

EGE UNIVERSITY FACULTY of ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Data Structure Project 3B

Rahmican İŞİN 05150000697

15/12/2017

Proje 3B' nin Kısa Tanımı

Projenin 1. bölümünde bizden dizi şeklinde ürün bilgilerini alıp ürün nesnesi oluşturup bu nesneler ile ağac oluşturmamız isteniyor. Daha sonra ürünleri kategorilerine göre bir listte tutup her kategori için bir ağaç oluşturmamız isteniyor. Kategori ağaçlarının derinliğini , eleman sayısını ve derinlik ortalamasını bulmamız isteniyor. Bilgisayar kategorisi silinip ürünleri elektronik kategorisine eklememiz isteniyor.

Projenin 2. bölümünde ürün nesnelerini bir Hashtable 'a yerleştirip içinde bilgisayar geçen ürünlere %10 indirim yaparak Hashtable 'ı güncellememiz bekleniyor.

Projenin 3. bölümünde bir Heap yapısı oluşturup maliyetleri içine atmamız, en ucuz 2 ürün ve en pahalı 2 ürünü Heap ten çekmemiz isteniyor.

Projenin 4. bölümünde bir Simple Sorting algoritması ve bir Advanced Sorting algoritması okuyup anlayıp kodlamamız isteniyor.

1. İkili Arama Ağacı (Binary Search Tree)

public static Urun[] UrunOlustur()

String tipinde diziden alınan ürün bilgileri ile ürün nesneleri oluşturulup ürün dizisinde tutulur.

public class Urun

Ürünlerin bilgilerini tutan sınıf.

class Tree {

public void preOrder(TreeNode localRoot)

Agacın preOrder Dolasılması.

public void inOrder(TreeNode localRoot)

Agacın inOrder Dolasılması.

public void postOrder(TreeNode localRoot)

Agacın postOrder Dolasılması.

public void insert(Urun newdata)

Agaca bir düğüm eklemeyi saglar.

public void düzeyListele(TreeNode etkin)

Agacın duzeyleri listelenir.

public void arama(TreeNode localRoot,String aranan)

Agacta ürün adına göre arama yapar.

public boolean isEmpty()

Agacın boş olup olmadığını kontrol eder.

public void agacBilgi(TreeNode rootNode, int elemanSay)

Agacın derinliği eleman sayısını ve ortalama derinliği bulunup yazdırılır.

public void derinlik(TreeNode node,int derinlik){

Derinlik bulmak için kullanılır.

public void karhesapla(TreeNode localRoot)

Ürünlerin satışından elde edilen kar hesaplanır.

public void dolas(TreeNode localRoot)

Agacı dolasmak için kullanılır.

public void agacEkle(TreeNode localRoot)

Agaca bir baska agacın elemanlarını eklemek için kullanılır.

}Tree class sonu

public static void Proje_1a(){

Ürünler dizisindeki string'leri sahalara ayrıştıarak Ürün Nesnelerini oluşturup, Ürün Adı + Marka + Model'e göre ikili arama ağacına yerleştiren metod

public static void Proje_1b(){

Ürünler dizisindeki string'leri sahalara ayrıştıarak Ürün Nesnelerini oluşturup, Ürün Adı+ Marka + Model'e göre kategori ağaçlarına yerleştiren metod

public static List<Tree> kategoriAgacOlustur(){

Kategorilere göre ürünleri bulup kategoriAgac listesine sıralı olarak yerleştirir. Kategori agacı listesini döndürür.

public static ArrayList<String> kategoriSıraDizi(){

Ürünlerin kategorilerini alarak bir diziye aktarır. Dizi alfabetik sıraya göre sıralanıp döndürülür.

public static void Proje_1c(){

Kategori ağaçlarının her birinin derinliğini, eleman sayısını, düğümlerin derinlik ortalamasını bulan ve toplam kar miktarını hesaplar. Ürün Adına göre arama yapılır.

public static void Proje_1d(){

Bilgisayar kategorisini silerek, içindeki tüm ürünleri Elektronik kategorisine ekler.

2. Hash Tablosu (Hash Table)

public static void Proje_2(){

Ürün Nesnelerini (Ürün Adı + Marka + Model) bileşimine göre bir Hash Table'a yerleştirilir. Adında Bilgisayar geçen ürünlerin Satış Fiyatlarında %10 indirim yaparak Hash Tablosunda güncellenir ve yeniden yazdırılır.

