



**EGE UNIVERSITY**

**FACULTY OF ENGINEERING**

**COMPUTER ENGINEERING DEPARTMENT**

**204 DATA STRUCTURES (3+1)**

**2022–2023 FALL SEMESTER**

**PROJECT-3 REPORT**

**(Arrays, Matrices, Methods, Random Numbers)**

**DELIVERY DATE**

18/12/2022

**PREPARED BY**

05200000067, Celaldoğan Güneş

05200000080, Siyam Acet

## İçindekiler

1) MİLLİ PARKLAR .....	3
1.a Milli Parkları İçeren Ağacın Oluşturulması .....	3
1.a.1 Kodlar .....	3
1.a.2 Açıklama .....	4
1.b Ağacın Derinliğini ve Düğüm Sayısını Bulma ve Ağaçtaki Bilgileri Ekrana Listeleme .....	5
1.b.1 Kodlar .....	5
1.b.2 Ekran Görüntüsü .....	6
1.b.3 Açıklama .....	6
1.c Milli Parkın Bulunduğu İli Bulma .....	6
1.c.1 Kodlar .....	6
1.c.2 Ekran görüntüleri.....	6
1.d Bilgi Cümlesindeki Kelimelerin Ayrı Bir İkili Arama Ağacına Yerleştirilmesi.....	6
1.d.1 Kodlar .....	7
1.d.2 Ekran görüntüleri .....	7
1.d.3 Açıklama .....	9
2) HASH TABLE .....	0
2.a Milli Parkları Hash Table'a Yerleştirme .....	0
2.a.1 Kaynak Kod .....	0
2.a.2 Ekran Görüntüsü .....	0
2.b Hash Table Güncelleme .....	3
2.b.1 Kaynak Kod.....	3
2.b.2 Açıklama .....	3
2.b.3 Ekran Görüntüsü .....	3
3) HEAP.....	4
3.a Heap .....	4
3.a.1 Kaynak Kod.....	4
3.b Max Heap .....	7
3.b.1 Kaynak Kod.....	7
3.c Max Heap Listeleme .....	7
3.c.1 Kaynak Kod .....	10
3.c.2 Ekran Görüntüsü.....	10
3.c.3 Açıklama .....	10
4) SIRALAMA .....	11
4.a Sıralama Algoritması .....	11
4.a.1 Kaynak Kod .....	11

4.a.2 Açıklama .....	11
4.b Zaman Karmaşıklığı .....	11
4.b.1 Açıklama .....	11
4.c Görselleştirmenin Etkisi.....	12
4.c.1 Açıklama .....	12
4) ÖZDEĞERLENDİRME TABLOSU.....	12

## 1) MİLLİ PARKLAR

IntelliJ IDEA 2022.2.3 (Ultimate Edition) ortamında Java ile geliştirildi.

### 1.a Milli Parkları İçeren Ağacın Oluşturulması

#### 1.a.1 Kodlar

```
public class Tree {  
  
    13 usages  
    class TreeNode {  
        6 usages  
        public MilliPark data;  
        7 usages  
        public TreeNode leftChild;  
        7 usages  
        public TreeNode rightChild;  
  
        1 usage  
        public void displayNode() {  
            System.out.println(data);  
        }  
    }  
  
    5 usages  
    private TreeNode root;  
  
    1 usage  
    public Tree() {  
        root = null;  
    }  
  
    6 usages  
    public TreeNode getRoot() {  
        return root;  
    }  
}
```

```

public void insert(MilliPark milliPark) {
    TreeNode newNode = new TreeNode();
    newNode.data = milliPark;

    if (root == null) {
        root = newNode;
    } else {
        TreeNode current = root;
        TreeNode parent;
        while (true) {
            parent = current;
            if (Main.sortStringsWithTurkceKarakterDestegi(Main.trCollator, milliPark.parkAdi, current.data.parkAdi) < 0) { //park < current ise
                current = current.leftChild;
                if (current == null) {
                    parent.leftChild = newNode;
                    return;
                }
            } else {
                current = current.rightChild;
                if (current == null) {
                    parent.rightChild = newNode;
                    return;
                }
            }
        }
    }
}
}

```

### 1.a.2 Açıklama

Binary Search Tree veri yapısını kullandık. Insert metodunda milli parkların isimlerini Türkçe karakter desteği ile karşılaştırıp yerleştiriyoruz.

## 1.b Ağacın Derinliğini ve Düğüm Sayısını Bulma ve Ağaçtaki Bilgileri Ekrana Listeleme

### 1.b.1 Kodlar

```
53      /*
54
55      1.B
56
57      */
58
59      public int maxDepth(TreeNode root) {// ağacın derinliğini bulduran metod
60          if (root == null) {
61              return 0;
62          }
63          int leftDepth = maxDepth(root.leftChild);
64          int rightDepth = maxDepth(root.rightChild);
65
66          if (leftDepth > rightDepth) {
67              return leftDepth + 1;
68          } else {
69              return rightDepth + 1;
70          }
71      }
72
73      public void inOrder(TreeNode localroot) { // ağacı inOrder dolaşma metodu
74          if (localroot != null) {
75              inOrder(localroot.leftChild);
76              localroot.displayNode();
77              inOrder(localroot.rightChild);
78          }
79      }
80
81
82      public int totalNodes(TreeNode root) {// toplam düğüm sayısını bulduran metod
83          if (root == null)
84              return 0;
85
86          int l = totalNodes(root.leftChild);
87          int r = totalNodes(root.rightChild);
88
89          return 1 + l + r;
90      }
91
```

```
91
92      // Week 8 Binary Trees Sayfa 13 kaynaklı yöntem
93      // Ek olarak, https://stackoverflow.com/questions/9837891/balanced-binary-trees-depth
94
95      public void countBalancedDepth(int totalNodeCount) { // Ağaç dengeli olsaydı derinlik kaç olurdu metodu
96          System.out.println("Ağaç dengeli olsaydı maksimum " + (int) (Math.floor((Math.log(totalNodeCount)) / (Math.log(2)))) + ". düzeyi olurdu.");
97      }
98
```

## 1.b.2 Ekran Görüntüsü

```
Ağacın Derinliği: 10

Ağacın içindeki Veriler InOrder Şeklinde Sırasıyla:
Milli Park Adı: Abant Gölü Milli Parkı
İl Adı: Bolu
Park Alanı: 1262
Tarih: 10.06.2022
Özellikler: [Abant Gölü, Abant Dağları Üzerinde oluşmuş bir krater ve birikinti gölüdür, Park alanında 1400 metreden 1700 metreye kadar yükseklikte olan birçok tepe vardır, Bitki örtüsü çam, köknar, kayın, meşe, kestane, gürc

Milli Park Adı: Ağrı Dağı Milli Parkı
İl Adı: Ağrı
Park Alanı: 88015
Tarih: 17.11.2004
Özellikler: [Ağrı Dağı Milli Parkı, 2004 yılında, Ağrı dağı ve çevresindeki 88.014 ha alanda ilan edilen milli park, Nuh tufanı, Türkiye'nin en yüksek dağı ve en büyük buzulu, meteor çukuru, yaban hayatın çeşitliliği ile dik

Milli Park Adı: Aladağlar Milli Parkı
İl Adı: Niğde
Park Alanı: 55064
Tarih: 21.04.1995
Özellikler: [Aladağlar, Kayseri-Niğde-Adana illeri arasında bulunan dağ sırası, bitki örtüsü ve hayvan çeşitliliği bakımından zengindir, Bu nedenle dağın 54.524 hektarlık bir bölümü önce Hacer Ormanı Tabiat Parkı, 1995 yılinc

Milli Park Adı: Altınbeşik Mağarası Milli Parkı

Ağacın Düğüm Sayısı: 48
Ağaç dengeli olsaydı maksimum 5. düzeyi olurdu.
```

## 1.b.3 Açıklama

Ağacın derinliğini bulurken recursive olarak her seferinde her bir node için alt tarafındaki derinliği tutan en büyük değişkeni bir arttırıyoruz.

