



EGE UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

204 DATA STRUCTURES (3+1)

2021–2022 FALL SEMESTER

PROJECT-3 REPORT

(Search Tree, Heap, Hash Table, Food Delivery Company)

DELIVERY DATE

15/01/2022

PREPARED BY

05200000030, Osman Bora Yılmazmıř

İçindekiler

1.a YemekSınıfı nesnesinin ve Yemek Sipariş Ağacının oluşturulması	3
1.b Ağaç derinliği ve ağaçtaki bilgilerin ekrana listelenmesi	4
1.b.1 Kaynak Kod	4
1.b.2 Ekran görüntüleri.....	5
1.c 150 TL üstündeki siparişlerin bilgilerini listeleme	8
1.c.1 Kaynak Kod.....	8
1.c.2 Ekran görüntüleri	8
1.d Bir yiyecek/içeceğin tüm ağaçta kaç adet sipariş verildiğini döndürme ve fiyat güncelleme	10
1.d.1 Kaynak Kod	10
1.d.2 Ekran görüntüleri.....	11
2.a Hash Tablosuna Ekleme	12
2.b Hash Tablosu Güncelleme	13
3.a Heap Veri Yapısı Tasarlama	14
3.b Max Heap düğüm yerleştirme	16
3.c Heap bilgi çekme.....	17
3.c.1 Kaynak Kod.....	17
3.c.2 Ekran görüntüleri	18
4.a Simple sorting veya Advanced sorting algoritması.....	18
4.b Sıralama algoritması ile karşılaştırılması	20
4.c Görselleştirme araçları.....	22
Özdeğerlendirme Tablosu	23

ARAMA AĞAÇLARI, YİĞİNLAR VE HASH TABLOSU: YEMEK DAĞITIM ŞİRKETİ

//The platform, version, and programming language used

1.a YemekSınıfı nesnesinin ve Yemek Sipariş Ağacının oluşturulması

//1-a maddesi için yazmış olduğunuz kod parçasını ve açıklamaları buraya ekleyiniz.

////////////////////////////////////

```
class YemekSınıfı //YEMEK SINIFIM
{
    public string yemek;
    public int yemeksayisi;
    public double fiyat;

    public YemekSınıfı(string yemek, int yemeksayisi, double fiyat)
    {
        this.yemek = yemek;
        this.yemeksayisi = yemeksayisi;
        this.fiyat = fiyat;
    }
}

class TreeNode //TREENODE SINIFIM
{
    public MahalleSınıfı Mahalle;
    public TreeNode leftChild;
    public TreeNode rightChild;
    public void displayNode()
    { Console.WriteLine(" " + Mahalle + " "); }
}

class Tree//TREE SINIFIM
{
    public TreeNode root;
    public void insert(MahalleSınıfı newdata) //AĞACA ELEMAN EKLEME METODUM
    {
        TreeNode newNode = new TreeNode();
        newNode.Mahalle = newdata;
        if (root == null)
            root = newNode;
        else
        {
            TreeNode current = root;
            TreeNode parent;
            while (true)
            {
                parent = current;
                if (newdata.MahalleAdi.CompareTo(current.Mahalle.MahalleAdi) < 0)
                {
                    current = current.leftChild;
                    if (current == null)
                    {

```

////////////////////////////////////

////////////////////////////////////

```

public int Derinlikbul(TreeNode Node)
{
    Console.WriteLine("Mahalle : " + Node.Mahalle.MahalleAdi);
    int sagderinlik = derinlik;
    int solderinlik = derinlik;
    if( Node.rightChild != null)
    {
        sagderinlik = Derinlikbul(derinlik+1, Node.rightChild);
    }

    if (Node.leftChild != null)
    {
        solderinlik = Derinlikbul(derinlik + 1, Node.leftChild);
    }
    if(sagderinlik < solderinlik) { return solderinlik; }
    return sagderinlik;
}

public void MahalleBuldur(TreeNode Node, string Mahalleadi)
{
    if (Node == null) { return; }
    if (Node.Mahalle.MahalleAdi == Mahalleadi)

```

```

        {
            Node.Mahalle.YüzelliBulYazdir();
        }
    else
    {
        MahalleBuldur(Node.leftChild, Mahalleadi);
        MahalleBuldur(Node.rightChild, Mahalleadi);
    }
}
////////////////////////////////////
Console.WriteLine(" ");
int derinlik = MahalleNesnesi.Derinlikbul(0, MahalleNesnesi.root);
Console.WriteLine(" ");
Console.WriteLine(" Derinlik : " + derinlik);
Console.WriteLine(" ");
Console.WriteLine("Kazımdirik Mahallesi Siparişler:");
MahalleNesnesi.MahalleBuldur(MahalleNesnesi.root, "Kazımdirik");
Console.WriteLine(" ");
Console.WriteLine("Erzene Mahallesi Siparişler:");
MahalleNesnesi.MahalleBuldur(MahalleNesnesi.root, "Erzene");
Console.WriteLine(" ");
Console.WriteLine("Atatürk Mahallesi Siparişler:");
MahalleNesnesi.MahalleBuldur(MahalleNesnesi.root, "Atatürk");
Console.WriteLine(" ");
Console.WriteLine("Özkanlar Mahallesi Siparişler:");
MahalleNesnesi.MahalleBuldur(MahalleNesnesi.root, "Özkanlar");
Console.WriteLine(" ");
Console.WriteLine("Evka 3 Mahallesi Siparişler:");
MahalleNesnesi.MahalleBuldur(MahalleNesnesi.root, "Evka 3");

```

1.b.2 Ekran görüntüleri

//1-b maddesi için üretilen konsol/ekran görüntüsünü buraya ekleyiniz

Mahalle : Evka 3
Mahalle : Özkanlar
Mahalle : Kazımdirik
Mahalle : Atatürk
Mahalle : Erzene

Derinlik : 2

Evka 3 Mahallesi Siparişler:

Yemek Bilgileri: hotdog 5 TL
Yemek Bilgileri: izmir kumrusu 20 TL
Yemek Bilgileri: salçalı tavuk 63 TL
Yemek Bilgileri: küfte 20 TL
Yemek Bilgileri: bulgur pilavı 70 TL
Yemek Bilgileri: pilav 97 TL
Yemek Bilgileri: Sucuklu Şahanem 70 TL
Yemek Bilgileri: nohutlu tavuk 97 TL
Yemek Bilgileri: spagetti 100 TL
Yemek Bilgileri: pilav 20 TL
Yemek Bilgileri: salçalı tavuk 17 TL
Yemek Bilgileri: bulgur pilavı 97 TL
Yemek Bilgileri: mantı 63 TL
Yemek Bilgileri: pilav 20 TL
Yemek Bilgileri: Sucuklu Şahanem 2,5 TL
Yemek Bilgileri: pilav 2,5 TL
Yemek Bilgileri: izmir kumrusu 20 TL
Yemek Bilgileri: Sucuklu Şahanem 70 TL
Yemek Bilgileri: hotdog 50 TL
Yemek Bilgileri: mantı 70 TL
Yemek Bilgileri: mantı 250 TL
Yemek Bilgileri: mantı 25 TL
Yemek Bilgileri: salçalı tavuk 250 TL
Yemek Bilgileri: dana rosto 17 TL
Yemek Bilgileri: izmir kumrusu 250 TL
Yemek Bilgileri: salçalı tavuk 90 TL

Özkanlar Mahallesi Siparişler:

Yemek Bilgileri: mantı 17 TL
Yemek Bilgileri: mantı 20 TL
Yemek Bilgileri: salçalı tavuk 90 TL
Yemek Bilgileri: salçalı tavuk 5 TL
Yemek Bilgileri: Sucuklu Şahanem 25 TL
Yemek Bilgileri: hatay dürüm 2,5 TL
Yemek Bilgileri: spagetti 97 TL
Yemek Bilgileri: hotdog 5 TL
Yemek Bilgileri: salçalı tavuk 97 TL
Yemek Bilgileri: yoğurtlu makarna 25 TL
Yemek Bilgileri: bulgur pilavı 25 TL
Yemek Bilgileri: nohutlu tavuk 5 TL
Yemek Bilgileri: bulgur pilavı 17 TL
Yemek Bilgileri: mantı 50 TL
Yemek Bilgileri: izmir kumrusu 10 TL
Yemek Bilgileri: küfte 63 TL
Yemek Bilgileri: pilav 70 TL
Yemek Bilgileri: mantı 100 TL
Yemek Bilgileri: spagetti 90 TL
Yemek Bilgileri: nohutlu tavuk 70 TL
Yemek Bilgileri: salçalı tavuk 5 TL
Yemek Bilgileri: izmir kumrusu 70 TL
Yemek Bilgileri: dana rosto 20 TL
Yemek Bilgileri: mantı 17 TL
Yemek Bilgileri: bulgur pilavı 90 TL
Yemek Bilgileri: pilav 63 TL
Yemek Bilgileri: pilav 2,5 TL
Yemek Bilgileri: küfte 2,5 TL
Yemek Bilgileri: izmir kumrusu 10 TL
Yemek Bilgileri: nohutlu tavuk 2,5 TL
Yemek Bilgileri: salçalı tavuk 10 TL
Yemek Bilgileri: hotdog 97 TL
Yemek Bilgileri: küfte 5 TL
Yemek Bilgileri: bulgur pilavı 3,5 TL
Yemek Bilgileri: küfte 2,5 TL
Yemek Bilgileri: dana rosto 97 TL
Yemek Bilgileri: yoğurtlu makarna 3,5 TL
Yemek Bilgileri: dana rosto 97 TL
Yemek Bilgileri: hotdog 5 TL

Atatürk Mahallesi Siparişler:

Yemek Bilgileri:	Sucuklu Şahanem	70 TL
Yemek Bilgileri:	dana rosto	25 TL
Yemek Bilgileri:	bulgur pilavı	97 TL
Yemek Bilgileri:	salçalı tavuk	5 TL
Yemek Bilgileri:	küfte	5 TL
Yemek Bilgileri:	dana rosto	20 TL
Yemek Bilgileri:	Sucuklu Şahanem	10 TL
Yemek Bilgileri:	salçalı tavuk	17 TL
Yemek Bilgileri:	mantı	20 TL
Yemek Bilgileri:	bulgur pilavı	5 TL
Yemek Bilgileri:	Sucuklu Şahanem	50 TL
Yemek Bilgileri:	dana rosto	63 TL
Yemek Bilgileri:	küfte	70 TL
Yemek Bilgileri:	salçalı tavuk	90 TL
Yemek Bilgileri:	izmir kumrusu	97 TL
Yemek Bilgileri:	hotdog	17 TL
Yemek Bilgileri:	küfte	5 TL
Yemek Bilgileri:	mantı	100 TL
Yemek Bilgileri:	izmir kumrusu	97 TL
Yemek Bilgileri:	spagetti	20 TL
Yemek Bilgileri:	spagetti	100 TL
Yemek Bilgileri:	spagetti	5 TL
Yemek Bilgileri:	yoğurtlu makarna	50 TL
Yemek Bilgileri:	izmir kumrusu	3,5 TL
Yemek Bilgileri:	salçalı tavuk	10 TL
Yemek Bilgileri:	dana rosto	50 TL
Yemek Bilgileri:	Sucuklu Şahanem	10 TL
Yemek Bilgileri:	hatay dürüm	50 TL
Yemek Bilgileri:	Sucuklu Şahanem	90 TL
Yemek Bilgileri:	Sucuklu Şahanem	70 TL
Yemek Bilgileri:	yoğurtlu makarna	100 TL
Yemek Bilgileri:	hatay dürüm	100 TL
Yemek Bilgileri:	küfte	97 TL
Yemek Bilgileri:	dana rosto	97 TL
Yemek Bilgileri:	bulgur pilavı	100 TL
Yemek Bilgileri:	küfte	25 TL
Yemek Bilgileri:	spagetti	97 TL
Yemek Bilgileri:	pilav	5 TL
Yemek Bilgileri:	nohutlu tavuk	5 TL

Kazım Dirik Mahallesi Siparişler:

Yemek Bilgileri:	bulgur pilavı	10 TL
Yemek Bilgileri:	dana rosto	3,5 TL
Yemek Bilgileri:	hotdog	20 TL
Yemek Bilgileri:	salçalı tavuk	63 TL
Yemek Bilgileri:	küfte	250 TL
Yemek Bilgileri:	pilav	2,5 TL
Yemek Bilgileri:	hatay dürüm	5 TL
Yemek Bilgileri:	pilav	10 TL
Yemek Bilgileri:	pilav	10 TL
Yemek Bilgileri:	izmir kumrusu	10 TL
Yemek Bilgileri:	salçalı tavuk	10 TL
Yemek Bilgileri:	bulgur pilavı	250 TL
Yemek Bilgileri:	pilav	2,5 TL
Yemek Bilgileri:	Sucuklu Şahanem	17 TL
Yemek Bilgileri:	Sucuklu Şahanem	63 TL
Yemek Bilgileri:	hatay dürüm	17 TL
Yemek Bilgileri:	izmir kumrusu	97 TL
Yemek Bilgileri:	mantı	17 TL
Yemek Bilgileri:	spagetti	5 TL
Yemek Bilgileri:	Sucuklu Şahanem	97 TL
Yemek Bilgileri:	küfte	20 TL
Yemek Bilgileri:	nohutlu tavuk	5 TL
Yemek Bilgileri:	mantı	5 TL
Yemek Bilgileri:	nohutlu tavuk	10 TL
Yemek Bilgileri:	nohutlu tavuk	17 TL
Yemek Bilgileri:	Sucuklu Şahanem	100 TL
Yemek Bilgileri:	spagetti	63 TL
Yemek Bilgileri:	salçalı tavuk	100 TL
Yemek Bilgileri:	salçalı tavuk	3,5 TL
Yemek Bilgileri:	salçalı tavuk	250 TL
Yemek Bilgileri:	hotdog	3,5 TL
Yemek Bilgileri:	spagetti	250 TL
Yemek Bilgileri:	pilav	100 TL
Yemek Bilgileri:	pilav	50 TL
Yemek Bilgileri:	nohutlu tavuk	90 TL
Yemek Bilgileri:	hatay dürüm	2,5 TL
Yemek Bilgileri:	nohutlu tavuk	5 TL
Yemek Bilgileri:	spagetti	90 TL
Yemek Bilgileri:	küfte	25 TL

Erzene Mahallesi Siparişler:

Yemek Bilgileri:	spagetti	97 TL
Yemek Bilgileri:	bulgur pilavı	17 TL
Yemek Bilgileri:	bulgur pilavı	17 TL
Yemek Bilgileri:	hotdog	2,5 TL
Yemek Bilgileri:	bulgur pilavı	70 TL
Yemek Bilgileri:	mantı	17 TL
Yemek Bilgileri:	spagetti	10 TL
Yemek Bilgileri:	yoğurtlu makarna	70 TL
Yemek Bilgileri:	küfte	50 TL
Yemek Bilgileri:	hotdog	2,5 TL
Yemek Bilgileri:	salçalı tavuk	63 TL
Yemek Bilgileri:	izmir kumrusu	25 TL
Yemek Bilgileri:	izmir kumrusu	17 TL
Yemek Bilgileri:	hotdog	250 TL
Yemek Bilgileri:	yoğurtlu makarna	20 TL
Yemek Bilgileri:	pilav	50 TL
Yemek Bilgileri:	mantı	50 TL
Yemek Bilgileri:	izmir kumrusu	63 TL
Yemek Bilgileri:	dana rosto	5 TL
Yemek Bilgileri:	hotdog	50 TL
Yemek Bilgileri:	pilav	97 TL
Yemek Bilgileri:	izmir kumrusu	70 TL
Yemek Bilgileri:	Sucuklu Şahanem	50 TL
Yemek Bilgileri:	pilav	50 TL
Yemek Bilgileri:	Sucuklu Şahanem	63 TL
Yemek Bilgileri:	izmir kumrusu	17 TL

1.c.1 Kaynak Kod

////////////////////////////////////

```
//YUKARIDA SİPARİŞLER SINIFININ İÇİNE TOPLAM ÜCRET METODUNU TANIMLADIM BURADA SİPARİŞTEKİ TOPLAM ÜCRET DOUBLE CİNSİNDEN BULUNUYOR DAHA SONRA AŞŞAĞIDAKİ MAHALLESİNİN İÇİNDEKİ YÜZELLİBULYAZDIR() METODU KULLANILARAK SİPARİŞ BİLGİLERİNDEKİ 150 TL Yİ GECEN SİPARİŞLER YAZDIRILMAKTA
```

```
class MahalleSınıfı //MAHALLE SINIFIM
{
    public string MahalleAdi;
    public List<SiparişlerSınıfı> SiparişBilgileri = new List<SiparişlerSınıfı>();

    public MahalleSınıfı(string mahalleadi)
    {
        this.MahalleAdi = mahalleadi;
    }
    public void YüzelliBulYazdır()//MAİN METODUNUN İÇİNE ÇAĞRILAN METOD
    {
        foreach(SiparişlerSınıfı siparişler in SiparişBilgileri)
        {
            if(siparişler.ToplamÜcret()> 150)
            {
                foreach(YemekSınıfı yemek in siparişler.SiparişBilgileri)
                {
                    Console.WriteLine("Yemek Bilgileri: " + yemek.yemek + "      "+"
yemek.fiyat+ " TL");
                }
            }
        }
    }
}
```



```

    }
    }
    }
}

////////////////////////////////////

        if(i == 2)
        {
            Console.WriteLine("Atatürk Mahallesi İçin 150 TL'nin üzerindeki siparişlerin bilgileri ");
            Mahalle.YuzelliBulYazdir();
        }