3. Yığın Ağacı (Heap)

public static void Proje_3(){

Sadece maliyetler bir Heap e yerleştirilir. Maliyetlerine göre en ucuz 2 ve en pahalı 2 ürünü Heap'ten çekerek listelenir.

public class Heap {

public int sag(int dugum){

Sağ düğümün index i bulunur.

public int sol(int dugum){

Sol düğümün indexi bulunur

public int ustDugum(int dugum){

Üst dugum indexi bulunur.

public void ekle(float eklenen){

Heap e eleman eklenip heap düzenlenir.

public void duzenleYukari(int index){

Aşağıdan yukarıya doğru düzenleme yaparak ebeveynin çocuklardan küçük olması sağlanır.

public void duzenleAsagi(int index){

Heapte yukarıdan aşağıya doğru elemanları düzenler. Ebeveynin çocuklardan küçük olması sağlanır.

public float cikar(){

Heaptten veriler çekilir ve silinir.

}Heap class sonu

```
public static void proje_4a(){
```

Karışık sayılardan dizi oluşturulur. Bubble sort ile liste sıranır.

```
public static void quickSort(int[] a, int altindis, int üstindis)
```

Quick sort algoritmasının bulunduğu metot.

```
public static void Proje_4b()
```

Sayılardan biri dizi oluşturulur. Quick sort algoritması ile dizi küçükten büyüğe sıralanır.

Bubble Sort ve Quick Sort Karşılaştırılması

Bubble sort un zaman karmaşıklığı n^2 dir. Quick sortun zaman karmaşıklığı ise $n \log n$ 'dir. Bubble sort dizideki bütün elemanları 2'li olarak karşılaştırır. Sadece karşılaştırdığı iki elemanın yerini değiştirir. Quick sort ise listeyi 2 ye böler ve listeden bir tane referans noktası seçer. Bu referans noktasından küçükler listenin soluna referans noktasından büyükler ise soluna yerleştirilir. Soldaki ve Sağdakiler kendi aralarında tekrar sıralanır. Küçük listelerde Bubble sort kullanışlı iken listeler büyüdükçe Quick sort daha kullanışlı olur.

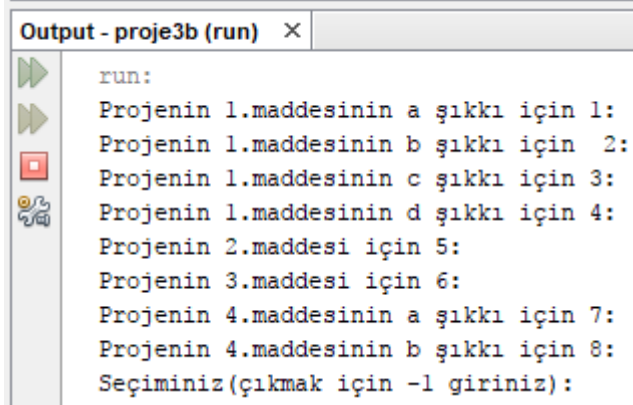
Görselleştirmenin Algoritmaları Anlamanıza Etkisini

Görseller ile algoritmaların nasıl çalıştığını daha iyi anladım. Algoritmaların mantığını bilsem bile nasıl hareket ettiklerini görmek daha anlaşılır hale getirdi ve mantığının oturmasında yardımcı oldu.

Ekran görüntüleri

5

Program başlatıldığında bizden projenin hangi maddesini çalıştırcağımızı seçmemiz istenir.



```
Output - proje3b (run) X
run:
Projenin 1.maddesinin a şıkkı için 1:
Projenin 1.maddesinin b şıkkı için 2:
Projenin 1.maddesinin c şıkkı için 3:
Projenin 1.maddesinin d şıkkı için 4:
Projenin 2.maddesi için 5:
Projenin 3.maddesi için 6:
Projenin 4.maddesinin a şıkkı için 7:
Projenin 4.maddesinin b şıkkı için 8:
Seçiminiz(çıkamak için -1 giriniz):
```

