



EGE UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

204 DATA STRUCTURES (3+1)

2023–2024 FALL SEMESTER

PROJECT-2 REPORT

LİSTE, YIĞIT, KUYRUK ve ÖNCELİKLİ KUYRUK VERİ YAPILARI

DELIVERY DATE

--/--/---

PREPARED BY

05220001001

Mahir Can Yalın

İçindekiler

1) UNESCO Dünya Mirası Listesi	3
1.a UM Alanı Sınıfı Oluşturma	3
1.a.1 Kodlar	3
1.a.2 Açıklama	3
1.b Yedi Elemanlı Bir Generic List Dizisi Oluşturma	4
1.b.1 Kodlar	4
1.c Dizideki Her Bir Listedeki Bilgilerin ve Her Bir Bölgedeki UM Alanı Sayısını Yazdırma	6
1.c.1 Kodlar	6
1.c.2 Ekran görüntüleri	9
2) YIĞIT ve KUYRUK	9
2.a UM Alanı Yığıtı Oluşturma	9
2.a.1 Yığıt Sınıfı Kaynak Kodu	9
2.a.2 Yığıt Kullanımı Kaynak Kodu	9
2.a.3 Ekran Görüntüleri	9
2.b UM Alanı Kuyruk Veri Yapısı Oluşturma	10
2.b.1 Kuyruk Sınıfı Kaynak Kodu	10
2.b.2 Kuyruk Kullanımı Kaynak Kodu	12
2.b.3 Ekran Görüntüleri	12
3) Öncelikli Kuyruk	14
3.a Öncelikli Kuyruk (sınıfı) Oluşturma	14
3.a.1 Öncelikli Kuyruk Sınıfı Kaynak Kod	14
3.a.2 Öncelikli Kuyruk Kullanımı Kaynak Kodu	16
3.a.3 Ekran görüntüleri	16
4) Kuyruk ve Öncelikli Kuyruk	19
4.a Kuyruk kullanarak süre hesaplanması	19
4.a.1 Kaynak Kod	19
4.a.2 Ekran görüntüleri	21
4.b Öncelikli Kuyruk sınıfı kullanarak süre hesaplanması	23
4.b.1 Güncellenmiş Öncelikli Kuyruk Kaynak Kod	23
4.b.2 Öncelikli Kuyruk Süre Hesaplaması Kaynak Kod	25
4.b.3 Ekran görüntüleri	27
4.c Kuyruk ve Öncelikli Kuyruk Yapılarının Karşılaştırılması	27
4.c.1 Kuyruk ve Öncelikli Kuyruk Karşılaştırması	27
Öz değerlendirme Tablosu	29

1) UNESCO Dünya Mirası Listesi

//The platform, version, and programming language used

1.a UM Alanı Sınıfı Oluşturma

1.a.1 Kodlar

```
class UM_Alani
{
    public string Alan_Adi { get; set; }
    public List<string> İl_Adları { get; set; }
    public int İlan_Yılı { get; set; }

    public UM_Alani(string alan_Adi, List<string> il_Adları, int ilan_Yılı)
    {
        Alan_Adi = alan_Adi;
        İl_Adları = il_Adları;
        İlan_Yılı = ilan_Yılı;
    }
}
```

1.a.2 Açıklama

Classı oluşturduk ve gerekli propları ekledik.