```

1.c.2 Ekran görüntüleri

//1-c maddesi için üretilen konsol/ekran görüntüsünü buraya ekleyiniz

```

Atatürk Mahallesi İçin 150 TL'nin üzerindeki siparişlerin bilgileri
Yemek Bilgileri: Sucuklu Şahanem    70 TL
Yemek Bilgileri: dana rosto        25 TL
Yemek Bilgileri: bulgur pilavı     97 TL
Yemek Bilgileri: salçalı tavuk     5 TL
Yemek Bilgileri: küfte             5 TL
Yemek Bilgileri: dana rosto        20 TL
Yemek Bilgileri: Sucuklu Şahanem    10 TL
Yemek Bilgileri: salçalı tavuk     17 TL
Yemek Bilgileri: mantı             20 TL
Yemek Bilgileri: bulgur pilavı     5 TL
Yemek Bilgileri: Sucuklu Şahanem    50 TL
Yemek Bilgileri: dana rosto        63 TL
Yemek Bilgileri: küfte             70 TL
Yemek Bilgileri: salçalı tavuk     90 TL
Yemek Bilgileri: izmir kumrusu     97 TL
Yemek Bilgileri: hotdog            17 TL
Yemek Bilgileri: küfte             5 TL
Yemek Bilgileri: mantı             100 TL
Yemek Bilgileri: izmir kumrusu     97 TL
Yemek Bilgileri: spagetti          20 TL
Yemek Bilgileri: spagetti          100 TL
Yemek Bilgileri: spagetti          5 TL
Yemek Bilgileri: yoğurtlu makarna   50 TL
Yemek Bilgileri: izmir kumrusu     3,5 TL
Yemek Bilgileri: salçalı tavuk     10 TL
Yemek Bilgileri: dana rosto        50 TL
Yemek Bilgileri: Sucuklu Şahanem    10 TL
Yemek Bilgileri: hatay dürüm       50 TL
Yemek Bilgileri: Sucuklu Şahanem    90 TL
Yemek Bilgileri: Sucuklu Şahanem    70 TL
Yemek Bilgileri: yoğurtlu makarna   100 TL
Yemek Bilgileri: hatay dürüm       100 TL
Yemek Bilgileri: küfte             97 TL
Yemek Bilgileri: dana rosto        97 TL
Yemek Bilgileri: bulgur pilavı     100 TL
Yemek Bilgileri: küfte             25 TL
Yemek Bilgileri: spagetti          97 TL
Yemek Bilgileri: pilav             5 TL
Yemek Bilgileri: nohutlu tavuk     5 TL

```

1.d.1 Kaynak Kod

////////////////////////////////////

```

public int YemekAdedi(TreeNode Node ,string yemekadi )
{
    int toplamYemekAdedi = 0;
    foreach(SiparişlerSınıfı siparis in Node.Mahalle.SiparişBilgileri)
    {
        foreach(YemekSınıfı yemek in siparis.SiparişBilgileri)
        {
            if(yemek.yemek == yemekadi)
            {
                toplamYemekAdedi += yemek.yemeksayisi;
                yemek.fiyat = yemek.fiyat * 0.9;
            }
        }
    }

    if (Node.rightChild != null)
    {
        toplamYemekAdedi += YemekAdedi(Node.rightChild, yemekadi);
    }

    if (Node.leftChild != null)
    {
        toplamYemekAdedi += YemekAdedi(Node.leftChild, yemekadi);
    }

    return toplamYemekAdedi;
}

```

////////////////////////////////////

```

        Console.WriteLine("Kazımdirik Mahallesi Siparişler:");
        MahalleNesnesi.MahalleBuldur(MahalleNesnesi.root, "Kazımdirik");

        Console.WriteLine(" ");
        Console.WriteLine("Güncelleştirildikten sonra ");
        Console.WriteLine("Küfte Adedi :
"+MahalleNesnesi.YemekAdedi(MahalleNesnesi.root, "küfte"));
        MahalleNesnesi.MahalleBuldur(MahalleNesnesi.root, "Kazımdirik");

```

1.d.2 Ekran görüntüleri

//1-d maddesi için üretilen konsol/ekran görüntüsünü buraya ekleyiniz

Güncelleştirilmeden Önce

//Güncelleştirilecek yemek küfte

```
Kazımdirik Mahallesi Siparişler:
Yemek Bilgileri: bulgur pilavı      10 TL
Yemek Bilgileri: dana rosto        3,5 TL
Yemek Bilgileri: hotdog             20 TL
Yemek Bilgileri: salçalı tavuk      63 TL
Yemek Bilgileri: küfte              250 TL
Yemek Bilgileri: pilav              2,5 TL
Yemek Bilgileri: hatay dürüm        5 TL
Yemek Bilgileri: pilav              10 TL
Yemek Bilgileri: pilav              10 TL
Yemek Bilgileri: izmir kumrusu      10 TL
Yemek Bilgileri: salçalı tavuk      10 TL
Yemek Bilgileri: bulgur pilavı      250 TL
Yemek Bilgileri: pilav              2,5 TL
Yemek Bilgileri: Sucuklu Şahanem    17 TL
Yemek Bilgileri: Sucuklu Şahanem    63 TL
Yemek Bilgileri: hatay dürüm        17 TL
Yemek Bilgileri: izmir kumrusu      97 TL
Yemek Bilgileri: mantı              17 TL
Yemek Bilgileri: spagetti           5 TL
Yemek Bilgileri: Sucuklu Şahanem    97 TL
Yemek Bilgileri: küfte              20 TL
Yemek Bilgileri: nohutlu tavuk      5 TL
Yemek Bilgileri: mantı              5 TL
Yemek Bilgileri: nohutlu tavuk      10 TL
Yemek Bilgileri: nohutlu tavuk      17 TL
Yemek Bilgileri: Sucuklu Şahanem    100 TL
Yemek Bilgileri: spagetti           63 TL
Yemek Bilgileri: salçalı tavuk      100 TL
Yemek Bilgileri: salçalı tavuk      3,5 TL
Yemek Bilgileri: salçalı tavuk      250 TL
Yemek Bilgileri: hotdog             3,5 TL
Yemek Bilgileri: spagetti           250 TL
Yemek Bilgileri: pilav              100 TL
Yemek Bilgileri: pilav              50 TL
Yemek Bilgileri: nohutlu tavuk      90 TL
Yemek Bilgileri: hatay dürüm        2,5 TL
Yemek Bilgileri: nohutlu tavuk      5 TL
Yemek Bilgileri: spagetti           90 TL
Yemek Bilgileri: küfte              25 TL
```