1' e basıldığında

-----Ağaca eklenen ürünler listeleniyor

```
urunAdi=Buzdolabı, kategori=Beyaz Eşya, marka=Regal, model=Cool RGL 3000, miktar=45, maliyet=899.5, satisFiyati=929.0
urunAdi=Dizüstü Bilgisayar, kategori=Bilgisayar, marka=Dell, model=XPS 13 9333 Intel Core i5, miktar=10, maliyet=5799.0, satisFiyati=5927.3
urunAdi=Elektrikli Testere, kategori=Bahçe, marka=Stihl, model=Ms 250 45 cm Pala, miktar=5, maliyet=1500.0, satisFiyati=2750.0
urunAdi=Kalem, kategori=Kırtasiye, marka=Rotring, model=Rapid Versatil Kalem, miktar=63, maliyet=10.0, satisFiyati=28.9
urunAdi=Oyun Bilgisayarı, kategori=Bilgisayar, marka=Monster, model=Tulpar T7 i7 7700, miktar=18, maliyet=5021.0, satisFiyati=6980.0
urunAdi=Televizyon, kategori=Elektronik, marka=Samsung, model=UE43MU7000 Ultra HD, miktar=25, maliyet=2500.0, satisFiyati=3953.0
urunAdi=Tıraş Makinesi, kategori=Kozmetik, marka=Philips, model=5000 Serisi S5050/06 Islak Kuru Şarjlı, miktar=48, maliyet=225.0, satisFiyati=383.0
```

Menu için 1 giriniz:

-----Kategori ağacı listeleniyor

urunAdi=Elektrikli Testere, kategori=Bahçe, marka=Stihl, model=Ms 250 45 cm Pala, miktar=5, maliyet=1500.0, satisFiyati=2750.0

urunAdi=Buzdolabı, kategori=Beyaz Eşya, marka=Regal, model=Cool RGL 3000, miktar=45, maliyet=899.5, satisFiyati=929.0

urunAdi=Dizüstü Bilgisayar, kategori=Bilgisayar, marka=Dell, model=XPS 13 9333 Intel Core i5, miktar=10, maliyet=5799.0, satisFiyati=5927.3

urunAdi=Oyun Bilgisayarı, kategori=Bilgisayar, marka=Monster, model=Tulpar T7 i7 7700, miktar=18, maliyet=5021.0, satisFiyati=6980.0

urunAdi=Televizyon, kategori=Elektronik, marka=Samsung, model=UE43MU7000 Ultra HD, miktar=25, maliyet=2500.0, satisFiyati=3953.0

urunAdi=Tıraş Makinesi, kategori>Kozmetik, marka=Philips, model=5000 Serisi S5050/06 Islak Kuru Şarjlı, miktar=48, maliyet=225.0, satisFiyati=383.0

urunAdi=Kalem, kategori>Kırtasiye, marka=Rotring, model=Rapid Versatil Kalem, miktar=63, maliyet=10.0, satisFiyati=28.9

3'e basıldığında

Bahçe Ağacı

Agacın Derinliği: 0

Eleman Sayısı :1

ortalama Derinlik: 0.0

Beyaz Eşya Ağacı

Agacın Derinliği: 0

Eleman Sayısı :1

ortalama Derinlik: 0.0

Bilgisayar Ağacı

Agacın Derinliği: 1

Eleman Sayısı :2

ortalama Derinlik: 0.5

Elektronik Ağacı

Agacın Derinliği: 0

Eleman Sayısı :1

ortalama Derinlik: 0.0

Kozmetik Ağacı

Agacın Derinliği: 0

Eleman Sayısı :1

ortalama Derinlik: 0.0

Kırtasiye Ağacı

Agacın Derinliği: 0

Eleman Sayısı :1

ortalama Derinlik: 0.0

Toplam Kar :89222.2

4'e basıldığında

7

Aramak istediğiniz ürünün adını giriniz:

Buzdolabı

Bulundu

urunAdi=Buzdolabı, kategori=Beyaz Eşya, marka=Regal, model=Cool RGL 3000, miktar=45, maliyet=899.5, satisFiyati=929.0

