

# Yoğun İndeks (Dense Index) Yapısı Projesi

Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Ömer Emircan Ayvaz-200202009

Engin Tosun-200202028

## Özet

Bu rapor Programlama Laboratuvarı I Dersinin 1. Projesini açıklamak ve sunumunu gerçekleştirmek amacıyla oluşturulmuştur. Bu proje C dilinde Code:Blocks ortamında binary file ve text file kullanılarak geliştirilmiştir. Raporda projenin tanımı, özet, yöntem, karşılaşılan sorunlar ve çözümler, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmaktadır. Proje aşamasında yararlanılan kaynaklar raporun son bölümünde bulunmaktadır.

## Proje Tanımı

Bu projenin amacı C dilinde dosyalama fonksiyonlarının pratiğinin yapılmasıdır. Bu amaçla ikili dosyaların (binary file) ve metin dosyalarının (text file) kullanıldığı bir uygulama geliştirilecektir

Bu proje kapsamında veri dosyasında bir okuldaki öğrencilere ait notlar saklanacaktır. Her öğrenci için aşağıdaki struct yapısında bilgi saklanacaktır. Bu yapıda ogrNo, dersKodu ve puan alanları bulunmak zorundadır. Ancak projenizi gerçekleştirirken ihtiyaç duyabileceğiniz başka alanlar da tanımlayabilirsiniz. Bir öğrenci için birden çok kayıt olabileceğini göz önünde bulundurunuz. Veri dosyasının bir ikili dosya (binary file) şeklinde gerçekleştirilmelidir.

## Yöntem

Algoritma tasarlayıp projeye başlamadan önce dense index yapısı, dosya işlemleri ve pointerlar konuları lazım olabileceği için tekrar edildi. Proje ilerledikçe gerekli yerlerinde lazım olabilecek hazır fonksiyonlar araştırıldı ve kodda uygulandı (remove, fileno, filelength, chsize gibi). Dosyadaki verilerin ve oluşturulan dizilerin içindeki verilerin doğruluğu konsol üzerinden çokça yerde printf ile kontrol edildi ve hatalı durumlarda düzenlemelere gidildi.

İlk olarak projede istenilen struct yapısı oluşturuldu. OgrNo, dersKodu ve puan ile oluşan 3 integer tanımlandı ve bunlar daha sonra struct çağırırken daha efektif olması için typedef öğrenci şeklinde oluşturuldu. Devamında txt şeklinde index dosyası ve bin şeklinde veri dosyası tanımlandı, projede istenilen fonksiyonlar oluşturuldu. **Kayıtekle** fonksiyonunda bin dosyası olan veri için fwrite ve fread, txt dosyası olan index için fprintf ve scanf kullanıldı. Index dosyasını silme ve tekrar oluşturma durumlarında veri dosyasının içindeki çıktıları tekrar içine yazdırabilmek için **indexeaktar** adında bir fonksiyon oluşturuldu ve her işlem sonunda ve başında veri kaybı olmaması için kullanıldı.

**bin\_indexveri\_sirala** fonksiyonu **kayıtbul** fonksiyonunda binary search uygulamak için gerekli sıralı verileri sıralamak için kullanılan fonksiyondur. Bu fonksiyonun işlevleri veri dosyasının içindekileri matrise aktarıldığı için

matrisi bubble sort ile sıralama, index dosyasının içindekileri indexe aktarıldığı için arrayi bubble sort ile sıralamaktır. Kısaca bu fonksiyon binary search için gereken sıralanmış array şartı için yazıldı.

**index\_final\_Sirala** fonksiyonunun işlevi istenilen formatta index dosyasını göstermek için adres ve öğrenci no'yu sıralamaktır. Bubble sort kullanımı ile öğrencileri no'larını küçükten büyüğe sıralar, adresleri ise aynı öğrenci no ları için küçükten büyüğe sıralar.

Veri dosyasındaki verileri göstermek için **veriDosyasiniGoster** fonksiyonu oluşturduk, Bu fonksiyon veri dosyasındaki tüm kayıtları listeler.