Güncelleştirildikten sonra

Güncelleştirildikten sonra
Küfte Adedi : 55
Yemek Bilgileri: bulgur pilavı 10 TL
Yemek Bilgileri: dana rosto 3,5 TL
Yemek Bilgileri: hotdog 20 TL
Yemek Bilgileri: salçalı tavuk 63 TL
Yemek Bilgileri: küfte 225 TL
Yemek Bilgileri: pilav 2,5 TL
Yemek Bilgileri: hatay dürüm 5 TL
Yemek Bilgileri: pilav 10 TL
Yemek Bilgileri: pilav 10 TL
Yemek Bilgileri: izmir kumrusu 10 TL
Yemek Bilgileri: salçalı tavuk 10 TL
Yemek Bilgileri: bulgur pilavı 250 TL
Yemek Bilgileri: pilav 2,5 TL
Yemek Bilgileri: Sucuklu Şahanem 17 TL
Yemek Bilgileri: Sucuklu Şahanem 63 TL
Yemek Bilgileri: hatay dürüm 17 TL
Yemek Bilgileri: izmir kumrusu 97 TL
Yemek Bilgileri: mantı 17 TL
Yemek Bilgileri: spagetti 5 TL
Yemek Bilgileri: Sucuklu Şahanem 97 TL
Yemek Bilgileri: küfte 18 TL
Yemek Bilgileri: nohutlu tavuk 5 TL
Yemek Bilgileri: mantı 5 TL
Yemek Bilgileri: nohutlu tavuk 10 TL
Yemek Bilgileri: nohutlu tavuk 17 TL
Yemek Bilgileri: Sucuklu Şahanem 100 TL
Yemek Bilgileri: spagetti 63 TL
Yemek Bilgileri: salçalı tavuk 100 TL
Yemek Bilgileri: salçalı tavuk 3,5 TL
Yemek Bilgileri: salçalı tavuk 250 TL
Yemek Bilgileri: hotdog 3,5 TL
Yemek Bilgileri: spagetti 250 TL
Yemek Bilgileri: pilav 100 TL
Yemek Bilgileri: pilav 50 TL
Yemek Bilgileri: nohutlu tavuk 90 TL
Yemek Bilgileri: hatay dürüm 2,5 TL
Yemek Bilgileri: nohutlu tavuk 5 TL
Yemek Bilgileri: spagetti 90 TL
Yemek Bilgileri: küfte 22,5 TL

2.a Hash Tablosuna Ekleme

//2-a maddesi için yazmış olduğunuz kodları ve açıklamaları buraya ekleyiniz

```

        string[] mahalleler = { "KURUDERE MAHALLESİ", "SARNIÇKÖY MAHALLESİ", "KAYADİBİ
MAHALLESİ", "ÇAMİÇİ MAHALLESİ", "BEŞYOL MAHALLESİ", "GÖKDERE MAHALLESİ", "ÇİÇEKLİ
MAHALLESİ", "LAKA MAHALLESİ", "KARAÇAM MAHALLESİ", "YAKAKÖY MAHALLESİ" };//MAHALLELERİN
LİSTESİ
        int[] nüfus = { 59, 81, 250, 265, 334, 392, 399, 432, 590, 1217 }; //NÜFUS
LİSTESİ
        Hashtable table = new Hashtable(); //HASHTABLE OLUŞTURUR
        Console.WriteLine("Oluşturulan Hashtable:");
        for (int i = 0; mahalleler.Length > i; i++)
        {
            table[mahalleler[i]] = nüfus[i];//HASH TABLE'a EKLEME GERÇEKLEŞİR
        }
        for (int i = 0; mahalleler.Length > i; i++)
        {
            Console.WriteLine(mahalleler[i] + " " + table[mahalleler[i]]);
//EKLENENLER EKRANA YAZDIRILIR
        }

```

```

Oluşturulan Hashtable:
KURUDERE MAHALLESİ 59
SARNIÇKÖY MAHALLESİ 81
KAYADİBİ MAHALLESİ 250
ÇAMİÇİ MAHALLESİ 265
BEŞYOL MAHALLESİ 334
GÖKDERE MAHALLESİ 392
ÇİÇEKLİ MAHALLESİ 399
LAKA MAHALLESİ 432
KARAÇAM MAHALLESİ 590
YAKAKÖY MAHALLESİ 1217

```

>>> HASHTABLE'la OLUŞTURULAN MAHALLELER

2.b Hash Tablosu Güncelleme

//2-b maddesi için yazmış olduğunuz kodları ve açıklamaları buraya ekleyiniz

```

        Console.WriteLine("Baş harfi girildikten sonra:");

        while (true)
        {
            char girilenharf = Console.ReadKey().KeyChar; //KULLANICIDAN HARF ALINIR

            Console.WriteLine(" ");
            for (int i = 0; mahalleler.Length > i; i++)
            {
                if (mahalleler[i].ToLower().StartsWith(girilenharf.ToString()))
                {
                    table[mahalleler[i]] = ++nüfus[i]; //BAŞ HARFİ GİRİLEN NÜFUS 1
ARTTIRILIR
                }
            }
            for (int i = 0; mahalleler.Length > i; i++)
            {
                Console.WriteLine(mahalleler[i] + " " + table[mahalleler[i]]);//NÜFUS
ARTTIRILIDİKTAN SONRA OLUŞAN LİSTE EKRANA YAZDIRILIR
            }
        }

```

```
Console.WriteLine(" ");
```

```
}
```

Baş harfi girildikten sonra:

```
k
KURUDERE MAHALLESİ 60
SARNIÇKÖY MAHALLESİ 81
KAYADİBİ MAHALLESİ 251
ÇAMIÇİ MAHALLESİ 265
BEŞYOL MAHALLESİ 334
GÖKDERE MAHALLESİ 392
ÇİÇEKLİ MAHALLESİ 399
LAKA MAHALLESİ 432
KARAÇAM MAHALLESİ 591
YAKAKÖY MAHALLESİ 1217
```

```
k
KURUDERE MAHALLESİ 61
SARNIÇKÖY MAHALLESİ 81
KAYADİBİ MAHALLESİ 252
ÇAMIÇİ MAHALLESİ 265
BEŞYOL MAHALLESİ 334
GÖKDERE MAHALLESİ 392
ÇİÇEKLİ MAHALLESİ 399
LAKA MAHALLESİ 432
KARAÇAM MAHALLESİ 592
YAKAKÖY MAHALLESİ 1217
```