1.b Yedi Elemanlı Bir Generic List Dizisi Oluřturma

1.b.1 Kodlar

```

List<UM_Alani> unescoAlanlari = new List<UM_Alani>();

unescoAlanlari.Add(new UM_Alani("Göreme Millî Parkı ve Kapadokya", new List<string>
{ "Nevşehir" }, 1985));

unescoAlanlari.Add(new UM_Alani("Hattuşa: Hitit Başkenti", new List<string> { "Çorum" },
1986));

unescoAlanlari.Add(new UM_Alani("Nemrut Dağı", new List<string> { "Adıyaman" }, 1987));

unescoAlanlari.Add(new UM_Alani("Hieropolis-Pamukkale", new List<string> { "Denizli" },
1988));

unescoAlanlari.Add(new UM_Alani("Xanthos-Letoon", new List<string> { "Antalya", "Muğla" },
1988));

unescoAlanlari.Add(new UM_Alani("Safranbolu Şehri", new List<string> { "Karabük" }, 1994));

unescoAlanlari.Add(new UM_Alani("Truva Arkeolojik Alanı", new List<string> { "Çanakkale" },
1998));

unescoAlanlari.Add(new UM_Alani("Edirne Selimiye Camii ve Külliyesi", new List<string>
{ "Edirne" }, 2011));

unescoAlanlari.Add(new UM_Alani("Çatalhöyük Neolitik Alanı", new List<string> { "Konya" },
2012));

unescoAlanlari.Add(new UM_Alani("Bursa ve Cumalıkızık: Osmanlı İmparatorluğunun Doğuşu",
new List<string> { "Bursa" }, 2014));

unescoAlanlari.Add(new UM_Alani("Bergama Çok Katmanlı Kültürel Peyzaj Alanı", new
List<string> { "İzmir" }, 2014));

unescoAlanlari.Add(new UM_Alani("Diyarbakır Kalesi ve Hevsel Bahçeleri Kültürel Peyzajı",
new List<string> { "Diyarbakır" }, 2015));

unescoAlanlari.Add(new UM_Alani("Efes", new List<string> { "İzmir" }, 2015));

unescoAlanlari.Add(new UM_Alani("Ani Arkeolojik Alanı", new List<string> { "Kars" }, 2016));

unescoAlanlari.Add(new UM_Alani("Aphrodisias", new List<string> { "Aydın" }, 2017));

unescoAlanlari.Add(new UM_Alani("Göbekli Tepe", new List<string> { "Şanlıurfa" }, 2018));

unescoAlanlari.Add(new UM_Alani("Arslantepe Höyüğü", new List<string> { "Malatya" },
2021));

unescoAlanlari.Add(new UM_Alani("Gordion", new List<string> { "Ankara" }, 2023));

unescoAlanlari.Add(new UM_Alani("Anadolu'nun Ortaçağ Dönemi Ahşap Hipostil Camiileri",
new List<string> { "Konya", "Kastamonu", "Eskişehir", "Afyon", "Ankara" }, 2023));

```

1.c Dizideki Her Bir Listedeki Bilgilerin ve Her Bir Bölgedeki UM Alanı Sayısını Yazdırma

1.c.1 Kodlar

```

foreach (var bölge in bölgeler)
{
    string bölgeAdı = bölge.Key;

    List<UM_Alani> alanlarBölgeyeAit = new List<UM_Alani>();

    foreach (var alan in unescoAlanlari)
    {
        foreach (var il in alan.İl_Adları)
        {
            if (bölge.Value.Contains(il))
            {
                alanlarBölgeyeAit.Add(alan);

                break;
            }
        }
    }

    AlanGrup.Add(bölgeAdı, alanlarBölgeyeAit);
}

AlanlariListele(AlanGrup);

Queue<UM_Alani> umAlanlariKuyruk = new Queue<UM_Alani>();

// unescoAlanlari listesinden UM_Alani nesnelerini kuyruğa ekleme
foreach (var alan in unescoAlanlari)
{
    umAlanlariKuyruk.Ekle(alan);
}

```

--- İç Anadolu Bölgesi ---

Toplam 4 UNESCO alanı bulunuyor:

Alan Adı: Göreme Millî Parkı ve Kapadokya

İl Adları: Nevşehir

İlan Yılı: 1985

Alan Adı: Çatalhöyük Neolitik Alanı

İl Adları: Konya

İlan Yılı: 2012

Alan Adı: Gordion

İl Adları: Ankara

İlan Yılı: 2023

Alan Adı: Anadolu'nun Ortaçağ Dönemi Ahşap Hipostil Camiileri

İl Adları: Konya, Kastamonu, Eskişehir, Afyon, Ankara

İlan Yılı: 2023

--- Karadeniz Bölgesi ---

Toplam 4 UNESCO alanı bulunuyor:

Alan Adı: Hattuşa: Hitit Başkenti

İl Adları: Çorum

--- Marmara Bölgesi ---

Toplam 3 UNESCO alanı bulunuyor:

Alan Adı: Truva Arkeolojik Alanı

İl Adları: Çanakkale

İlan Yılı: 1998

Alan Adı: Edirne Selimiye Camii ve Külliyesi

İl Adları: Edirne

İlan Yılı: 2011

Alan Adı: Bursa ve Cumalıkızık: Osmanlı İmparatorluğunun Doğuşu

İl Adları: Bursa

İlan Yılı: 2014

--- Akdeniz Bölgesi ---

Toplam 1 UNESCO alanı bulunuyor:

Alan Adı: Xanthos-Letoon

İl Adları: Antalya, Muğla

İlan Yılı: 1988

--- Doğu Anadolu Bölgesi ---

Toplam 2 UNESCO alanı bulunuyor:

Alan Adı: Ani Arkeolojik Alanı

İl Adları: Kars

İlan Yılı: 2016

Alan Adı: Arslantepe Höyüğü

İl Adları: Malatya

İlan Yılı: 2021

1.c.2 Ekran görüntüleri

2) YIĞIT ve KUYRUK

//The platform, version, and programming language used

2.a UM Alanı Yığıt Oluşturma

2.a.1 Yığıt Sınıfı Kaynak Kodu

//İlgili kod parçasını buraya ekleyiniz

2.a.2 Yığıt Kullanımı Kaynak Kodu

//İlgili kod parçasını buraya ekleyiniz

2.a.3 Ekran Görüntüleri

//Konsol çıktısına ait ekran görüntülerini buraya ekleyiniz

2.b UM Alanı Kuyruk Veri Yapısı Oluřturma

2.b.1 Kuyruk Sınıfı Kaynak Kodu

```

class Queue<T>
{
    private List<T> elements = new List<T>();

    public bool Bosmu()
    {
        return elements.Count == 0;
    }

    public void Ekle(T item)
    {
        elements.Add(item);
    }

    public T Sil()
    {
        if (Bosmu())
        {
            throw new InvalidOperationException("Kuyruk boş.");
        }

        T item = elements[0];
        elements.RemoveAt(0);
        return item;
    }

    public T Peek()
    {
        if (Bosmu())
        {

```

2.b.2 Kuyruk Kullanımı Kaynak Kodu

```
Queue<UM_Alani> umAlanlariKuyruk = new Queue<UM_Alani>();
```

```
// unescoAlanlari listesinden UM_Alani nesnelerini kuyruğa ekleme
```

```
foreach (var alan in unescoAlanlari)
```

```
{
```

```
    umAlanlariKuyruk.Ekle(alan);
```

```
}
```

```
KuyrukListele(umAlanlariKuyruk);
```

```
static void KuyrukListele(Queue<UM_Alani> Queue)
{
    Console.WriteLine("Kuyruk İle Listelenen Um_Alani Bilgileri:");
    while (!Queue.Bosmu())
    {
        UM_Alani alan = Queue.Sil();
        Console.WriteLine($"Alan Adı: {alan.Alan_Adi}");
        Console.WriteLine($"İl Adları: {string.Join(", ", alan.İl_Adları)}");
        Console.WriteLine($"İlan Yılı: {alan.İlan_Yılı}");
        Console.WriteLine();
    }
}
```

2.b.3 Ekran Görüntüleri

Kuyruk İle Listelenen Uın Alanı Bilgileri:
Alan Adı: Göreme Millî Parkı ve Kapadokya
İl Adları: Nevşehir
İlan Yılı: 1985

Alan Adı: Hattuşa: Hitit Başkenti
İl Adları: Çorum
İlan Yılı: 1986

Alan Adı: Nemrut Dağı
İl Adları: Adıyaman
İlan Yılı: 1987

Alan Adı: Hieropolis-Pamukkale
İl Adları: Denizli
İlan Yılı: 1988

Alan Adı: Xanthos-Letoon
İl Adları: Antalya, Muğla
İlan Yılı: 1988

Alan Adı: Safranbolu Şehri
İl Adları: Karabük
İlan Yılı: 1994

Alan Adı: Truva Arkeolojik Alanı
İl Adları: Çanakkale
İlan Yılı: 1998

Alan Adı: Edirne Selimiye Camii ve Külliyesi
İl Adları: Edirne
İlan Yılı: 2011

Alan Adı: Çatalhöyük Neolitik Alanı
İl Adları: Konya
İlan Yılı: 2012

Alan Adı: Bursa ve Cumalıkızık: Osmanlı İmparatorluğunun Doğuşu
İl Adları: Bursa
İlan Yılı: 2014

Alan Adı: Bergama Çok Katmanlı Kültürel Peyzaj Alanı
İl Adları: İzmir
İlan Yılı: 2014