Ağacın içindeki verileri yazdırırken recursive olarak her bir node için önce sol altındaki kısmın bilgisini, sonra kendi bilgisini sonra da sağ altındaki kısmın bilgisini yazdırıyoruz.

Toplam node sayısını bulurken recursive olarak her bir node için sağ ve sol altındaki node sayısını topluyoruz.

Ağaç dengeli olsaydı olacak olan düzey sayısını bulurken logaritmik hesaplama yapıyoruz.

## 1.c Milli Parkın Bulunduğu İli Bulma

### 1.c.1 Kodlar

```
99  /*
100
101  1.C
102
103  */
104  public void searchNode(TreeNode temp, String value) { // ilk üç harfi verilen milli parkı yazdırma
105      if (temp != null) {
106          String str = temp.data.parkAdi.substring(0, 3);
107          if (str.equals(value)) {
108              System.out.println("Parkın bulunduğu il adı: " + temp.data.ilAdi);
109              return;
110          }
111          searchNode(temp.leftChild, value);
112          searchNode(temp.rightChild, value);
113      }
114  }
115  }
116  }
```

### 1.c.2 Ekran görüntüleri

```
Aratmak istediğiniz park adının ilk üç harfini giriniz: Yum
Parkın bulunduğu il adı: Adana
```

## 1.d Bilgi Cümlesindeki Kelimelerin Ayrı Bir İkili Arama Ağacına Yerleştirilmesi

### 1.d.1 Kodlar

#### Main

```
70 //Bölüm 1.d
71 StringTree stringTree = new StringTree();
72 agac.addTreeToStringTreeInOrder(agac.getRoot(), stringTree);
73 System.out.println("");
74 System.out.println("");
75 System.out.println("");
76 System.out.println("Kelime Ağacı İçindeki Veriler InOrder Şeklinde Sırasıyla: ");
77 stringTree.inOrder(stringTree.getRoot());
78 System.out.println("");
79 System.out.println("");
80 System.out.println("");
```

#### Tree Sınıfı

```
118 /*
119 1.D
120 */
121
122 public void addTreeToStringTreeInOrder(TreeNode localroot, StringTree stringTree) { // Ağacı gezerken cümleleri kelimelere ayırıp stringTree ağacına ekleyen metot
123     if (localroot != null) {
124         addTreeToStringTreeInOrder(localroot.leftChild, stringTree);
125         //Ağaca diğer ağacı ekle
126         for (String cumle : localroot.data.cumleler) {
127             String[] kelimeler = cumle.split(regex);
128             for (String kelime : kelimeler) {
129                 stringTree.insert(kelime);
130             }
131         }
132     }
133     addTreeToStringTreeInOrder(localroot.rightChild, stringTree);
134 }
135 }
136 }
137 }
138 }
```

#### StringTree sınıfı

```
Tree.java x StringTree.java x Main.java x
1 public class StringTree { // cümleleri kelimelere ayırıp diğer ağaca eklemek için başka bir class açtık
2
3     class StringTreeNode {
4         public String kelime;
5         public int sayi = 0;
6         public StringTreeNode rightChild;
7         public StringTreeNode leftChild;
8
9         public void displayNode() { System.out.println(kelime + ": " + sayi); } // yazdırma metodu
10    }
11
12    private StringTreeNode root;
13
14    public StringTreeNode getRoot() { return root; }
15
16
17 }
```



```

21 public void insert(String kelime) { // kelimeleri eklemek için metod
22     StringTreeNode newNode = new StringTreeNode();
23     newNode.kelime = kelime;
24
25     if (root == null) {
26         root = newNode;
27     } else {
28         StringTreeNode current = root;
29         StringTreeNode parent;
30         while (true) {
31             parent = current;
32             if (Main.sortStringsWithTurkceKarakterDestegi(Main.trCollator, kelime, current.kelime) < 0) { //kelime < current ise
33                 current = current.leftChild;
34                 if (current == null) {
35                     parent.leftChild = newNode;
36                     parent.leftChild.sayi += 1;
37                     return;
38                 }
39             } else if (Main.sortStringsWithTurkceKarakterDestegi(Main.trCollator, kelime, current.kelime) > 0) { //kelime > current ise
40                 current = current.rightChild;
41                 if (current == null) {
42                     parent.rightChild = newNode;
43                     parent.rightChild.sayi += 1;
44                     return;
45                 }
46             }
47             else { //kelimeler aynı
48                 current.sayi += 1;
49                 return;
50             }
51         }
52     }
53 }
54 }

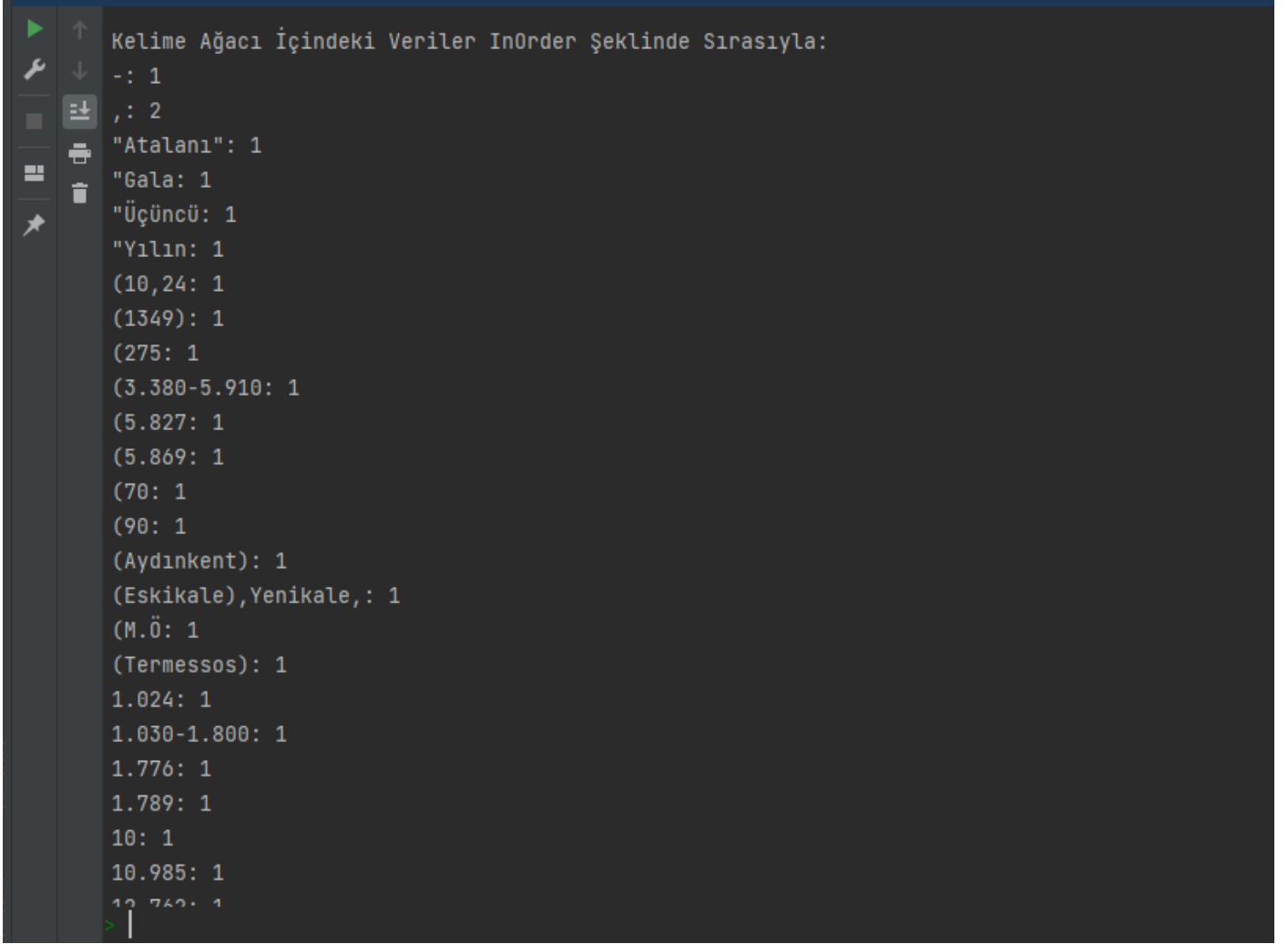
```

```

56 public void inOrder(StringTreeNode localroot) { // kelimeleri yazdırmak için metod
57     if (localroot != null) {
58         inOrder(localroot.leftChild);
59         localroot.displayNode();
60         inOrder(localroot.rightChild);
61     }
62 }
63 }
64 }