5'e basıldığında

-----Hastable dan ürünler yazdırılıyor

urunAdi=Dizüstü Bilgisayar, kategori=Bilgisayar, marka=Dell, model=XPS 13 9333 Intel Core i5, miktar=10, maliyet=5799.0, satisFiyati=5927.3

urunAdi=Elektrikli Testere, kategori=Bahçe, marka=Stihl, model=Ms 250 45 cm Pala, miktar=5, maliyet=1500.0, satisFiyati=2750.0

urunAdi=Buzdolabı, kategori=Beyaz Eşya, marka=Regal, model=Cool RGL 3000, miktar=45, maliyet=899.5, satisFiyati=929.0

urunAdi=Televizyon, kategori=Elektronik, marka=Samsung, model=UE43MU7000 Ultra HD, miktar=25, maliyet=2500.0, satisFiyati=3953.0

urunAdi=Kalem, kategori=Kırtasiye, marka=Rotring, model=Rapid Versatil Kalem, miktar=63, maliyet=10.0, satisFiyati=28.9

urunAdi=Tıraş Makinesi, kategori=Kozmetik, marka=Philips, model=5000 Serisi S5050/06 Islak Kuru Şarjlı, miktar=48, maliyet=225.0, satisFiyati=383.0

urunAdi=Oyun Bilgisayarı, kategori=Bilgisayar, marka=Monster, model=Tulpar T7 i7 7700, miktar=18, maliyet=5021.0, satisFiyati=6980.0

-----Adında bilgisayar geçen ürünlere %10 indirim uygulandıktan sonra

urunAdi=Dizüstü Bilgisayar, kategori=Bilgisayar, marka=Dell, model=XPS 13 9333 Intel Core i5, miktar=10, maliyet=5799.0, satisFiyati=5334.57

urunAdi=Elektrikli Testere, kategori=Bahçe, marka=Stihl, model=Ms 250 45 cm Pala, miktar=5, maliyet=1500.0, satisFiyati=2750.0

urunAdi=Buzdolabı, kategori=Beyaz Eşya, marka=Regal, model=Cool RGL 3000, miktar=45, maliyet=899.5, satisFiyati=929.0

urunAdi=Televizyon, kategori=Elektronik, marka=Samsung, model=UE43MU7000 Ultra HD, miktar=25, maliyet=2500.0, satisFiyati=3953.0

urunAdi=Kalem, kategori=Kırtasiye, marka=Rotring, model=Rapid Versatil Kalem, miktar=63, maliyet=10.0, satisFiyati=28.9

urunAdi=Tıraş Makinesi, kategori=Kozmetik, marka=Philips, model=5000 Serisi S5050/06 Islak Kuru Şarjlı, miktar=48, maliyet=225.0, satisFiyati=383.0

urunAdi=Oyun Bilgisayarı, kategori=Bilgisayar, marka=Monster, model=Tulpar T7 i7 7700, miktar=18, maliyet=5021.0, satisFiyati=6282.0

6'ya basıldığında

En ucuz ürünler

urunAdi=Kalem, kategori=Kırtasiye, marka=Rotring, model=Rapid Versatil Kalem, miktar=63, maliyet=10.0, satisFiyati=28.9

urunAdi=Tıraş Makinesi, kategori=Kozmetik, marka=Philips, model=5000 Serisi S5050/06 Islak Kuru Şarjlı, miktar=48, maliyet=225.0, satisFiyati=383.0

En pahalı ürünler

urunAdi=Oyun Bilgisayarı, kategori=Bilgisayar, marka=Monster, model=Tulpar T7 i7 7700, miktar=18, maliyet=5021.0, satisFiyati=6980.0

urunAdi=Dizüstü Bilgisayar, kategori=Bilgisayar, marka=Dell, model=XPS 13 9333 Intel Core i5, miktar=10, maliyet=5799.0, satisFiyati=5927.3