Index file'ı ise **indexDosyasiniGoster** fonksiyonu içinde okuyup ekrana index dosyasını ve adreslerini sıralı bir şekilde gösterildi. Bu fonksiyon içinde 2 dinamik dizi tanımlandı while döngüsünde fread ile veri içindeki no ları ve adresleri arraye aktarıldı. Son olarak hem ekrana hem dosyaya yazdırmak için döngüde fscanf kullanıldı.

Ayrıca index file'ı ekrandan giriş yapıp silmek ve oluşturmak için **indexDosyasiniOlustur**, **indexDosyasiniSil** fonksiyonları oluşturuldu.. Oluşturmak için a+ komutunu indeks dosyasını diskten silmek için ise remove'u kullanıldı. **IndexDosyasiniOlustur** işlevi kodda fonksiyonlarda kullanılacak index.txt ve adres ve girdileri sıralı göreceğimiz index\_final dosyalarını oluşturur.

**KayitGuncelle** fonksiyonunda ise bir anahtar için aranılan kayıt bulunup sadece puan alanı güncellenmiştir. Girilen öğrenci no ve ders kodu için sağlanan kayıttan puanını değiştirme fonksiyonudur. Bu kısımda kullanıcıdan hem öğrenci no hem de ders kodu istenerek istenilen veriye ulaşılmıştır ve yeni puan kullanıcıdan istenmiştir. İşlevleri kayıt bulunup bulunmadığını kontrol etmek, bulunan değerleri yazdırma, puanını değiştirme işlemi, dosya pointerını fseek ile tekrar 1 struct geri götürme, son halini o kaydın üzerine yazma, kontrol değişkeni ile kayıt bulunamadıysa if koşulu ile kayıt bulunamadı yazdırmaktır.

**kayitSil** fonksiyonunun işlevi verilen bir anahtar için ilgili doğru kaydı bulup bu kaydı siler. **KayitGuncelle** ile aynı mantık çalışır. Silme işlemi indeks dosyasına da yansıtılmıştır. Kayıt silme fonksiyonunda da **kayitguncelle** gibi kullanıcıdan hem öğrenci no hem de ders kodu istenmiştir. değeri bulduktan sonra silinecek değer arkasındaki değerleri dosya pointerı kullanılarak tekrar yazdırma işlemi yapılmıştır.

**kayitBul** binary search kullanarak girilen öğrenci no için tüm kayıtları yazdırmak için oluşturduk. Verilen bir anahtar için veri dosyasındaki ilk kayıt gösterilmiştir. İlgili anahtar için birden çok kayıt bulunabileceği için girilen anahtara ait tüm kayıtlar listelendi. İndeks dosyasında anahtara ait ilk kaydın bulunmasında binary search algoritması kullandık. Fonksiyonun sırasıyla yaptıkları veri dosyasında her bir değeri oluşturulan index ve veri arraylerine aktarma, iki arraye de bubble sort uygulama fonksiyonu, binary search için değişken tanımlama, birden fazla aynı öğrenci no'dan olabileceği için kendinden önce ve sonraki indisleri kontrol etme ve istenilen anahtarı sağlayan kendinden önce ve sonra gelen değerleri yazdırma, tüm değerler yazıldıktan sonra break ile döngüden çıkma.

Son olarak **main** fonksiyonunda switch case ile kullanıcıdan seçim isteyerek hangi fonksiyonu çalıştırmak istiyorsa integer değeri olarak bu değer alındı. her case'in içine farklı bir fonksiyon atandı. Program birden fazla çalıştırılmak istenilebileceği için while(1) döngüsü içine alındı ve döngüden çıkıp programı sonlandırmak için exit komutu caselerden birine atandı.

#### Karşılaşılan Sorunlar ve Çözümleri

**Index Dosyasındaki Veri Eksikliği:** Index dosyasını sil ve yeniden oluştur fonksiyonlarını ekledikten sonra şöyle bir sorunla karşılaştık. Index dosyasının içine kayıt ekleme

sadece kayıtle fonksiyonu içinde olduğundan silip tekrar oluşturunca içine eski kayıtlar tekrar yazılmıyordu bunu engellemek için **indexeaktar** adında bir fonksiyon oluşturduk ve bunu her işlemimizden sonra ve önce kullandık.