```

## 1.d.2 Ekran görüntüleri



```
Kelime Ağacı İçindeki Veriler InOrder Şeklinde Sırasıyla:
-: 1
,: 2
"Atalanı": 1
"Gala: 1
"Üçüncü: 1
"Yılın: 1
(10,24: 1
(1349): 1
(275: 1
(3.380-5.910: 1
(5.827: 1
(5.869: 1
(70: 1
(90: 1
(Aydınkent): 1
(Eskikale),Yenikale,: 1
(M.Ö: 1
(Termessos): 1
1.024: 1
1.030-1.800: 1
1.776: 1
1.789: 1
10: 1
10.985: 1
12.762: 1
```

## 1.d.3 Açıklama

Kelimeleri ve kullanım miktarlarını tutması için StringTree adında bir sınıf oluşturduk. (Tree ile benzer kodlara sahip.)

Tree üstünde recursive olarak dolaşırken TreeNode içindeki cümleleri kelimelere ayırıp StringTree üzerinde bulundukları node'a yerleştiriyoruz. Bu işlem sırasında eğer o kelime daha önceden eklenmişse, kendisine ait node'daki kullanım miktarını tutan değişkeni arttırıyoruz.

## 2) HASH TABLE

### 2.a Milli Parkları Hash Table'a Yerleştirme

#### 2.a.1 Kaynak Kod

```
Hashtable<String, MilliPark> milliParkHashtable = new Hashtable<>();
```

```
//Bölüm 2.a
```

```
milliParkTablosunuDoldurma(milliParkHashtable, milliPark);
```

```
// Bölüm 2.a
```

```
1 usage
```

```
public static void milliParkTablosunuDoldurma(Hashtable<String, MilliPark> hashtable, MilliPark park) {  
    hashtable.put(park.parkAdi, park);  
}
```

#### 2.a.2 Ekran Görüntüsü

```
Milli Parkların Bulunduğu HashTable Verisi  
{Saklıkent Milli Parkı=Milli Park Adı: Saklıkent Milli Parkı  
İl Adı: Muğla  
Park Alanı: 1643  
Tarih: 6.06.1996  
Özellikler: [Saklıkent Milli Parkı veya yaygın adıyla Saklıkent Kanyonu, Antalya-Muğla sınırını çizen Egen Çayı'nın kolu olan Karaçay'ın oluşturduğu kanyondur, Muğla'nın Seydikem  
, Tek Tek Dağları Milli Parkı=Milli Park Adı: Tek Tek Dağları Milli Parkı  
İl Adı: Şanlıurfa  
Park Alanı: 19335  
Tarih: 29.05.2007  
Özellikler: [Tek Tek Dağları Milli Parkı, Şanlıurfa ilinde 19.335 ha alanda 2007 yılında ilan edilen milli park, Tektek Dağların üzerinde yer alır, Park alanı; Şuayip Şehri ve So  
, Aladağlar Milli Parkı=Milli Park Adı: Aladağlar Milli Parkı  
İl Adı: Niğde  
Park Alanı: 55064  
Tarih: 21.04.1995  
Özellikler: [Aladağlar, Kayseri-Niğde-Adana illeri arasında bulunan dağ sırası, bitki örtüsü ve hayvan çeşitliliği bakımından zengindir, Bu nedenle dağın 54.524 hektarlık bir böl  
, Malazgirt Meydan Muharebesi Tarihi Milli Parkı=Milli Park Adı: Malazgirt Meydan Muharebesi Tarihi Milli Parkı  
İl Adı: Muş  
Park Alanı: 238  
Tarih: 17.03.2018  
Özellikler: [2018/11366 sayılı Bakanlar Kurulu kararı ile 5 Şubat 2018 tarihinde ilân edilen milli park, Malazgirt Meydan Muharebesi'nin gerçekleştiği alan üzerinde kurulu bulunm  
, Soğuksu Milli Parkı=Milli Park Adı: Soğuksu Milli Parkı  
İl Adı: Ankara  
Park Alanı: 1186  
Tarih: 19.02.1959  
Özellikler: [Soğuksu Milli Parkı, Ankara'nın Kızılcahamam ilçesinde yer alan ve 19 Şubat 1959'da kurulan bir milli parktır, Ankara şehir merkezine 80 kilometre uzaklıktadır, Park  
, Kop Dağı Müdafaaası Tarihi Milli Parkı=Milli Park Adı: Kop Dağı Müdafaaası Tarihi Milli Parkı  
İl Adı: Bayburt  
İl Adı: Bayburt  
Park Alanı: 6335  
Tarih: 15.11.2016  
Özellikler: [2016/9486 sayılı Bakanlar Kurulu kararı ile 31 Ekim 2016 tarihinde milli park olarak ilân edilen alan I, Dünya Savaşı sırasında Ruslara karşı Kop Savunması olarak b  
, Hakkari Cilo ve Sat Dağları Milli Parkı=Milli Park Adı: Hakkari Cilo ve Sat Dağları Milli Parkı  
İl Adı: Hakkari  
Park Alanı: 27500  
Tarih: 26.09.2020  
Özellikler: [Hakkari Cilo ve Sat Dağları Milli Parkı, Türkiye'nin Hakkari ilini Merkez, Şemdinli ve Yüksekova ilçeleri sınırları içinde yer alan bir milli parktır, 25 Eylül 202  
, Munzur Vadisi Milli Parkı=Milli Park Adı: Munzur Vadisi Milli Parkı  
İl Adı: Tunceli  
Park Alanı: 42674  
Tarih: 21.12.1971  
Özellikler: [Munzur Vadisi Milli Parkı, Tunceli ve Ovacık arasında uzanan Munzur Vadisi'nde, 42.000 hektarlık bir alandır, 1971 yılında milli park olarak ilan edilmiştir, Bu böl  
, İstiklal Yolu Tarihi Milli Parkı=Milli Park Adı: İstiklal Yolu Tarihi Milli Parkı  
İl Adı: Kastamonu  
Park Alanı: 236  
Tarih: 2.11.2018  
Özellikler: [İstiklal Yolu, İnebolu sahilinden başlayıp Kastamonu ve Çankırı üzerinden Ankara'ya uzanan, Türk Kurtuluş Savaşı boyunca İnebolu'ya deniz yoluyla gelen cephanenin ka  
, İğneada Longoz Ormanları Milli Parkı=Milli Park Adı: İğneada Longoz Ormanları Milli Parkı  
İl Adı: Kırklareli  
Park Alanı: 3155  
Tarih: 13.11.2007  
Özellikler: [İğneada Longoz Ormanları Milli Parkı, Marmara bölgesi, Kırklareli ili, Demirköy ilçesinde, 3.155 ha alanda, 2007 yılında kurulan milli park.]  
, Köprülü Kanyon Milli Parkı=Milli Park Adı: Köprülü Kanyon Milli Parkı  
İl Adı: Antalya  
Park Alanı: 47473  
Tarih: 12.12.1973
```