7'ye basıldığında

Sıralanmadan önceki liste

8 6 32 25 85 12 75 2 48 95 15 28

Buble sort ile sıralandıktan sonra liste

2 6 8 12 15 25 28 32 48 75 85 95

8'e basıldığında

Sıralamadan önceki liste

8 6 32 25 85 12 75 2 48 95 15 28

quick sort ile Sıralamadan sonraki liste

2 6 8 12 15 25 28 32 48 75 85 95


```
package proje3b;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collection;

import java.util.Collections;

import java.util.Comparator;

import java.util.Hashtable;

import java.util.List;

import java.util.Scanner;

public class Proje3b {

    public static void main(String[] args) {

        menu();

    }

    static void menu(){

        Scanner giris=new Scanner(System.in);

        int secim=0;

        while(secim!=-1){

            System.out.print("Projenin 1.maddesinin a şıkkı için 1: "

+ "\nProjenin 1.maddesinin b şıkkı için 2: "

+ "\nProjenin 1.maddesinin c şıkkı için 3: "

+ "\nProjenin 1.maddesinin d şıkkı için 4: "

+ "\nProjenin 2.maddesi için 5: "

+ "\nProjenin 3.maddesi için 6: "

+ "\nProjenin 4.maddesinin a şıkkı için 7: "

+ "\nProjenin 4.maddesinin b şıkkı için 8: "

+ "\nSeçiminiz(çıkamak için -1 giriniz): ");

            secim=giris.nextInt();

        }

    }

}
```

```
switch(secim){  
    case 1:  
        Proje_1a();  
        break;  
    case 2:  
        Proje_1b();  
        break;  
    case 3:  
        Proje_1c();  
        break;  
    case 4:  
        System.out.println("Bilgisayar kategorisini silip ürünleri elektronik kategorisine ekleme");  
        Proje_1d();  
        break;  
    case 5:  
        Proje_2();  
        break;  
    case 6:  
        Proje_3();  
        break;  
    case 7:  
        proje_4a();  
        break;  
    case 8:  
        Proje_4b();  
        break;  
}  
System.out.println("\nMenu için 1 giriniz: ");  
giris.next();  
}}
```

```

public static Urun[] UrunOlustur(){//Dizide bulunan ürün bilgileri split metodu ile ayrılarak nesneler
oluşturulur dizi içinde tutularak return edilir

String[] urunler = { "Dizüstü Bilgisayar, Bilgisayar, Dell, XPS 13 9333 Intel Core i5, 10, 5799,
5927.3", "Elektrikli Testere, Bahçe, Stihl, Ms 250 45 cm Pala, 5, 1500, 2750"
, "Buzdolabı, Beyaz Eşya, Regal, Cool RGL 3000, 45, 899.5, 929"
, "Televizyon, Elektronik, Samsung, UE43MU7000 Ultra HD, 25, 2500, 3953",
"Kalem, Kırtasiye, Rotring, Rapid Versatil Kalem, 63, 10, 28.9", "Traş Makinesi, Kozmetik, Philips, 5000
Serisi S5050/06 Islak Kuru Şarjlı, 48, 225, 383"
, "Oyun Bilgisayarı, Bilgisayar, Monster, Tulpar T7 i7 7700, 18, 5021, 6980"};

Urun[] urunDizi=new Urun[7];

for(int i=0;i<urunler.length;i++){

String[] bilgiler=urunler[i].split(" ");

Urun u=new
Urun(bilgiler[0],bilgiler[1],bilgiler[2],bilgiler[3],Integer.parseInt(bilgiler[4]),Float.parseFloat(bilgiler[5]
),Float.parseFloat(bilgiler[6]));

urunDizi[i]=u;

}

return urunDizi;

} //end urunOlustur

```

```

public static void Proje_1a(){//ürünler dizisindeki string'leri sahalarına ayrıştırarak Ürün
// Nesnelerini oluşturup, Ürün Adı + Marka + Model'e göre ikili arama ağacına yerleştiren metod

Urun[] urunDizi=UrunOlustur();//dizideki bilgiler ile oluşturulmuş nesneler elde edilir.

Tree aramaAgac=new Tree();

System.out.println("-----Ağaca eklenen ürünler listeleniyor\n");

for(int i=0;i<urunDizi.length;i++){

aramaAgac.insert(urunDizi[i]);

}

aramaAgac.inOrder(aramaAgac.getRoot());

} //Proje_1a end

```

```

public static void Proje_1b(){
//ürünler dizisindeki string'leri sahalarına ayrıştırarak Ürün Nesnelerini oluşturup,
// Ürün Adı+ Marka + Model'e göre kategori ağaçlarına yerleştiren metod
List<Tree> kategoriAgac=kategoriAgacOlustur();
System.out.println("-----Kategori ağacı listeleniyor");
//yazdırma
for(Tree agac:kategoriAgac){
agac.inOrder(agac.getRoot());
}
}
}

public static List<Tree> kategoriAgacOlustur(){
//kategorilere göre ürünleri bulup kategoriAgac listesine sıralı olarak
//yerleştirir
Urun[] urunDizi=UrunOlustur();
List<Tree> kategoriAgac=new ArrayList();
ArrayList<String> kategoriAd=kategoriSıraDizi();
for(int i=0;i<kategoriAd.size();i++){
int girdi=0;
for(int k=0;k<urunDizi.length;k++){
if(kategoriAd.get(i).equalsIgnoreCase(urunDizi[k].getKategori())){
if(girdi==0){//aynı kategoride olanları ayırmak için
Tree t=new Tree();
t.insert(urunDizi[k]);
kategoriAgac.add(t);
girdi++;
}
else
kategoriAgac.get(i).insert(urunDizi[k]);
}
}
}
return kategoriAgac;}