**Kayıtbul Fonksiyonundaki Binary Search Algoritmasını Oluşturma:** Binary Search tek bir değeri bulmaya dayalı olduğundan dolayı çoklu çıktı verebilen bu fonksiyona uygun bir yapısı yoktu.Bunun önüne geçmek için değeri veride bulduğunda break ile çıkmasını silip içine while döngüsü içinde aranan değer sağladığı sürece yazdırma işlemi ekledik.

**Binary Search'de Çıktıların Eksik Yazılması:** Binary Search çalışırken bulunan değer her zaman sıralı verilen dizideki ilk değer olmadığından bazı durumlarda çıktıyı eksik veriyordu.Örnek olarak 3 tane 5 öğrenci no'suna ait kayıt varsa eğer ilk bulduğu değer ortadaki 5 ise kodumuzun şeklinden dolayı ortadaki ve sondaki 5 i yazdırıp fonksiyonu sonlandırıyordu bunu önlemek için sonrasındaki değerler gibi sağladığı kendinden önce gelen her değer için sayaç kullanıp bunu küçükten büyüğe sıralamak için başka bir döngü içerisinde yazdırdık.

#### Pseudo Code(Sözde Kod)

oluştur(index dosyası)

yaz(1- Kayıt Ekle)

yaz(2- Veri Dosyasını Göster)

yaz(3- Yeniden Index Dosyası Oluştur)

yaz(4- Index Dosyasını Sil)

yaz(5- Index Dosyasini Goster)

yaz(6- Kayıt Güncelle)

yaz(7- Kayıt Sil)

yaz(8- Kayıt Bul)

yaz(9- Çıkış)

yaz(bir seçim yapınız)

oku(seçim)

\_\_seçim 1 ise;

#### Kayıt Ekle:

**void kayitEkle(ogrenci \*s1)**

tanımla \*dosya\_index,\*dosya\_veri  
dosya\_index(index.txt,r+),dosya\_veri(veri.bin,  
r+b)

yazdır (öğrenci no,ders kodu,puan)  
kullanıcıdan iste (öğrenci no,ders kodu,puan)  
dosya\_veri pointerını en sona ayarla

If(fwrite(&s1,dosya\_veri)  
dosya\_index pointerını en sona ayarla  
dosya\_index içine öğrenci no'yu yazdır  
else

yazdır(kayıt eklenemedi)  
dosyakapat(dosya\_index)  
dosyakapat(dosya\_veri)

\_\_seçim 2 ise;

#### Veri Dosyasını Göster:

**void veriDosyasiniGoster(ogrenci \*s1)**

tanımla \*dosya,kontrol=1  
dosya(veri.bin,rb)

While(fread(&s1,dosya)

If(öğrenci no !=0)

if(kontrol==1)

yazdır(öğrenci no ,ders kodu ,puan)

kontrol=0

yazdır(s1->ogrNo,s1->dersKodu,s1->puan)

If(kontrol==1)

Yazdır(veri dosyası bulunamadı)

dosyakapat(dosya)

\_seçim 3 ise;

#### **Yeniden Index Dosyası Oluştur:**

##### **void indexDosyasiniOluştur()**

tanımla \*dosya,\*dosya\_final  
dosya(index.txt, a+)

dosya\_final(index\_final.txt,a+)

dosyakapat(index)  
dosyakapat(index\_final)

\_seçim 4 ise;

#### **Index Dosyasını Sil:**

##### **void indexDosyasiniSil()**

if (remove(index.txt)==0)

    If(remove(index\_final.txt)==0)

        Yazdır(index dosyası başarıyla silindi)

Else

    Yazdır(index dosyası silinemedi)

\_seçim 5 ise;