3.a Heap Veri Yapısı Tasarlama

//3-a maddesi için yazmış olduğunuz kodları ve açıklamaları buraya ekleyiniz

HEAP SINIFININ YAPISI VE METODLARI

```
class Heap //HEAP VERİ YAPISI
{
    int maxSize;
    int currentSize = 0;
    MahalleSınıfı[] array;
    public Heap(int maxSize)
    {
        this.maxSize = maxSize;
        array = new MahalleSınıfı[maxSize];
    }
    public bool insert(MahalleSınıfı key) //İLGİLİ MAHALLEYİ EKLEYEN METOD
    {
        if (currentSize == maxSize)
        {
            return false;
        }
    }
}
```

```

    }
    array[currentSize] = key;

    trickleUp(currentSize++);
    return true;
}
public void trickleUp(int index) //TRICKLEUP METODU
{
    int parent = (index - 1) / 2;
    MahalleSınıfı bottom = array[index];
    while (index > 0 && array[parent].nüfussayisi < bottom.nüfussayisi)
    {
        array[index] = array[parent];
        index = parent;
        parent = (parent - 1) / 2;
    }
    array[index] = bottom;
}
public MahalleSınıfı remove() //İLGİLİ MAHALLEYİ SİLEN VE DÖNDÜREN METOD
{
    MahalleSınıfı root = array[0];
    array[0] = array[--currentSize];
    trickledown(0);
    return root;
}
public void trickledown(int index)// TRICKLEDOWN METODU
{
    int largerChild;
    MahalleSınıfı top = array[index];
    while (index < currentSize / 2)
    {
        int leftChild = 2 * index + 1;
        int rightChild = leftChild + 1;
        if (rightChild < currentSize && array[leftChild].nüfussayisi <
array[rightChild].nüfussayisi)
        {
            largerChild = rightChild;
        }
        else
        {
            largerChild = leftChild;
        }

        if (top.nüfussayisi >= array[largerChild].nüfussayisi)
        {
            break;
        }
        array[index] = array[largerChild];
        index = largerChild;
    }
    array[index] = top;
}
}

```

3.b Max Heap düğüm yerleştirme

//3-b maddesi için yazmış olduğunuz kodları ve açıklamaları buraya ekleyiniz

```
////////////////////////////////////  
EKLEME SIRASINDA KULLANILAN METODLAR
```

```
    public bool insert(MahalleSınıfı key) //İLGİLİ MAHALLEYİ EKLEYEN METOD  
    {  
        if (currentSize == maxSize)  
        {  
            return false;  
        }  
        array[currentSize] = key;  
  
        trickleUp(currentSize++);  
        return true;  
    }  
    public void trickleUp(int index) //TRICKLEUP METODU  
    {  
        int parent = (index - 1) / 2;  
        MahalleSınıfı bottom = array[index];  
        while (index > 0 && array[parent].nüfussayisi < bottom.nüfussayisi)  
        {  
            array[index] = array[parent];  
            index = parent;  
            parent = (parent - 1) / 2;  
        }  
        array[index] = bottom;  
    }  
}   
////////////////////////////////////  
  
    static string[] mahalleler = { "KURUDERE MAHALLESİ", "SARNIÇKÖY MAHALLESİ",  
    "KAYADİBİ MAHALLESİ", "ÇAMIÇI MAHALLESİ", "BEŞYOL MAHALLESİ", "GÖKDERE MAHALLESİ",  
    "ÇİÇEKLİ MAHALLESİ", "LAKA MAHALLESİ", "KARAÇAM MAHALLESİ", "YAKAKÖY MAHALLESİ"  
}; //MAHALLELERİN TUTULDUĞU LİSTE  
    static int[] nüfus = { 59, 81, 250, 265, 334, 392, 399, 432, 590, 1217 };  
//NÜFUSLARIN TUTULDUĞU LİSTE  
  
    Heap MahallelerVeNüfus = new Heap(50); //Heap sınıfı tipinde mahalleler ve  
nüfus oluşturulur  
    for(int i = 0; mahalleler.Length > i; i++)  
    {  
        MahalleSınıfı eklenecekmahallevenüfus = new MahalleSınıfı(nüfus[i],  
mahalleler[i]); //MAHALLE SINIFI TİPİNDE eklenecekmahallevenüfus oluşturulur  
        MahallelerVeNüfus.insert(eklenecekmahallevenüfus); //İLGİLİ MAHALLEVENÜFUS  
HEAP VERİ YAPISINA EKLENİR  
        Console.WriteLine(eklenecekmahallevenüfus.Mahalleadi + " " +  
eklenecekmahallevenüfus.nüfussayisi); //MAHALLE ADI VE NÜFUS SAYISI YAZDIRILIR  
    }  
}
```


>>> OLUŞTURULAN HEAP VERİ YAPISININ ELEMANLARI

```

public MahalleSınıfı remove() //İLGİLİ MAHALLEYİ SİLEN VE DÖNDÜREN METOD
{
    MahalleSınıfı root = array[0];
    array[0] = array[--currentSize];
    trickledown(0);
    return root;
}

public void trickledown(int index)// TRICKLEDOWN METODU
{
    int largerChild;
    MahalleSınıfı top = array[index];
    while (index < currentSize / 2)
    {
        int leftChild = 2 * index + 1;
        int rightChild = leftChild + 1;
        if (rightChild < currentSize && array[leftChild].nüfussayisi <
array[rightChild].nüfussayisi)
        {
            largerChild = rightChild;

```

```

    }
    else
    {
        largerChild = leftChild;
    }
    if (top.nüfussayisi >= array[largerChild].nüfussayisi)
    {
        break;
    }
    array[index] = array[largerChild];
    index = largerChild;
}
array[index] = top;
}

////////////////////////////////////

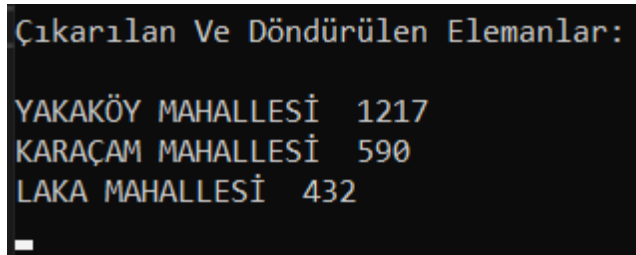
for (int i = 0; 3>i; i++)
{
    MahalleSınıfı ÇıkanEleman = MahallelerVeNüfus.remove(); //HEAPTEN ELEMEN
ÇIKARILIR VE DÖNDÜRÜLÜR
    Console.WriteLine(ÇıkanEleman.Mahalleadi + " " + ÇıkanEleman.nüfussayisi
); //DÖNDÜRÜLEN ELEMEN KONSOLA YAZDIRILIR

}

```

3.c.2 Ekran görüntüleri

//3-c maddesi için üretilen konsol/ekran görüntüsünü buraya ekleyiniz



4.a Simple sorting veya Advanced sorting algoritması

//4-a maddesi için yazmış olduğunuz kodları ve açıklamaları buraya ekleyiniz

```

class Program
{
    public static void BubbleSort(int[] siralanacakDizi) //BUBBLE SORT METODU
    {
        int i = 1, j, deger;
        int diziAdet = siralanacakDizi.Length;
        while (i < diziAdet)
        {
            j = diziAdet - 1;
            while (j >= 1)
            {