Alan Adı: Diyarbakır Kalesi ve Hevsel Bahçeleri Kültürel Peyzajı
İl Adları: Diyarbakır
İlan Yılı: 2015

3) Öncelikli Kuyruk

3.a Öncelikli Kuyruk (sınıfı) Oluşturma

3.a.1 Öncelikli Kuyruk Sınıfı Kaynak Kod

```

class PriorityQueue<Int>
{
    private List<Int> elements = new List<Int>();

    public void Ekle(Int item)
    {
        elements.Add(item);
    }

    public Int Sil()
    {
        if (Bosmu())
        {
            throw new InvalidOperationException("Kuyruk boş.");
        }

        Int enÖncelikliEleman = FindPriority();
        elements.Remove(enÖncelikliEleman);
        return enÖncelikliEleman;
    }

    private Int FindPriority()
    {
        Int Priority = default(Int);

        foreach (Int item in elements)
        {
            if (Priority == null || Karşılaştır(item, Priority) < 0)
            {

```

3.a.2 Öncelikli Kuyruk Kullanımı Kaynak Kodu

```
PriorityQueue<UM_Alani> oncelikliKuyruk = new PriorityQueue<UM_Alani>();
```

```
// UM_Alani nesnelerini öncelikli kuyruğa ekleme  
  
foreach (var alan in unescoAlanlari)  
{  
    oncelikliKuyruk.Ekle(alan);  
}  
  
OncelikliKuyrukListele(oncelikliKuyruk);  
  
HesaplaVeYazdir();  
}
```

```
static void OncelikliKuyrukListele(PriorityQueue<UM_Alani> Queue)  
{  
    Console.WriteLine("Öncelikli Kuyruk ile Listelenen Um_Alani Bilgileri:");  
    while (!Queue.Bosmu())  
    {  
        UM_Alani alan = Queue.Sil();  
        Console.WriteLine($"Alan Adı: {alan.Alan_Adi}");  
        Console.WriteLine($"İl Adları: {string.Join(", ", alan.İl_Adları)}");  
        Console.WriteLine($"İlan Yılı: {alan.İlan_Yılı}");  
        Console.WriteLine();  
    }  
}
```

3.a.3 Ekran görüntüleri

Alan Adı: Ani Arkeolojik Alanı

İl Adları: Kars

İlan Yılı: 2016

Alan Adı: Aphrodisias

İl Adları: Aydın

İlan Yılı: 2017

Alan Adı: Göbekli Tepe

İl Adları: Şanlıurfa

İlan Yılı: 2018

Alan Adı: Arslantepe Höyüğü

İl Adları: Malatya

İlan Yılı: 2021

Alan Adı: Gordion

İl Adları: Ankara

İlan Yılı: 2023

Alan Adı: Anadolu'nun Ortaçağ Dönemi Ahşap Hipostil Camileri

İl Adları: Konya, Kastamonu, Eskişehir, Afyon, Ankara

İlan Yılı: 2023

Öncelikli Kuyruk İle Listelenen Um Alanı Bilgileri:

Alan Adı: Anadolu'nun Ortaçağ Dönemi Ahşap Hipostil Camileri

İl Adları: Konya, Kastamonu, Eskişehir, Afyon, Ankara

İlan Yılı: 2023

Alan Adı: Ani Arkeolojik Alanı

İl Adları: Kars

İlan Yılı: 2016

Alan Adı: Aphrodisias

İl Adları: Aydın

İlan Yılı: 2017

Alan Adı: Arslantepe Höyüğü

İl Adları: Malatya

İlan Yılı: 2021

Alan Adı: Bergama Çok Katmanlı Kültürel Peyzaj Alanı

İl Adları: İzmir

İlan Yılı: 2014

Alan Adı: Bursa ve Cumalıkızık: Osmanlı İmparatorluğunun Doğuşu

İl Adları: Bursa

İlan Yılı: 2014

Alan Adı: Çatalhöyük Neolitik Alanı

İl Adları: Konya

İlan Yılı: 2012

Alan Adı: Diyarbakır Kalesi ve Hevsel Bahçeleri Kültürel Peyzajı

İl Adları: Diyarbakır

İlan Yılı: 2015

Alan Adı: Edirne Selimiye Camii ve Külliyesi

İl Adları: Edirne

İlan Yılı: 2011

Alan Adı: Efes

İl Adları: İzmir

İlan Yılı: 2015

Alan Adı: Gordion

İl Adları: Ankara

İlan Yılı: 2023

Alan Adı: Göbekli Tepe

İl Adları: Şanlıurfa

İlan Yılı: 2018

Alan Adı: Göbekli Tepe Neolitik Alanı

Alan Adı: Hattuşa: Hitit Başkenti

İl Adları: Çorum

İlan Yılı: 1986

Alan Adı: Hieropolis-Pamukkale

İl Adları: Denizli

İlan Yılı: 1988

Alan Adı: Nemrut Dağı

İl Adları: Adıyaman

İlan Yılı: 1987

Alan Adı: Safranbolu Şehri

İl Adları: Karabük

İlan Yılı: 1994

Alan Adı: Truva Arkeolojik Alanı

İl Adları: Çanakkale

İlan Yılı: 1998

Alan Adı: Xanthos-Letoon

İl Adları: Antalya, Muğla

İlan Yılı: 1988

4) Kuyruk ve Öncelikli Kuyruk

4.a Kuyruk kullanarak süre hesaplanması

4.a.1 Kaynak Kod

class TamSayiKuyrugu

```
{  
  
    private List<int> elements = new List<int>();  
  
  
    public bool Bosmu()  
    {  
        return elements.Count == 0;  
    }  
  
    public void Ekle(int item)  
    {  
        elements.Add(item);  
    }  
  
  
    public int Sil()  
    {  
        if (Bosmu())  
        {  
            throw new InvalidOperationException("Kuyruk boş.");  
        }  
  
        int item = elements[0];  
        elements.RemoveAt(0);  
        return item;  
    }  
  
  
    public int Count()
```

```
{  
    return elements.Count;  
}  
  
public int Peek()  
{  
    if (Bosmu())  
    {  
        throw new InvalidOperationException("Kuyruk boş.");  
    }  
  
    return elements[0];  
}  
  
public void Clear()  
{  
    elements.Clear();  
}  
}
```

```

static void HesaplaVeYazdir()
{
    TamSayiKuyrugu tamSayiKuyruk = new TamSayiKuyrugu();

    tamSayiKuyruk.Ekle(10);
    tamSayiKuyruk.Ekle(4);
    tamSayiKuyruk.Ekle(8);
    tamSayiKuyruk.Ekle(6);
    tamSayiKuyruk.Ekle(7);
    tamSayiKuyruk.Ekle(1);
    tamSayiKuyruk.Ekle(15);
    tamSayiKuyruk.Ekle(9);
    tamSayiKuyruk.Ekle(3);
    tamSayiKuyruk.Ekle(2);

    double toplamSure = 0;
    int kuyrukBoyutu = tamSayiKuyruk.Count();

    while (!tamSayiKuyruk.Bosmu())
    {
        int urunSayisi = tamSayiKuyruk.Sil();
        double islemSuresi = urunSayisi * 2.5;
        toplamSure = islemSuresi + toplamSure;

        Console.WriteLine($"{urunSayisi} ürünü olan müşterinin işlemi {islemSuresi} saniye kadar sürdü.");
    }

    double ortalamaSure = toplamSure / kuyrukBoyutu;

    Console.WriteLine($"Ortalama işlem tamamlanma süresi: {ortalamaSure} saniye");
}

```

4.a.2 Ekran görüntüleri

//Konsol çıktısına ait ekran görüntülerini buraya ekleyiniz

```
10 ürünü olan müşterinin işlemi 25 saniye kadar sürdü.
4 ürünü olan müşterinin işlemi 10 saniye kadar sürdü.
8 ürünü olan müşterinin işlemi 20 saniye kadar sürdü.
6 ürünü olan müşterinin işlemi 15 saniye kadar sürdü.
7 ürünü olan müşterinin işlemi 17,5 saniye kadar sürdü.
1 ürünü olan müşterinin işlemi 2,5 saniye kadar sürdü.
15 ürünü olan müşterinin işlemi 37,5 saniye kadar sürdü.
9 ürünü olan müşterinin işlemi 22,5 saniye kadar sürdü.
3 ürünü olan müşterinin işlemi 7,5 saniye kadar sürdü.
2 ürünü olan müşterinin işlemi 5 saniye kadar sürdü.
Ortalama işlem tamamlanma süresi: 16,25 saniye
1 ürün olan müşterinin işlemi 2,5 saniye sürdü.
2 ürün olan müşterinin işlemi 5 saniye sürdü.
3 ürün olan müşterinin işlemi 7,5 saniye sürdü.
4 ürün olan müşterinin işlemi 10 saniye sürdü.
6 ürün olan müşterinin işlemi 15 saniye sürdü.
7 ürün olan müşterinin işlemi 17,5 saniye sürdü.
8 ürün olan müşterinin işlemi 20 saniye sürdü.
9 ürün olan müşterinin işlemi 22,5 saniye sürdü.
10 ürün olan müşterinin işlemi 25 saniye sürdü.
15 ürün olan müşterinin işlemi 37,5 saniye sürdü.
Öncelikli Kuyruk ile ortalama işlem tamamlanma süresi: 16,25 saniye
```

