmamız, bulduğumuz soy sayılarına göre karşılaştırmalar yaparak çeşitlik skorları güncellememiz isteniyordu. Öncelikle soy sayılarını nasıl bulabileceğim üzerine düşündüm sonrasında tree üzerinde gezinme ile benzer bir recursive yapı kurabileceğimi farkettim ve onun için bir fonksiyon yazdım. Ayrıca bu değeri her bir node nesnesinde tutabilmek için Node sınıfına value değişkenini ekledim.

Daha sonra her bir node için soy sayılarını hesaplayarak ağaçların toplam say sayıları bulan fonksiyonu yazdım. Bu aşamayı tamamladıktan sonra soy sayılarına göre karşılaştırma yaparak çeşitlik skorları güncellemem ve ağaçlar üzerinde çeşitli ekleme ve çıkarma işlemleri yapmam gerekiyordu. Ekleme işlemi için yazdığım standart insert fonksiyonunu kullanabilirdim. Ancak node silme işlemi için iki farklı fonksiyon daha yazdım. Bunlardan birisi root nodunu silmek için diğeri ise maksimum değere sahip nodu silmek içindi. Bu fonksiyonları da tamamladıktan sonra ödevi bitirebilirdim.

Sonsuz bir döngü içinde, adım adım ağaçları karşılaştıran; her adımda kazanana göre skorları güncelleyen, ve ağaçlar üzerinde ekleme çıkarma işlemlerini yapan ana fonksiyomu da tamamladıktan sonra ödevi bitirdim.

Ödev ile ilgili görselleri aşağıta ekliyorum.

```
35     root = insertNode(root, key);
36 }
37
38 Node insertNode(Node root, int key) {
39     if (root == null) {
40         root = new Node(key);
41         return root;
42     }
43     if (key < root.key)
44         root.left = insertNode(root.left, key);
45     else if (key > root.key)
46         root.right = insertNode(root.right, key);
47     return root;
48 }
49
50 void postorder() {
51     postorderRec(root);
52 }
53
54 void postorderRec(Node root) {
55     if (root != null) {
56         postorderRec(root.left);
57         postorderRec(root.right);
58         System.out.print(root.value + " ");
59     }
60 }
```

Problems | Javadoc | Declaration | Console

<terminated> BST [Java Application] C:\Users\... \p2\poo\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe (29 Oca 2021 19:52:38 - 19:52:50)

=====Oyuncu 1 bu turu kazandı=====

Tur 10 Toplam Skor

Oyuncu1: 7

Oyuncu2: 2

Devam (0)

d

oyuncu1 agaci postorder gezinti sonucu :

[60, 0] [63, 1] [83, 2] [85, 3] [87, 4] [97, 0] [90, 6] [105, 0] [101, 1] [99, 9] [5, 10]

oyuncu1 agaci toplam soy sayisi : 36

=====

oyuncu2 agaci postorder gezinti sonucu :

[7, 0] [7, 1] [8, 2] [12, 3] [12, 4] [34, 5] [38, 6] [51, 0] [47, 1] [45, 9] [5, 10]

oyuncu2 agaci toplam soy sayisi : 41

=====Oyuncu 1 bu turu kazandı=====

Tur 11 Toplam Skor

Oyuncu1: 8

Oyuncu2: 2

Devam (0)

Oyun bitti! Oyuncu 1 kazandı!

```

37
38 Node insertNode(Node root, int key) {
39
40     if (root == null) {
41         root = new Node(key);
42         return root;
43     }
44
45     if (key < root.key)
46         root.left = insertNode(root.left, key);
47     else if (key > root.key)
48         root.right = insertNode(root.right, key);
49
50     return root;
51 }
52
53 void postorder() {
54     postorderRec(root);
55 }
56

```

Problems Javadoc Declaration Console

BST [Java Application] C:\Users\...p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe (29 Oca 2021 19:45:02)

```

oyuncu1 agaci postorder gezinti sonucu :
[7, 0] [83, 0] [12, 2] [87, 0] [97, 0] [90, 2] [101, 0] [105, 1] [99, 5] [85, 9]
oyuncu1 agaci toplam soy sayisi : 19
*****
oyuncu2 agaci postorder gezinti sonucu :
[8, 0] [5, 1] [34, 0] [38, 1] [51, 0] [47, 1] [60, 2] [45, 5] [63, 6] [12, 9]
oyuncu2 agaci toplam soy sayisi : 25
=====Oyuncu 1 bu turu kazandi=====
Tur 1 Toplam Skor
Oyuncu1: 1
Oyuncu2: 0
Devam (D)
d
oyuncu1 agaci postorder gezinti sonucu :
[7, 0] [63, 0] [12, 2] [87, 0] [97, 0] [90, 2] [101, 0] [105, 1] [99, 5] [83, 9]
oyuncu1 agaci toplam soy sayisi : 19
*****
oyuncu2 agaci postorder gezinti sonucu :
[8, 0] [5, 1] [34, 0] [38, 1] [51, 0] [47, 1] [85, 0] [60, 3] [45, 6] [12, 9]
oyuncu2 agaci toplam soy sayisi : 21
=====Oyuncu 1 bu turu kazandi=====

```

```

119 int soySayisi(Node root) {
120     if (root == null)
121         return 0;
122     int res = root.value;
123     res += soySayisi(root.right);
124     res += soySayisi(root.left);
125     return res;
126 }
127
128 public static void main(String[] args) {
129     String[] parts = null;
130     String[] parts2 = null;
131     try {
132         File myObj = new File("oyuncu1.txt");
133         Scanner myReader = new Scanner(myObj);
134         String key = myReader.nextLine();
135         parts = key.split("#");
136         myReader.close();

```

Problems Javadoc Declaration Console

<terminated> BST [Java Application] C:\Users\...p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_15.0.1.v20201027-0507\jre\bin\javaw.exe (29 Oca 2021 19:45:02)

```

oyuncu2 agaci toplam soy sayisi : 21
=====Oyuncu 2 bu turu kazandi=====
Tur 12 Toplam Skor
Oyuncu1: 6
Oyuncu2: 4
Devam (D)
D
oyuncu1 agaci postorder gezinti sonucu :
[8, 0] [83, 1] [85, 2] [87, 3] [97, 0] [90, 5] [101, 0] [99, 7] [5, 8]
oyuncu1 agaci toplam soy sayisi : 26
*****
oyuncu2 agaci postorder gezinti sonucu :
[7, 0] [12, 1] [34, 2] [38, 3] [51, 0] [47, 1] [105, 0] [63, 1] [60, 4] [45, 9] [5, 10]
oyuncu2 agaci toplam soy sayisi : 31
=====Oyuncu 1 bu turu kazandi=====
Tur 13 Toplam Skor
Oyuncu1: 7
Oyuncu2: 4
Devam (D)
Oyun bitti! Oyuncu 1 kazandi!

```