Özellikler: [Alanda doğal güzelliklerin yanında tarihi unsurlarda yer alır: Selge antik şehir kalıntılarında tiyatro, agora, Artemis ve Zeus tapınakları, sarnıçlar ve su kemerleri, Sarıçalı Dağı Milli Parkı=Milli Park Adı: Sarıçalı Dağı Milli Parkı  
İl Adı: Ankara  
Park Alanı: 1024  
Tarih: 28.10.2021  
Özellikler: [Sarıçalı Dağı Milli Parkı, Türkiye'nin Ankara ilinin Nallıhan ilçesinde yer alan bir millî parktır, 28 Ekim 2021 tarihinde kurulmuştur, 1.024 hektar (10,24 km2) alanı kapsar, Nene Hatun Tarihi Milli Parkı=Milli Park Adı: Nene Hatun Tarihi Milli Parkı  
İl Adı: Erzurum  
Park Alanı: 387  
Tarih: 6.06.2009  
Özellikler: [Aziziye Tabyasında gösterdiği kahramanlıkla adını tarihe yazdıran Kahraman Türk Kadını Nene Hatun bir cesaret sembolü olarak tanınmış ve anılmıştır, Ömrünün son demlerini burada geçirmiştir, Kaçkar Dağları Milli Parkı=Milli Park Adı: Kaçkar Dağları Milli Parkı  
İl Adı: Rize  
Park Alanı: 52970  
Tarih: 31.08.1994  
Özellikler: [Kaçkar Dağları Milli Parkı büyük bölümü Rizenin Çamlıhemşin ilçesinde, bir bölümü Erzurum ve Artvin illerine uzanan millî park, 51.550 hektarlık millî park Fırtına Deresi, Gölük Dağı - Termessos Milli Parkı=Milli Park Adı: Gölük Dağı - Termessos Milli Parkı  
İl Adı: Antalya  
Park Alanı: 6700  
Tarih: 3.11.1970  
Özellikler: [Gölük Dağı (Termessos) Milli Parkı, Antalya ilinde 1970 yılında, 6.702 hektarlık alan üzerinde kurulan millî park, Toros dağlarının eteklerinde, Korkuteli ilçe sınırlarında, Nemrut Dağı Milli Parkı=Milli Park Adı: Nemrut Dağı Milli Parkı  
İl Adı: Adıyaman  
Park Alanı: 13827  
Tarih: 7.12.1988  
Özellikler: [Antiochos'un tümölüsü ve dev heykelleri, Arsameia (Eskikale), Yenikale, Karakuş Tepe ve Cendere Köprüsü Milli Park içerisinde kalan kültürel değerlerdir, Eski çağlardan itibaren bitki ve hayvanatçıların ilgisini çeken Botan Vadisi Milli Parkı=Milli Park Adı: Botan Vadisi Milli Parkı  
İl Adı: Siirt

İl Adı: Siirt  
Park Alanı: 11358  
Tarih: 15.08.2019  
Özellikler: [Botan Vadisi, Türkiye'nin Siirt ilini sınırları içerisinde bulunan millî park statüsündeki tarihi alan, 1421 sayılı cumhurbaşkanlığı kararının 30859 sayılı T.C. Resmi Gazete'de yayımlanmasıyla, Uludağ Milli Parkı=Milli Park Adı: Uludağ Milli Parkı  
İl Adı: Bursa  
Park Alanı: 13024  
Tarih: 20.09.1961  
Özellikler: [Uludağ Milli Parkı 1961 yılında Millî Park olarak ilan edildi, 1961 yılında koruma altına alınan alanı 12.762 hektar'dır, Daha sonra millî park alanı 27.300 hektara çıkarılmıştır, Dilek Y. - B. Menderes D. Milli Parkı=Milli Park Adı: Dilek Y. - B. Menderes D. Milli Parkı  
İl Adı: Aydın  
Park Alanı: 27598  
Tarih: 19.05.1966  
Özellikler: [Dilek Yarımadası - Büyük Menderes Deltası Milli Parkı, Aydın il sınırları içinde Dilek Dağı'nın Ege Denizi'ne uzandığı son noktada yer alan millî park, 27.675 hektarlık alanı kapsar, Altındere Vadisi Milli Parkı=Milli Park Adı: Altındere Vadisi Milli Parkı  
İl Adı: Trabzon  
Park Alanı: 4468  
Tarih: 9.09.1987  
Özellikler: [Altındere Milli Parkı, Trabzon'un güneyinde Maçka yakınlarındadır, Bu millî parkın içinde, neredeyse bulutların bile üzerinde yer alan muhteşem Sümela Manastırı bulundurmaktadır, Yozgat Çamlığı Milli Parkı=Milli Park Adı: Yozgat Çamlığı Milli Parkı  
İl Adı: Yozgat  
Park Alanı: 517  
Tarih: 31.12.2001  
Özellikler: [Yozgat Çamlığı Milli Parkı, İç Anadolu Bölgesi'nde insan etkisi ile meydana gelen step içerisinde yer alan sayılı kalıntı ormanlardan biridir, Türkiye'nin ilk millî parkıdır, Abant Gölü Milli Parkı=Milli Park Adı: Abant Gölü Milli Parkı  
İl Adı: Bolu  
Park Alanı: 1262  
Tarih: 10.06.2022

Park Alanı: 1262  
Tarih: 10.06.2022  
Özellikler: [Abant Gölü, Abant Dağları üzerinde oluşmuş bir krater ve birikinti gölüdür, Park alanında 1400 metreden 1700 metreye kadar yükseklikte olan birçok tepe vardır, Bitki ve hayvanatçıların ilgisini çeken Kuşçenneti Milli Parkı=Milli Park Adı: Kuşçenneti Milli Parkı  
İl Adı: Balıkesir  
Park Alanı: 17058  
Tarih: 27.07.1959  
Özellikler: [Milletlerarası düzeyde önem taşıyan millî parktaki kuş zenginliği ve millî park tanımı içindeki başarılı koruma uygulaması nedeniyle, 1975 yılında Avrupa Konseyi'nce koruma altına alınmıştır, Honaz Dağı Milli Parkı=Milli Park Adı: Honaz Dağı Milli Parkı  
İl Adı: Denizli  
Park Alanı: 9429  
Tarih: 21.04.1998  
Özellikler: [Honaz Dağı, Türkiye'nin Denizli ilinin güneydoğusu boyunca, Ege ve Akdeniz bölgelerini ayıran doğal sınır olarak uzanan bir dağdır.]  
Karatepe - Aslantaş Milli Parkı=Milli Park Adı: Karatepe - Aslantaş Milli Parkı  
İl Adı: Osmaniye  
Park Alanı: 4143  
Tarih: 29.05.1958  
Özellikler: [Karatepe-Aslantaş Milli Parkı, 1958 yılında Akdeniz Bölgesi'nde, Osmaniye ilinin Kadirli ilçesine 22 km uzaklıkta ve Ceyhan Nehrinin kenarında kurulmuştur.]  
Troia Tarihi Milli Parkı=Milli Park Adı: Troia Tarihi Milli Parkı  
İl Adı: Çanakkale  
Park Alanı: 13517  
Tarih: 7.11.1996  
Özellikler: [Troia Tarihi millî parkının ana değeri, Dünya miras listesinde yer alan Troia antik kenti ve arkeolojik zenginlikleridir, Bu alan Homeros'un destanlarında anlattığı Troya Savaşı'nın geçtiği yerdir, Altınbeşik Mağarası Milli Parkı=Milli Park Adı: Altınbeşik Mağarası Milli Parkı  
İl Adı: Antalya  
Park Alanı: 1147  
Tarih: 31.08.1994  
Özellikler: [Altınbeşik Mağarası Milli Parkı, Antalya ilinde bulunan millî park, İbradi (Aydıncık) ilçesine 9.7 km uzaklıkta olan Ürünü köyünün yaklaşık 5 km güneydoğusundadır,