4.b Öncelikli Kuyruk sınıfı kullanarak süre hesaplanması

4.b.1 Güncellenmiş Öncelikli Kuyruk Kaynak Kod

```

class PriorityQueueForNumbers
{
    private List<int> elements = new List<int>();

    public void Ekle(int item)
    {
        elements.Add(item);
    }

    public int Sil()
    {
        if (Bosmu())
        {
            throw new InvalidOperationException("Kuyruk boş.");
        }

        int enÖncelikliEleman = FindPriority();
        elements.Remove(enÖncelikliEleman);
        return enÖncelikliEleman;
    }

    private int FindPriority()
    {
        int enOncelikli = elements[0];

        foreach (int item in elements)
        {
            if (item < enOncelikli)
            {

```


4.b.2 Öncelikli Kuyruk Süre Hesaplaması Kaynak Kod

```
PriorityQueueForNumbers PriorityQueue = new PriorityQueueForNumbers();
```

```
    PriorityQueue.Ekle(10);
```

```
    PriorityQueue.Ekle(4);
```

```
    PriorityQueue.Ekle(8);
```

```
    PriorityQueue.Ekle(6);
```

```
    PriorityQueue.Ekle(7);
```

```
    PriorityQueue.Ekle(1);
```

```
    PriorityQueue.Ekle(15);
```

```
    PriorityQueue.Ekle(9);
```

```
    PriorityQueue.Ekle(3);
```

```
    PriorityQueue.Ekle(2);
```

```
    HesaplaVeYazdir(PriorityQueue);
```

```
}
```

```
static void HesaplaVeYazdir(PriorityQueueForNumbers Queue)
```

```
{
```

```
    int KuyrukBoyut = Queue.Count();
```

```
    double toplamSure = 0;
```

```
    while (!Queue.Bosmu())
```

```
    {
```

```
        int urunSayisi = Queue.Sil();
```

```
        double islemSuresi = urunSayisi * 2.5;
```

```
        toplamSure += islemSuresi;
```

```
        Console.WriteLine($"{urunSayisi} ürün olan müşterinin işlemi {islemSuresi} saniye sürdü.");
```

```
    }
```

4.b.3 Ekran görüntüleri

//Konsol çıktısına ait ekran görüntülerini buraya ekleyiniz

```
10 ürünü olan müşterinin işlemi 25 saniye kadar sürdü.
4 ürünü olan müşterinin işlemi 10 saniye kadar sürdü.
8 ürünü olan müşterinin işlemi 20 saniye kadar sürdü.
6 ürünü olan müşterinin işlemi 15 saniye kadar sürdü.
7 ürünü olan müşterinin işlemi 17,5 saniye kadar sürdü.
1 ürünü olan müşterinin işlemi 2,5 saniye kadar sürdü.
15 ürünü olan müşterinin işlemi 37,5 saniye kadar sürdü.
9 ürünü olan müşterinin işlemi 22,5 saniye kadar sürdü.
3 ürünü olan müşterinin işlemi 7,5 saniye kadar sürdü.
2 ürünü olan müşterinin işlemi 5 saniye kadar sürdü.
Ortalama işlem tamamlanma süresi: 16,25 saniye
1 ürün olan müşterinin işlemi 2,5 saniye sürdü.
2 ürün olan müşterinin işlemi 5 saniye sürdü.
3 ürün olan müşterinin işlemi 7,5 saniye sürdü.
4 ürün olan müşterinin işlemi 10 saniye sürdü.
6 ürün olan müşterinin işlemi 15 saniye sürdü.
7 ürün olan müşterinin işlemi 17,5 saniye sürdü.
8 ürün olan müşterinin işlemi 20 saniye sürdü.
9 ürün olan müşterinin işlemi 22,5 saniye sürdü.
10 ürün olan müşterinin işlemi 25 saniye sürdü.
15 ürün olan müşterinin işlemi 37,5 saniye sürdü.
Öncelikli Kuyruk ile ortalama işlem tamamlanma süresi: 16,25 saniye
```