```

```

public static ArrayList<String> kategoriSiraDizi(){
    Urun[] urunDizi=UrunOlustur();
    ArrayList<String> kategoriAd=new ArrayList();
    for(int j=0;j<urunDizi.length;j++){//kategoriler bir listeye alınır
        if(!kategoriAd.contains(urunDizi[j].getKategori())){
            kategoriAd.add(urunDizi[j].getKategori());
        }
    }
    Collections.sort(kategoriAd);//kategoriler alfabetik sıralanır
    return kategoriAd;
}

public static void Proje_1c(){
    //Kategori ağaçlarının her birinin derinliğini, eleman sayısını,
    //düğümlerin derinlik ortalamasını bulan ve toplam kar miktarını hesaplar
    List<Tree> kategoriAgac=kategoriAgacOlustur();
    Scanner giris=new Scanner(System.in);
    float toplamKar=0;
    for(Tree agac:kategoriAgac){//kar hesaplanır
        System.out.println("\n"+agac.getRoot().data.getKategori()+" Ağacı");
        agac.agacBilgi(agac.getRoot(), agac.elemanSay);
        agac.karhesapla(agac.getRoot());
        toplamKar+=agac.kar;
    }
    System.out.println("\n"+"Toplam Kar :"+toplamKar);
    // Kullanıcıdan Ürün Adı olarak arama işlemi yapılır
    System.out.println("Aramak istediğiniz ürünün adını giriniz: ");
    String aranan=giris.nextLine();
    for(Tree agac:kategoriAgac){
        agac.arama(agac.getRoot(), aranan);
    }
}

```

```

public static void Proje_1d(){
//Bilgisayar kategorisini silerek, içindeki tüm ürünleri Elektronik kategorisine ekler
List<Tree> kategoriAgac=kategoriAgacOlustur();
ArrayList<String> kategoriSıra=kategoriSıraDizi();
int bilgisayarYer=kategoriSıra.indexOf("Bilgisayar");
int elektronikYer=kategoriSıra.indexOf("Elektronik");
kategoriAgac.get(elektronikYer).agacEkle(kategoriAgac.get(bilgisayarYer).getRoot());
kategoriAgac.remove(bilgisayarYer);
for(Tree agac:kategoriAgac){
System.out.println("\n"+agac.getRoot().data.getKategori()+" Ağacı");
agac.inOrder(agac.getRoot());
}}

public static void Proje_2(){
// Ürün Nesnelerini (Ürün Adı + Marka + Model) bileşimine göre bir Hash Table'a yerleştirilir
Urun[] urunDizi=UrunOlustur();
Hashtable<String, Urun> urunHash = new Hashtable<String, Urun>();
for(int i=0;i<urunDizi.length;i++)
urunHash.put(urunDizi[i].getAdMarkaModel(), urunDizi[i]);
System.out.println("\n-----Hastable dan ürünler yazdırılıyor");
for(int i=0;i<urunDizi.length;i++){
System.out.println(urunHash.get(urunDizi[i].getAdMarkaModel()));}

//Adında Bilgisayar geçen ürünlerin Satış Fiyatlarında %10 indirim yaparak Hash Tablosunda günceller
for(Urun dizi:urunDizi){
if(dizi.getUrunAdi().indexOf("Bilgisayar")!=-1){
dizi.setSatisFiyati((dizi.getSatisFiyati()*9)/10);
urunHash.replace(dizi.getAdMarkaModel(), dizi);}}

System.out.println("\n-----Adında bilgisayar geçen ürünlere %10 indirim uygulandıktan sonra\n");
for(int i=0;i<urunDizi.length;i++){
System.out.println(urunHash.get(urunDizi[i].getAdMarkaModel()));}