İl Adı: Konya  
Park Alanı: 1147  
Tarih: 7.06.2022  
Özellikler: [Derebucak Çamlık Mağaraları Milli Parkı, Türkiye'nin Konya ilinin Derebucak ilçesinde yer alan bir milli parktır.]  
, Spil Dağı Milli Parkı=Milli Park Adı: Spil Dağı Milli Parkı  
İl Adı: Manisa  
Park Alanı: 6801  
Tarih: 22.04.1968  
Özellikler: [Spil Dağı Milli Parkı, Türkiye'nin Manisa ve İzmir il sınırları içerisinde yer alan bir milli parktır, Spil Dağı'nı ve çevresini kapsayan park, 22 Nisan 1968'de ilan  
, Ilgaz Dağı Milli Parkı=Milli Park Adı: Ilgaz Dağı Milli Parkı  
İl Adı: Kastamonu  
Park Alanı: 1118  
Tarih: 2.06.1976  
Özellikler: [Ilgaz Dağı Milli Parkı, Batı Karadeniz bölümünde Kastamonu ve Çankırı illerinde, Ilgaz Dağları üzerinde 1976 yılında kurulmuş milli park.]  
, Beydağları Sahil Milli Parkı=Milli Park Adı: Beydağları Sahil Milli Parkı  
İl Adı: Antalya  
Park Alanı: 31166  
Tarih: 16.03.1972  
Özellikler: [Olimpos Beydağları Milli Parkı, 1972 tarihinde Antalya ili Kemer ilçesi sınırları içinde bulunan doğal ve tarihi güzelliklerin korunması için sit alanı olarak korunma  
, Sultan Sazlığı Milli Parkı=Milli Park Adı: Sultan Sazlığı Milli Parkı  
İl Adı: Kayseri  
Park Alanı: 24358  
Tarih: 17.03.2006  
Özellikler: [Alanı besleyen başlıca akarsular Yahyalı, Yeşilhisar ve Dündarlı dereleri ile Develi çayı ve Ağaçar yakınlarından çıkan su sayılabilir, ilkbaharda yağışların artması  
, Sarıkamış-Allahuekber Dağları Milli Parkı=Milli Park Adı: Sarıkamış-Allahuekber Dağları Milli Parkı  
İl Adı: Kars  
Park Alanı: 22520  
Tarih: 19.10.2004  
Özellikler: [Park; Sarıkamış, Şenkaya ve Selim ilçeleri sınırlarında bulunur, Park alanının birinci kaynak değeri Alpin çayırlar kuşağında yayılış gösteren saf sarıçam ormanlarıdır  
, Beyşehir Gölü Milli Parkı=Milli Park Adı: Beyşehir Gölü Milli Parkı  
İl Adı: Konya  
Park Alanı: 82157  
Tarih: 11.01.1993  
Özellikler: [Beyşehir Gölü Milli Parkı, Beyşehir Gölü'nün yakınında Türkiye'nin en büyük milli parkıdır, Derlenen bilgilere göre, 1993 yılında Bakanlar Kurulu kararıyla kurulan m  
, Ağrı Dağı Milli Parkı=Milli Park Adı: Ağrı Dağı Milli Parkı  
İl Adı: Ağrı  
Park Alanı: 88015  
Tarih: 17.11.2004  
Özellikler: [Ağrı Dağı Milli Parkı, 2004 yılında, Ağrı dağı ve çevresindeki 88.014 ha alanda ilan edilen milli park, Nuh tufanı, Türkiye'nin en yüksek dağı ve en büyük buzulu, me  
, Karagöl - Sahara Milli Parkı=Milli Park Adı: Karagöl - Sahara Milli Parkı  
İl Adı: Artvin  
Park Alanı: 3251  
Tarih: 31.08.1994  
Özellikler: [Karagöl-Sahara Milli Parkı, Türkiye'deki milli park alanlarından birisidir, Artvin'in Şavşat ilçesi sınırları içerisinde yer alan park, Karagöl ve Sahara Yaylası olma  
, Yedigöller Milli Parkı=Milli Park Adı: Yedigöller Milli Parkı  
İl Adı: Bolu  
Park Alanı: 1623  
Tarih: 29.04.1965  
Özellikler: [Milli parkta hakim bitki örtüsü kayın ağaçlarıdır, Ayrıca meşe, gürgen, kızılçam, karaçam, sarıçam, köknar, karaağaç, ıhlamur ve porsuk gibi değişik tür ağaçlar da  
, Başkomutan Tarihi Milli Parkı=Milli Park Adı: Başkomutan Tarihi Milli Parkı  
İl Adı: Afyonkarahisar  
Park Alanı: 34834  
Tarih: 8.11.1981  
Özellikler: [Başkomutan Tarihi Milli Parkı, Türkiye'nin İç Anadolu Bölgesi'nde tarihi ve kültürel değerlerin korunması amacıyla , 8 Kasım 1981'de ilan edilen milli park, Millî Mü  
, Hatila Vadisi Milli Parkı=Milli Park Adı: Hatila Vadisi Milli Parkı  
İl Adı: Artvin  
Park Alanı: 16944  
Tarih: 31.08.1994  
Özellikler: [Hatila Vadisi Milli Parkı, Artvin merkez ilçe sınırları içerisinde Hatila Vadisindeki Hatila Deresi ve birçok yan derelerini içerir, Ulaşım, Artvin il merkezinden 10  
, Marmaris Milli Parkı=Milli Park Adı: Marmaris Milli Parkı  
İl Adı: Muğla  
Park Alanı: 29286  
Tarih: 8.03.1996  
Özellikler: [Alanda bulunan 541 bitkiden, 54 endemik, 9 tür alana özgü endemik, 5 tür tehlike altında, 74 tür nadir bitki sınıfındadır, Endemik Sığla ağacı milli parkın kıyıların  
, Kızıldağ Milli Parkı=Milli Park Adı: Kızıldağ Milli Parkı  
İl Adı: Isparta  
Park Alanı: 80200  
Tarih: 9.05.1969  
Özellikler: [Kızıldağ Milli Parkı, öncelikle bir dağdır, Karaçamıyla, sedir ağaçlarıyla, meşe ve ardıç ağaçlarıyla örtülüdür, Mavi yapraklı sedir ormanının bulunduğu tek bölgedir  
, Gala Gölü Milli Parkı=Milli Park Adı: Gala Gölü Milli Parkı  
İl Adı: Edirne  
Park Alanı: 5923  
Tarih: 5.03.2005  
Özellikler: [Gala Gölü Milli Parkı, Enez ve İpsala ilçeleri arasında bulunan Gala ve Pamuklu göllerini kapsayan alan 5 Mart 2005 tarihli Resmî Gazete'de yayınlanan 2005/8547 sayılı  
, Sakarya Meydan Muharebesi Tarihi Milli Parkı=Milli Park Adı: Sakarya Meydan Muharebesi Tarihi Milli Parkı  
İl Adı: Ankara  
Park Alanı: 13850  
Tarih: 8.02.2015  
Özellikler: [Sakarya Meydan Muharebesi, Kurtuluş Savaşı'nın dönüm noktası olarak görülmektedir, Savaşın gerçekleştiği alandaki siperler, mevziler, şehitlikler, savaşta kullanılan  
, Yumurtalık Lagünü Milli Parkı=Milli Park Adı: Yumurtalık Lagünü Milli Parkı

```

, Yumurtalık Lagünü Milli Parkı=Milli Park Adı: Yumurtalık Lagünü Milli Parkı
İl Adı: Adana
Park Alanı: 16980
Tarih: 6.12.2008
Özellikler: [Lagün, toplam 16.430 hektar alana sahiptir, Ulaşım Adana-Karataş-Yumurtalık yolu (70 km) veya Adana-Ceyhan-Yumurtalık yolu (90 km) üzerinden sağlanmaktadır.]
, Kovada Gölü Milli Parkı=Milli Park Adı: Kovada Gölü Milli Parkı
İl Adı: Isparta
Park Alanı: 6551
Tarih: 3.11.1970
Özellikler: [Kovada Gölü Milli Parkı, Isparta'nın Eğirdir ilçesi sınırları içerisinde yer alır, Milli park sâhası içerisinde konaklama ve piknik yapma amaçlı hazırlanmış yerler m
, Kazdağı Milli Parkı=Milli Park Adı: Kazdağı Milli Parkı
İl Adı: Balıkesir
Park Alanı: 20935
Tarih: 17.04.1994
Özellikler: [Milli park sınırları içinde yer alan Pınarbaşı ve Hasanboğuldu piknik alanları başlıca gezi noktalarıdır, Ayrıca; Mehmetalan Köyü, milli park sınırları içinde kalır,
, Küre Dağları Milli Parkı=Milli Park Adı: Küre Dağları Milli Parkı
İl Adı: Kastamonu
Park Alanı: 37753
Tarih: 7.07.2000
Özellikler: [Çeşitli faktörlerle giderek tahrip olan Küre Dağları ve üzerindeki ormanların batı bölümü bitkisel çeşitlilik yönünden çok zengindir, Milletlerarası kuruluşların Kar
, Boğazköy - Alacahöyük Milli Parkı=Milli Park Adı: Boğazköy - Alacahöyük Milli Parkı
İl Adı: Çorum
Park Alanı: 2600
Tarih: 21.09.1988
Özellikler: [Anadolu'da kurulan en eski uygarlıklardan biri olan Hititlerden kalma arkeolojik değerler barındırır, Hitit devletinin başkenti Hattuşaş'a ait; kent surları, yer kap
}