#### **Index Dosyasini Goster:**

##### **void indexDosyasiniGoster(ogrenci \*s1)**

Tanımla \*dosya\_veri,\*dosya\_index

Tanımla \*dosya\_index\_final

tanımla dinamik array\_no, array\_adres

Tanımla n=1,i=0

dosya\_index(index.txt, r+)

dosya\_veri(veri.bin,rb)

dosya\_final(index\_final.txt,r+)  
dosya\_index pointer başa getir  
while(fread(&s1,dosya\_veri)  
    array\_no=öğrenci no,n++

Array\_adres=adres

dosyaKapat(dosya\_veri)

If(n>1)

    Index\_final\_sirala(array\_no,(n-1),array\_adres)

While(fscanf(dosya\_index,%d,ogrenci no)==1)

    If(öğrenci no !=0)

if(kontrol==1)

    yazdır(öğrenci no ,adres)

    kontrol=0

yazdır(array\_no[i],array\_adres[i]),i++

If(kontrol==1)

Yazdır(index dosyası bulunamadı)

dosyakapat(dosya\_index)

Dosyakapat(dosya\_index\_final)

Free(array\_no,array\_adres)

\_seçim 6 ise;

#### **Kayıt Güncelle:**

##### **void kayitGuncelle(ogrenci \*s1,int a,int b)**

Tanımla kontrol=0,\*dosya\_index,\*dosya\_veri

dosya\_veri(veri.bin, r+b)

Dosya\_veri pointer başa tanımla

While(fread(&s1,dosya\_veri)==1)

    if(s1->ogrNo==a && s1->dersKodu==b)

        Kontrol=1

        dosya\_veri pointerını 1 struct geri kaydır

        Fread(&s1,dosya\_veri)

        yazdır(s1->ogrNo,s1->dersKodu,s1->puan)

        Yazdır(yeni puan giriniz)

        Kullanıcıdan al(s1->puan)

        dosya\_veri pointerını 1 struct geri kaydır

        yazdır(s1->ogrNo,s1->dersKodu,s1->puan)

```

        Break;
If(!kontrol)
    Yazdır(Kayıt bulunamadı)
Dosyakapat(dosya_veri)

__seçim 7 ise;
Kayıt Sil:
void kayıtSil(ogrenci *s1, int a,int b)
Tanımla kontrol=0,*dosya_index,*dosya_veri
dosya_veri(veri.bin, r+b)
Dosya_index(index.txt,r+)
Dosya_veri pointer başa tanımla
Fileno(dosya_index),filelength(dosya_index)
fileno(dosya_veri),filelength(dosya_veri)
While(fread(&s1,dosya_veri)==1)
    if(s1->ogrNo==a && s1->dersKodu==b)
        Kontrol=1
        dosya_veri pointerını 1 struct geri kaydır
        Fread(&s1,dosya_veri)
        yazdır(s1->ogrNo,s1->dersKodu,s1->puan)
        dosya_veri pointerını 1 struct geri kaydır
        While(fscanf(dosya_index,%d,r)
            Dosya pointerı 2 int geri kaydır
            Arkadaki kayıtları dosyaya yazdır
        While(fscanf(dosya_index,%d,r)
            Dosya pointerı 2 struct geri kaydır
            Arkadaki kayıtları dosyaya yazdır
        Break;
If(!kontrol)
    Yazdır(Kayıt bulunamadı)
__chsize(fileno(veri),filelength(veri)-
sizeof(struct)

```

```

__chsize(fileno(int),filelength(int)-sizeof(int)
Dosyakapat(dosya_veri)
Dosyakapat(dosya_index)

__seçim 8 ise;
Kayıt Bul:
void kayıtBul(ogrenci *s1, int a)
Tanımla kontrol=0,n=0,*dosya_veri
Tanımla array_yindex,array_yvveri
dosya_veri(veri.bin,r+b)
While(fread(&s1,dosya_veri)
    Array_yveri=s1->ogrno,derskodu,puan
    Array_yindex=s1->ogrno
    n++;
Dosyakapat(dosya_veri)
bin_indexveri_sirala(n,yindex,yveri)
Tanımla first,last,middle,search
While(first<=last)
    if (array_yindex[middle] < search)
        first = middle + 1
    else if (array_yindex[middle] == search)
        int k=1;
        while(yindex[middle] == yindex[middle-k])
            k++;
        while(k>0)
            if(search == array_yveri[middle-k][0])
                Yazdır(array_yveri[middle-k][0])
                Yazdır(array_yveri[middle-k][1])
                Yazdır(array_yveri[middle-k][2])
            kontrol = 1;
            k--;