```

```
public static void Proje_3(){
// Sadece maliyetler bir Heap e yerleştirilir
// Maliyetlerine göre en ucuz 2 ve en pahalı 2 ürünü Heap'ten çekerek listelenir
Urun[] urunler=UrunOlustur();
Heap maliyet=new Heap();//heap oluşturulur
for(Urun u:urunler){ //maliyetler heap e eklenir
maliyet.ekle(u.getMaliyet());
}
// for(int i=0;i<urunler.length;i++)
// System.out.println(maliyet.cıkar());
float[] maliyetEnler=new float[4]; //en küçük ve en büyük 2 maliyetleri tutmak için
for(int i=0;i<urunler.length;i++){//heaptan en büyük ve en küçük 2 eleman çekilir
if(i<2)//en küçük iki eleman için
maliyetEnler[i]=maliyet.cıkar();
else if(i>urunler.length-3)//en büyük 2 eleman için
maliyetEnler[i-3]=maliyet.cıkar();
else{
maliyet.cıkar();
}
}
for(int i=0;i<maliyetEnler.length;i++){ //ürünler yazdırılır
if(i==0)
System.out.println("En ucuz ürünler\n");
else{ if(i==2)
System.out.println("En pahalı ürünler\n");
}
for(Urun u:urunler){
if(u.getMaliyet()==maliyetEnler[i])
System.out.println(u);
}}
}
```

```
public static void proje_4a(){//buble sort ile liste sıranır
int[] dizi={8,6,32,25,85,12,75,2,48,95,15,28};
System.out.println("Sıralanmadan önceki liste");
for(int i:dizi)
System.out.print(i+" ");
int degisim=0;
while(degisim==0){//değişim olmadığında döngü sonlanır ve dizi sıralanmış olur
int girdi=0;
for(int i=0;i<dizi.length-1;i++){//2 li çiftler halinde elemanları karşılaştırır
if(dizi[i]>dizi[i+1]){//küçük olan başa alınır
int kopya=dizi[i];
dizi[i]=dizi[i+1];
dizi[i+1]=kopya;
girdi=1;
}
}
if(girdi!=1)
degisim=1;
}
System.out.println("\nBuble sort ile sıralandıktan sonra liste");
for(int i:dizi){
System.out.print(i+" ");
}
System.out.println("\n");
}
```



```

public static void quickSort(int[] a, int altindis, int üstindis) {
    // altindis o adımda sıralanan altdizinin ek küçük indisidir
    // üstindis o adımda sıralanan altdizinin ek büyük indisidir
    int i = altindis, j = üstindis, h; // x terimi, mukayesenin yapılacağı mihenk'dir (pivot)
    int x = a[(altindis + üstindis) / 2];
    // Takas eylemiyle diziyi ayırıştırma
    do {
        while (a[i] < x)
            i++;
        while (a[j] > x)
            j--;
        if (i <= j) {
            h = a[i];
            a[i] = a[j];
            a[j] = h;
            i++; j--;
        }
    } while (i <= j);
    // yinelge (recursion)
    if (altindis < j)
        quickSort(a, altindis, j);
    if (i < üstindis)
        quickSort(a, i, üstindis);}

public static void Proje_4b(){//dizi quickSort metodu kullanılarak sıralanır
    int[] dizi={8,6,32,25,85,12,75,2,48,95,15,28};
    System.out.println("Sıralamadan önceki liste");
    for(int i:dizi){
        System.out.print(i+" ");
    }
    quickSort(dizi,0,dizi.length-1);
    System.out.println("\nquick sort ile Sıralamadan sonraki liste");
    for(int i:dizi){
        System.out.print(i+" ");
    }
}

```

Öz değerlendirme Tablosu

Proje 3 Maddeleri	Not	Tahmini Not	Açıklama
1a)Ağaç	10	10	Yapıldı ve istenilen şekilde çalışıyor.
1 b)Kategori Ağacı	10	10	Yapıldı ve istenilen şekilde çalışıyor.
1 c) Değerlerin hesaplanması , listelemeler	10	10	Yapıldı ve istenilen şekilde çalışıyor.
1 d) Kategori birleştirme	10	10	Yapıldı ve istenilen şekilde çalışıyor.
2) Hash Tablosu	15	15	Yapıldı ve istenilen şekilde çalışıyor.
3) Yığın Ağacı(heap)	15	15	Yapıldı ve istenilen şekilde çalışıyor.
4)Sıralama Algoritmaları	20	20	Yapıldı ve istenilen şekilde çalışıyor.
5)Özdeğerlendirme Tablosu	10	10	Yapıldı.