```

## 2.b Hash Table Güncelleme

### 2.b.1 Kaynak Kod

```

// Bölüm 2.b
1 usage
public static void milliParkTarihDegistirme(Hashtable<String, MilliPark> milliParkHashtable) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("İlan tarihini değiştirmek istediğiniz park adını giriniz: ");
    String parkName = scanner.nextLine();

    if (!milliParkHashtable.containsKey(parkName)) {
        System.out.println("Park bulunamadı.");
        return;
    }

    MilliPark park = milliParkHashtable.get(parkName);

    System.out.print("Yeni ilan tarihini girin: ");
    park.tarih = scanner.nextLine();
    System.out.println("Park güncellendi: " + park);
}

```

### 2.b.2 Açıklama

Öncelikle kullanıcıdan tarihini değiştirmek istediği park ismini alıyoruz. Scanner ile aldığımız tarih değeri ile milli parkın tarih değerini değiştiriyoruz.

### 2.b.3 Ekran Görüntüsü

```

İlan tarihini değiştirmek istediğiniz park adını giriniz: Yozgat Çamlığı Milli Parkı
Yeni ilan tarihini girin: 31.12.2001
Park güncellendi: Milli Park Adı: Yozgat Çamlığı Milli Parkı
İl Adı: Yozgat
Park Alanı: 517
Tarih: 31.12.2001
Özellikler: [Yozgat Çamlığı Milli Parkı, İç Anadolu Bölgesi'nde insan etkisi ile meydana gelen step içerisinde yer alan sayılı kalıntı ormanlardan biridir, Türkiye'nin ilk milli

```

## 3) HEAP

### 3.a Heap

#### 3.a.1 Kaynak Kod

```
// heap sınıfı için node sınıfı
class HeapNode {
    3 usages
    private int heapdata;

    1 usage
    public HeapNode(int heapdata) { this.heapdata = heapdata; }

    9 usages
    public int getHeapdata() { return heapdata; }

    1 usage
    public void setHeapdata(int heapdata) { this.heapdata = heapdata; }
}

// Heap sınıfı başlangıcı
2 usages
public class Heap {
    22 usages
    private HeapNode[] heapArray;
    4 usages
    private int maxSize;
    11 usages
    private int currentSize;

    1 usage
    public Heap(int mx) // constructor
    {
```

```
    1 usage
    public Heap(int mx) // constructor
    {
        maxSize = mx;
        currentSize = 0;
        heapArray = new HeapNode[maxSize]; // create array
    }

    public boolean isEmpty() { return currentSize == 0; }

    10 usages
    public boolean insert(int key) { // heap sınıfı için insert metodu
        if (currentSize == maxSize) {
            return false;
        }

        HeapNode newNode = new HeapNode(key);
        heapArray[currentSize] = newNode;
        trickleUp(currentSize++);
        //displayHeap(); // Her eleman eklenmesinde ağacı gösteriyoruz.
        return true;
    }
```

```

2 usages
public void trickleUp(int index) {
    int parent = (index - 1) / 2;
    HeapNode bottom = heapArray[index];
    while (index > 0 && heapArray[parent].getHeapdata() < bottom.getHeapdata()) {
        heapArray[index] = heapArray[parent]; //Yukarı taşı
        index = parent;
        parent = (parent - 1) / 2;
    }
    heapArray[index] = bottom;
}

1 usage
public HeapNode remove() { // heap sınıfı için remove metodu
    HeapNode root = heapArray[0];
    heapArray[0] = heapArray[--currentSize];
    trickleDown(index: 0);
    return root;
}

```

```

2 usages
public void trickleDown(int index) {
    int largerChild;
    HeapNode top = heapArray[index];
    while (index < currentSize / 2) {
        int leftChild = 2 * index + 1;
        int rightChild = leftChild + 1;

        if (rightChild < currentSize && heapArray[leftChild].getHeapdata() < heapArray[rightChild].getHeapdata()) {
            largerChild = rightChild;
        } else {
            largerChild = leftChild;
        }

        if (top.getHeapdata() >= heapArray[largerChild].getHeapdata()) {
            break;
        }

        heapArray[index] = heapArray[largerChild];
        index = largerChild;
    }
    heapArray[index] = top;
}

```

```

public boolean change(int index, int newValue) { // indexi verilen değerlin verilen degerle yer deđiřtirmesi metodu
    if (index < 0) {
        return false;
    }
    int oldValue = heapArray[index].getHeapdata();
    heapArray[index].setHeapdata(newValue);
    if (oldValue < newValue) {
        trickleUp(index);
    }
    else {
        trickleDown(index);
    }
    return true;
}

```



```

public void displayHeap(){ // heapi yazdırmak için metod
    System.out.print("heapArray: ");
    for (int m = 0; m < currentSize; m++)
    {
        if(heapArray[m] != null){
            System.out.print(heapArray[m].getHeapdata() + " ");
        }
        else{
            System.out.print("-- ");
        }
    }
    System.out.println();

    int nBlanks = maxSize+1;
    int itemsPerRow = 1;
    int column = 0;
    int j = 0;
    String dots = ".....";
    System.out.println(dots + dots);

    while (currentSize > 0)
    {
        if (column == 0)
        {
            for(int k = 0; k < nBlanks; k++){
                System.out.print(' ');
            }

```

```

        }
    }
    System.out.print(heapArray[j].getHeapdata());
    if (++j == currentSize)
    {
        break;
    }

    if (++column == itemsPerRow){
        nBlanks /= 2;
        itemsPerRow *= 2;
        column = 0;
        System.out.println();
    }
    else {
        for(int k = 0; k < nBlanks*2 - 2; k++)
        {
            System.out.print(' ');
        }
    }
}
System.out.println("\n"+dots+dots);
}