4.c Kuyruk ve Öncelikli Kuyruk Yapılarının Karşılaştırılması

4.c.1 Kuyruk ve Öncelikli Kuyruk Karşılaştırması

Öncelikli Kuyruk (Priority Queue) ve Queue (Kuyruk) avantajlarını birleştiren bir yaklaşım, her iki senaryonun gereksinimlerini karşılamak için uygulanabilir. Bu yaklaşım, PQ'nun öncelikli işleri hızla işlemesini ve FIFO'nun beklemeyi minimize etmesini amaçlar.

Yöntemin Açıklaması:

1) İşler Öncelikle Öncelikli Kuyruğa Eklenir:

Gelen işler öncelikli bir kuyruğa eklenir.

Her işin bir öncelik seviyesi vardır ve bu seviyeye göre PQ'da sıralanır.

2) İşler Zamanla Öncelikleri Azalır:

İşler başlangıçta yüksek önceliğe sahip olsalar da, zamanla öncelikleri azalır.

Bir iş belirli bir süre boyunca bekledikçe veya sıraya girdikçe, önceliği azalır.

3) Öncelikli İşler Önce İşlenir:

PQ, öncelik sırasına göre işleri işler.

Ancak, bir iş belirli bir süre veya sıraya girdiği sıradan sonra önceliği artar.

4) FIFO İşler Sırayla İşlenir:

PQ'dan çıkartılmış veya önceliği azalmış işler, FIFO mantığına göre bir kuyruğa eklenir.

FIFO kuyruğu, PQ'dan çıkartılan işlerin sırayla işlenmesini sağlar.

5) Öncelik Sırası ve Sıra İşbirliği Yapar:

PQ, yüksek öncelikli işleri hızla işler.

FIFO kuyruğu, diğer işleri sırayla işler.

Öz deęerlendirme Tablosu

Proje 2 Maddeleri	Puan	Ta h m i n i Not	Açıklama
1 a) Sınıf Oluşturma	5	5	Sınıfları oluşturup özelliklerini eklemede sıkıntı yaşamadım
1 b) Bileşik veri yapısının oluşturulması ve elemanların eklenmesi; Kaynak kod, ekran görüntüsü.	15	15	Veri yapılarını oluşturdu m elemanlar ı ve çıktıları edinmede sıkıntı yaşamadım.
1 c) Veri yapısının elemanlarının listelenmesi ve her bölgedeki UM alanı sayısının yazdırılması; Kaynak kodlar, ekran görüntüsü.	10	10	Çıktılarda ve yöntemler i yazmada sıkıntı yaşamadım.Kodu ve çıktıları da düzenli bir biçimde yazdım.
2 a) Yığıt kaynak kod ve ekran görüntüleri	10	0	
2 b) Kuyruk kaynak kod ve ekran görüntüleri	10	10	Kuyruk yapısına iyi bi şekilde anladım ve bunu koda yansıtım

3) Öncelikli Kuyruk kod ve ekran görüntüleri	20	20	Öncelikle kuyruk yapısını iyi bir şekilde anladım ve bunu koda yansıtım
4) Kod, sonuçlar tablosu, ekran görüntüleri, karşılaştırma ve soruların cevapları.	20	15	Kodu anlaşılır ve düzenli bir biçimde yazdım ve sorulara cevap ürettim
5) Özdeğerlendirme Tablosu	10	10	Tabloyu gayet açık bir şekilde yaptım.
Toplam	100		

Açıklama kısmında yapıldı, yapılmadı bilgisi ve hangi maddelerin nasıl yapıldığı veya neden yapılamadığı kısaca yazılmalıdır. Tahmini not kısmına da ilgili maddeden kaç almayı beklediğinizi yazmalısınız.

Not: Raporu teslim edilmeyen projeler değerlendirmeye alınmayacaktır.