```

```

        if (siralanacakDizi[j - 1] > siralanacakDizi[j])//EĞER SOLDAKİ ELEMAN
SAĞDAN BÜYÜKSE
        {
            deger = siralanacakDizi[j];//KÜÇÜK OLAN ELEMANIN DEĞERİ ALINIR
            siralanacakDizi[j] = siralanacakDizi[j - 1];//BÜYÜK OLAN ELEMAN
KÜÇÜK OLAN ELEMANIN YERİNE YAZILIR
            siralanacakDizi[j - 1] = deger;//KÜÇÜK OLAN, BÜYÜK OLAN ELEMANIN
ESKİ YERİNE YAZILIR
            // KISACA KÜÇÜK OLAN ELEMANLA BÜYÜK OLAN ELEMANLARIN YERLERİ LİSTE
ÜZERİNDE DEĞİŞTİRİLİR
        }
        j--;
    }
    i++;
}

static void Main(string[] args)
{
    int[] sayilar = { 7, 5, 9, 1, 6, 4, 3, 2, 18, 95, 65 }; //SIRALANACAK LİSTE
    BubbleSort(sayilar);//LİSTE BUBBLE SORT'A GÖRE SIRALANIR
    for (int i = 0; sayilar.Length > i; i++)
    {
        Console.WriteLine(sayilar[i]);//SIRALANAN LİSTE EKRANA YAZDIRILIR
    }
    Console.ReadKey();
}
}

```

Temel mantığımız her diziyi baştan sona gezdiğimizde bir elemanı doğru bir şekilde sıralamış olacağız. Yani her elemanı doğru sıralamak için diziyi uzunluğu kadar gezmemiz gerekmekte. Metodumuzun içindeki while döngüleri bu işe yaramakta

```
int[] sayilar = { 7, 5, 9, 1, 6, 4, 3, 2, 18, 95, 65 }; //SIRALANACAK LİSTE
```

Sıralanacak olan listemizin elemanları.

▲ siralanacakDizi	{int[11]}	int[]
☞ [0]	7	int
☞ [1]	5	int
☞ [2]	9	int
☞ [3]	1	int
☞ [4]	6	int
☞ [5]	4	int
☞ [6]	3	int
☞ [7]	2	int
☞ [8]	18	int
☞ [9]	95	int
☞ [10]	65	int
☞ i	1	int
☞ j	10	int

i değeri while döngüsünün kontrolü için j değeri ise hem while döngüsünün kontrolü hem de sıralanacak olan listenin elemanlarına erişebilmek için oluşturulmuştur.

```
while (i < diziAdet)
{
    j = diziAdet - 1;
    while (j > 0)
    {
        if (siralanacakDizi[j - 1] > siralanacakDizi[j])//EĞER SOLDAKİ ELEMEN SAĞDAN BÜYÜKSE
        {
            deger = siralanacakDizi[j];//KÜÇÜK OLAN ELEMENİN DEĞERİ ALINIR
            siralanacakDizi[j] = siralanacakDizi[j - 1];//BÜYÜK OLAN ELEMEN KÜÇÜK OLAN ELEMENİN YERİNE YAZILIR
            siralanacakDizi[j - 1] = deger;//KÜÇÜK OLAN, BÜYÜK OLAN ELEMENİN ESKİ YERİNE YAZILIR
            // KISACA KÜÇÜK OLAN ELEMENLA BÜYÜK OLAN ELEMENLARIN YERLERİ LİSTE ÜZERİNDE DEĞİŞTİRİLİR
        }
        j--;
    }
    i++;
}
```

J değeri 10 olduğundan dolayı if bloğunda sıralanacak olan dizinin 9. Elemanı yani 95 , 10. Elemanı yani 65 ten büyükse;

deger	65	int
-------	----	-----

deger değişkenine 65 atanır.

[9]	95
[10]	95

Büyük olan eleman küçük olan elemanın yerine yazılır.

[9]	65
[10]	95

Küçük olan eleman büyük olan elemanın yerine yazılır.

Metod böyle ikili baloncuklar halinde listenin tüm elemanlarını kontrol ederek sıralamayı gerçekleştirir

4.b Sıralama algoritması ile karşılaştırılması

//4-b maddesi için açıklamalarınızı buraya yazınız

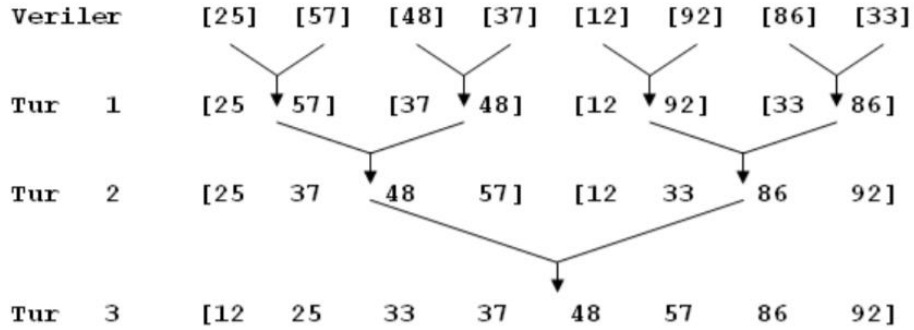
En temel fark algoritmanın zaman ve hafıza karmaşıklığı arasındaki farklardır. Verimli algoritmalar $O(N \log N)$ zamanda çalışırken verimsiz algoritmalar $O(N^2)$ zamanında çalışır.

Yukarıdaki örneğimde kullandığım sıralama Bubble sorttur. Bubble sort $O(N^2)$ zamanda çalışan verimsiz bir algoritmadır.

Karşılaştıracığım algoritma ise Merge Sorttur. Merge sort $O(N \log N)$ çalışan verimli bir algoritmadır.

Merge Sort kısaca

Merge Sort'un Zaman Karmaşıklığı



- **Analiz : $\log_2 n$ tur ve her turda n veya daha az karşılaştırma.**

Veriler parçalara ayrılır ve tekrar birleştirilip sıralı bir şekilde bir araya getirilir

Sıralama algoritmalarının çalışma hızları, sıralanacak verinin büyüklüğü, kısmen sıralı olması, tersten sıralı olması veya tümüyle karışık yapıda olmasına bağlı olarak farklılık göstermektedir. Seçim yapılırken verinin bu anlamda değerlendirilmesi ve uygun algoritmanın tercihi en iyi sonuç için faydalı olacaktır.