```

### 3.b Max Heap

#### 3.b.1 Kaynak Kod

```
1 // node sınıfı
2 class MilliParkHeapNode {
3     private MilliPark heapdata;
4
5     public MilliParkHeapNode(MilliPark heapdata) { this.heapdata = heapdata; }
6
7
8     public MilliPark getHeapdata() { return heapdata; }
9
10
11     public void setHeapdata(MilliPark heapdata) { this.heapdata = heapdata; }
12 }
13
14 // Milli park heap sınıfı başlangıcı
15 public class MilliParkHeap {
16     private MilliParkHeapNode[] heapArray;
17     private int maxSize;
18     private int currentSize;
19
20     public MilliParkHeap(int mx) // constructor
21     {
22         maxSize = mx;
23         currentSize = 0;
24         heapArray = new MilliParkHeapNode[maxSize]; // create array
25     }
26
27     public boolean isEmpty() { return currentSize == 0; }
28
29     public boolean insert(MilliPark key) { // insert metodu
30         if (currentSize == maxSize) {
31             return false;
32         }
33
34         MilliParkHeapNode newNode = new MilliParkHeapNode(key);
35         heapArray[currentSize] = newNode;
36         trickleUp(currentSize++);
37         return true;
38     }
39
40 }
```

```
41 public void trickleUp(int index) {
42     int parent = (index - 1) / 2;
43     MilliParkHeapNode bottom = heapArray[index];
44     while (index > 0 && heapArray[parent].getHeapdata().alan < bottom.getHeapdata().alan) {
45         heapArray[index] = heapArray[parent]; //Yukarı taşı
46         index = parent;
47         parent = (parent - 1) / 2;
48     }
49     heapArray[index] = bottom;
50 }
51
52 public MilliParkHeapNode remove() { // remove metodu
53     MilliParkHeapNode root = heapArray[0];
54     heapArray[0] = heapArray[--currentSize];
55     trickleDown(index: 0);
56     return root;
57 }
58 }
```

```

67     public void trickleDown(int index) {
68         int largerChild;
69         MilliParkHeapNode top = heapArray[index];
70         while (index < currentSize / 2) {
71             int leftChild = 2 * index + 1;
72             int rightChild = leftChild + 1;
73
74             if (rightChild < currentSize && heapArray[leftChild].getHeapdata().alan < heapArray[rightChild].getHeapdata().alan) {
75                 largerChild = rightChild;
76             } else {
77                 largerChild = leftChild;
78             }
79
80             if (top.getHeapdata().alan >= heapArray[largerChild].getHeapdata().alan) {
81                 break;
82             }
83
84             heapArray[index] = heapArray[largerChild];
85             index = largerChild;
86         }
87         heapArray[index] = top;
88     }

```

```

90     public boolean change(int index, MilliPark newValue) {
91         if (index < 0) {
92             return false;
93         }
94         int oldValue = heapArray[index].getHeapdata().alan;
95         heapArray[index].setHeapdata(newValue);
96         if (oldValue < newValue.alan) {
97             trickleUp(index);
98         } else {
99             trickleDown(index);
100         }
101         return true;
102     }

```

```

104 public void displayHeap() {
105     System.out.print("MilliParkHeapArray: ");
106     for (int m = 0; m < currentSize; m++) {
107         if (heapArray[m] != null) {
108             System.out.println(heapArray[m].getHeapdata());
109         }
110     }
111     System.out.println();
112
113     int nBlanks = maxSize + 1;
114     int itemsPerRow = 1;
115     int column = 0;
116     int j = 0;
117     String dots = ".....";
118     System.out.println(dots + dots);
119
120     while (currentSize > 0) {
121         if (column == 0) {
122             for (int k = 0; k < nBlanks; k++) {
123                 System.out.print(' ');
124             }
125         }
126         System.out.print(heapArray[j].getHeapdata().parkAdi);
127         if (++j == currentSize) {
128             break;
129         }
130
131         if (++column == itemsPerRow) {
132             nBlanks /= 2;
133             itemsPerRow *= 2;
134             column = 0;
135             System.out.println();
136         } else {
137             for (int k = 0; k < nBlanks * 2 - 2; k++) {
138                 System.out.print(' ');
139             }
140         }
141     }
142
143     System.out.println("\n" + dots + dots);
144 }

```

Main

```

19 MilliParkHeap yuzOlcumuHeap = new MilliParkHeap(max: 48);

```

```

41
42 //Bölüm 3.b
43 yuzOlcumuHeap.insert(milliPark);

```

### 3.c Max Heap Listeleme

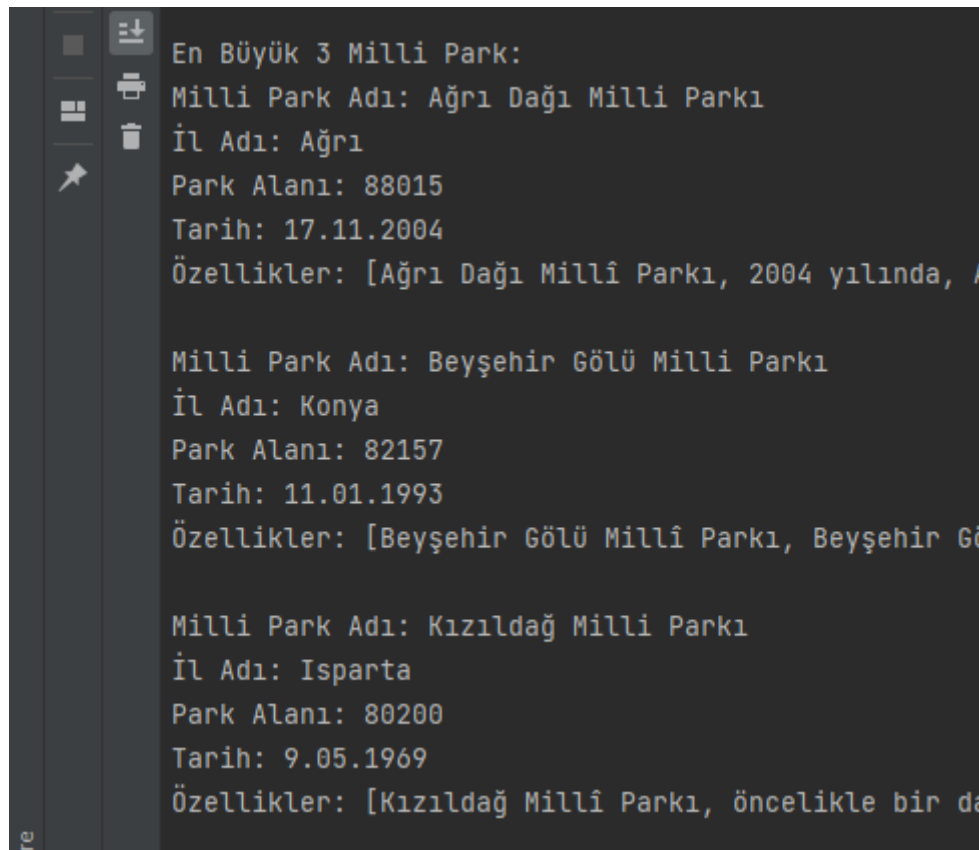
#### 3.c.1 Kaynak Kod

Main

```
109 //Bölüm 3.c
110 heaptEnBuyukUcParkiSil(yuzOlcumuHeap);
```

```
141 // Bölüm 3.c
142 public static void heaptEnBuyukUcParkiSil(MilliParkHeap milliParkHeap) {
143     System.out.println("En Büyük 3 Milli Park:");
144
145     for (int i = 0; i < 3; i++) {
146         System.out.println(milliParkHeap.remove().getHeapdata());
147     }
148 }
```

#### 3.c.2 Ekran Görüntüsü



```
En Büyük 3 Milli Park:
Milli Park Adı: Ağrı Dağı Milli Parkı
İl Adı: Ağrı
Park Alanı: 88015
Tarih: 17.11.2004
Özellikler: [Ağrı Dağı Millî Parkı, 2004 yılında, A

Milli Park Adı: Beyşehir Gölü Milli Parkı
İl Adı: Konya
Park Alanı: 82157
Tarih: 11.01.1993
Özellikler: [Beyşehir Gölü Millî Parkı, Beyşehir Gölü

Milli Park Adı: Kızıldağ Milli Parkı
İl Adı: Isparta
Park Alanı: 80200
Tarih: 9.05.1969
Özellikler: [Kızıldağ Millî Parkı, öncelikle bir da
```

#### 3.c.3 Açıklama

Heap bir max heap yapısında. Node'lar, veri olarak MilliPark sınıfından oluşturulmuş obje tutuyor.

Heap içine milli parkı yerleştirirken main metodu içindeki dosya okuma kısmında oluşturulan

MilliPark objelerini kullanıyoruz. Aynı referansları heap içine atıyoruz.

En büyük 3 taneyi sıralarken ise 3 defa silme işlemi yapıyoruz. Her bir silme işleminde heap içinde yüz ölçümü en büyük park döndüğünden sırasıyla en büyük 3 parkı elde ediyoruz.

## 4) SIRALAMA

### 4.a Sıralama Algoritması

#### 4.a.1 Kaynak Kod

```
1 usage
2
3 public static void selectionSort(int[] dizi) {
4     System.out.println("SelectionSort - Dizinin En Bastaki Hali:");
5     printArray(dizi);
6
7     int sayiMiktari = dizi.length;
8     for (int i = 0; i < sayiMiktari - 1; i++) {
9
10        System.out.println();
11        System.out.println("Tur " + i + " Sonu");
12
13        int enKucukSayininIndeksi = i;
14
15        for (int j = i + 1; j < sayiMiktari; j++) {
16            if (dizi[j] < dizi[enKucukSayininIndeksi]) {
17                enKucukSayininIndeksi = j;
18            }
19        }
20
21        int geciciDegisken = dizi[enKucukSayininIndeksi];
22        dizi[enKucukSayininIndeksi] = dizi[i];
23        dizi[i] = geciciDegisken;
24        printArray(dizi);
25    }
26
27    System.out.println();
28    System.out.println("Son Hal");
29    printArray(dizi);
30 }
}
```

#### 4.a.2 Açıklama

Selection sort algoritması ilk başta dizinin ilk elemanını tutar. Ondan sonra diğer elemanlar ile teker teker karşılaştırır. Bulduğu en küçük eleman ile yerlerini değiştirir. Ondan sonra dizinin ikinci elemanı ile 2. elemandan sonraki elemanları karşılaştırır. Bulduğu en küçük eleman ile 2. elemanın yerini değiştirir. Sırayla tüm elemanlar diziyi böyle dolaşır ve sıralı bir dizi elde edilmiş olur.

### 4.b Zaman Karmaşıklığı

#### 4.b.1 Açıklama

İç içe iki tane for döngüsü olduğu için Selection sort algoritmasının zaman karmaşıklığı  $O(n^2)$  dir.

Karşılaştırma olarak da bubble sort seçtik.

İkisinde de aynı miktarda karşılaştırma yapılıyor.  $[n * (n-1) / 2 \text{ adet}]$ . Ama selection sort sürecinde daha az yer değişimi yapıldığı için selection sort, bubble sorttan şüphesiz daha hızlı.

```

public static void selectionSort(int[] dizi) { dizi: [11, 25, 12, 22, 64]
    System.out.println("SelectionSort - Dizinin En Bastaki Hali:");
    printArray(dizi);

    int sayiMiktari = dizi.length; sayiMiktari: 5
    for (int i = 0; i < sayiMiktari - 1; i++) { i: 0

        System.out.println();
        System.out.println("Tur " + i + " Sonu");

        int enKucukSayininIndeksi = i; enKucukSayininIndeksi: 4

        for (int j = i + 1; j < sayiMiktari; j++) { sayiMiktari: 5
            if (dizi[j] < dizi[enKucukSayininIndeksi]) {
                enKucukSayininIndeksi = j;
            }
        }

        int geciciDegisken = dizi[enKucukSayininIndeksi]; geciciDegisken: 11
        dizi[enKucukSayininIndeksi] = dizi[i]; enKucukSayininIndeksi: 4
        dizi[i] = geciciDegisken; i: 0 geciciDegisken: 11
        printArray(dizi); dizi: [11, 25, 12, 22, 64]
    }
}

```

Yukarıda selection sort algoritmasının debug içerisindeki görüntüsünü görmektesiniz.

#### 4.c Görselleştirmenin Etkisi

##### 4.c.1 Açıklama

İzlediğimiz youtube videosu sıralama algoritmalarını karşılaştırmak açısından faydalı oldu. Algoritmaların kaç tane karşılaştırma işlemi yaptığı ve hangisinin daha etkin olduğu konusunda fikir sahibi olmamıza yardımcı oldu. Bu sayede hangisini kodlayacağımıza ve hangi yöntem ile karşılaştırmak istediğimize karar verdik.

Videolar ve etkileşimli görselleştirme araçlarını karşılaştırdığımızda ise videolar bizce daha faydalı. Çünkü Youtube alışık olduğumuz bir platform ve diğer araçları kullanmayı öğrenmek yerine videoyu izlemek daha hızlı ve basit. Örnek olarak visualgo.net sitesinde kullanım kılavuzu otomatik açılıyor ama onu okumak ve anlamak zaman alıcı.

## 4) ÖZDEĞERLENDİRME TABLOSU

// Özdeğerlendirme tablosuna tahmini notunuzu giriniz. Açıklama kısmını da mutlaka en az 2 anlamlı cümle içerecek şekilde doldurun. **Açıklama kısmında yapıldı, yapılmadı bilgisi ve hangi maddelerin nasıl yapıldığı veya neden yapılmadığı kısaca yazılmalıdır.**

### Özdeğerlendirme Tablosu

Proje 3 Maddeleri	Not	Tahmini Not	Açıklama
1 a) Ağaç (Milli Parklar İkili Arama Ağacı)	10	10	Madde hakkıyla gerçekleştirildi. Binary Search Tree veri yapısını kullandık. Insert metodunda milli

			parkların isimlerini Türkçe karakter desteği ile karşılaştırıp yerleştiriyoruz.
<b>1 b) Derinlik Bulma, Ağacı Listeleme, Düğüm Sayısı Buldurma, Dengeli Ağaç Derinliği Hesaplama</b>	<b>10</b>	<b>10</b>	Madde hakkıyla gerçekleştirildi. Dengeli ağaç derinliği logaritmik formülle bulunurken diğer işlemler recursive olarak yapıldı.
<b>1 c) Arama ve Listeleme</b>	<b>10</b>	<b>10</b>	Madde hakkıyla gerçekleştirildi. Recursive olarak ağaç dolaşılıyor. Node'ların içindeki park isimleri, kişinin girdiği string ile karşılaştırılıyor.
<b>1 d) Kelime Ağacı Oluşturarak Kelimeleri Sayma</b>	<b>10</b>	<b>10</b>	Madde hakkıyla gerçekleştirildi. Milli park ağacı inorder şeklinde dolaşılırken kelimeler işleniyor ve yeni bir ağaç oluşturuluyor.
<b>2) Hash Tablosu</b>	<b>15</b>	<b>15</b>	Madde hakkıyla gerçekleştirildi. Hazır hashtable sınıfı kullanıldı. Main içindeki dosya okuma kısmında milli park objesi oluşturulunca aynı referansı hashtable içine atıyoruz.
<b>3) Yığın Ağacı (Heap)</b>	<b>15</b>	<b>15</b>	Madde hakkıyla gerçekleştirildi. Kitaptan ve Aybars hocamızın slaytlarından yardım alındı.



<b>4) Sıralama Algoritmaları</b>	<b>20</b>	<b>20</b>	Madde hakkıyla gerçekleştirildi. Selection sort kodlandı ve bubble sort ile kıyaslandı.
<b>5) Özdeğerlendirme Tablosu</b>	<b>10</b>	<b>10</b>	Madde hakkıyla gerçekleştirildi. Tablo doldurulurken her konuyu kısaca özetlemeye çalıştık.