Eğer liste bize sıralı bir şekilde gelirse (ki bu en iyi senaryo olmaktadır) ,Bubble sort sıralaması Merge Sorta Göre daha avantajlı olur. Fakat liste En kötü durumda karşımıza çıkarsa Merge Sort sıralaması bizim için daha avantajlı olur

Algoritma	En iyi senaryo	Ortalama durum	En kötü durum
Bubblesort (kabarcık sıralama)	$O(n)$	$O(n^2)$	$O(n^2)$
Selection sort (seçmeli sıralama)	$O(n^2)$	$O(n^2)$	$O(n^2)$
Insertion sort (eklemeli sıralama)	$O(n)$	$O(n^2)$	$O(n^2)$
Quick sort (hızlı sıralama)	$O(n \log(n))$	$O(n \log(n))$	$O(n^2)$
Merge sort (Birleştirmeli Sıralama)	$O(n \log(n))$	$O(n \log(n))$	$O(n \log(n))$

Eğer Sıralamamız gereken liste çok fazla elemana sahipse Merge Sort Kullanımı bizim için daha iyi olur çünkü çok daha kısa sürede bu işlemi gerçekleştirebilir.

ALGORİTMA	10000 KAYIT	100000 KAYIT	1000000 KAYIT
Bubblesort (Kabarcık sıralama)	241 ms	29,2 s	48.5 dak
Selection sort (seçmeli sıralama)	33 ms	1,9 s	2,9 dakika
Insertion sort (eklemeli sıralama)	1 ms	4 ms	12 ms
Quick sort (hızlı sıralama)	2 ms	23 ms	34 ms
Merge sort (Birleştirmeli Sıralama)	5 ms	26 ms	137 ms

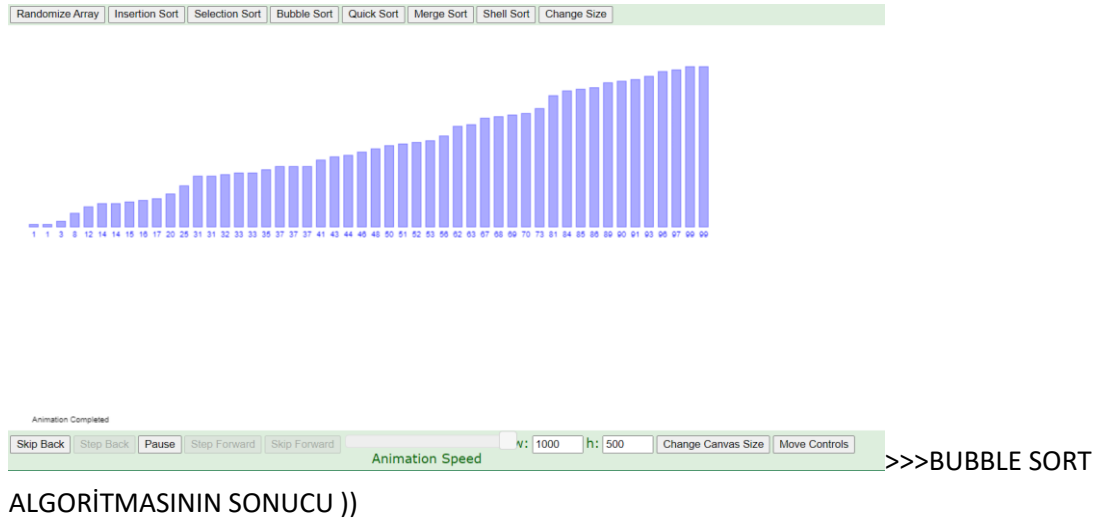
Fakat Merge Sort'ta atama işlemleri daha fazla ve aux dizi için daha fazla yer gerekmekte.

4.c Görselleştirme araçları

//4-c maddesi için açıklamanızı buraya yazınız

Algoritmaları görselleştirme videolarını izledikten sonra şunun farkına vardım ki bu videolar algoritmaları daha iyi kavramama olanak sağladı projemde oluşturduğum algoritmalar dışında daha kapsamlı daha kompleks algoritmaların sıralanışını videoyla kısa sürede izlemek debug yapıp kontrol etmekten çok daha etkili geldi. Özellikle sondaki Bogo sort algoritması çok dikkatimi çekti en iyi durum senaryosu olarak $O(1)$ olmasına rağmen bütün dizideki elemanları rastgele yerlere atayarak her seferinde bir defada sıralı olarak buldurmaya çalışması yüzümde bir tebessüm ettirdi.

Veri yapılarını öğrenme açısından videolar ile etkileşimli görselleştirme araçlarını karşılaştıracak olursam kesinlikle etkileşimli görselleştirme araçlarını seçerim .çünkü etkileşimli görselleştirme araçları liste üzerinde ekleme çıkarma yerlerini değiştirme önceki hamleye geçme sonraki hamleye geçme gibi adımları teker teker incelememe olanak sağlıyor ayrıca kendi listemi oluşturup o liste üzerinde yapacağım sıralama çeşidini dilediğim gibi seçebiliyorum. Sıralama dışında bir çok konuyuda içinde barındırıyor (Heap , recursive graph algorithms) ayrıca bu yapılar içinde listedeki gibi oynama yapılabilmekte



ALGORİTMASININ SONUCU))

Özdeğerlendirme Tablosu

Özdeğerlendirme Tablosu

Proje 3 Maddeleri	Not	Tahmini Not	Açıklama
1 a) Ağaç (Yemek Sipariş Ağacı Oluşturma)	10	10	Yapıldı
1 b) Derinlik Bulma, Ağacı Listeleme	10	10	Yapıldı
1 c) Arama ve Listeleme	10	10	Yapıldı
1 d) Ağaçta bir yemek türüne indirim yapma	10	10	Yapıldı
2) Hash Tablosu	15	15	Yapıldı

3) Yığın Ağacı (Heap)	15	15	Yapıldı
4) Sıralama Algoritmaları	20	20	Yapıldı
5) Özdeğerlendirme Tablosu	10	10	Yapıldı
Toplam	100	100	

Açıklama kısmında yapıldı, yapılmadı bilgisi ve hangi maddelerin nasıl yapıldığı (ve nelerin yapılmadığı / yapılamadığı) yazılmalıdır. Tahmini not kısmına da ilgili maddeden kaç almayı beklediğinizi yazmalısınız.