```

```
While(1)

if(search == array_yveri[middle+k][0])

    Yazdır(array_yveri[middle+k][0])

    Yazdır(array_yveri[middle+k][1])

    Yazdır(array_yveri[middle+k][2])

    kontrol = 1;

K++;

Else break;

Break;

Else

    Last=middle-1

    Middle=(last+first)/2

if ((first > last)&&kontrol==0)

    Yazdır(kayıt bulunamadı)

_seçim 9 ise;

Çıkış: Programdan çıkış yap
```

### **Kaynakça**

**1-)** Binary Search hakkında bilgi edinmek için:

<https://www.tutorialspoint.com/>

**2-)** Dosya işlemleri ve pointerlar hakkında bilgi edinmek için:

<https://programiz.com/>

<https://www.cs.utah.edu/>

**3-)** Karşılaşılan diğer sorunlar ve algoritma problemleri için:

<https://stackoverflow.com/>

<https://cboard.cprogramming.com/>

## Deneysel Yöntemler

### Kayıt Ekle:

```
1:kayitEkle
2:veriDosyasiniGoster
3:Yeniden indexDosyasiniOlustur
4:indexDosyasiniSil
5:indexDosyasiniGoster
6:kayitGuncelle
7:kayitSil
8:kayitBul
9:Exit
```

1

Ogrenci No: 12

Ders Kodu: 5

Puan: 5

### Veri Dosyasını Göster:

```
1:kayitEkle
2:veriDosyasiniGoster
3:Yeniden indexDosyasiniOlustur
4:indexDosyasiniSil
5:indexDosyasiniGoster
6:kayitGuncelle
7:kayitSil
8:kayitBul
9:Exit
```

2

Ogrenci No	dersKodu	puan
1	1	1
2	2	2
6	6	6
3	3	3
4	5	5
9	9	9
1	2	2
6	8	12
12	14	14
12	5	5

### Yeniden Index Dosyası Oluştur:

```
1:kayitEkle
2:veriDosyasiniGoster
3:Yeniden indexDosyasiniOlustur
4:indexDosyasiniSil
5:indexDosyasiniGoster
6:kayitGuncelle
7:kayitSil
8:kayitBul
9:Exit
```

3

Index dosyasi olusturuldu.

### İndex Dosyasını Sil:

```
1:kayitEkle
2:veriDosyasiniGoster
3:Yeniden indexDosyasiniOlustur
4:indexDosyasiniSil
5:indexDosyasiniGoster
6:kayitGuncelle
7:kayitSil
8:kayitBul
9:Exit

4
Index dosyasi basariyla silindi.
```

### İndex Dosyasını Göster:

```
1:kayitEkle
2:veriDosyasiniGoster
3:Yeniden indexDosyasiniOlustur
4:indexDosyasiniSil
5:indexDosyasiniGoster
6:kayitGuncelle
7:kayitSil
8:kayitBul
9:Exit

5

Ogrenci No      adres
1              6422036
1              6422108
2              6422048
3              6422072
4              6422084
6              6422060
6              6422120
9              6422096
12             6422132
12             6422144
```

### Kayıt Güncelle:

```
1:kayitEkle
2:veriDosyasiniGoster
3:Yeniden indexDosyasiniOlustur
4:indexDosyasiniSil
5:indexDosyasiniGoster
6:kayitGuncelle
7:kayitSil
8:kayitBul
9:Exit

6

Guncellenecek kaydın öğrenci no'sunu giriniz : 12
Guncellenecek kaydın ders kodunu giriniz : 5

Ogrenci No:12
Ders Kodu:5
Puan:5

Yeni puan giriniz: 76
```



### Kayıt Sil:

```
1:kayitEkle
2:veriDosyasiniGoster
3:Yeniden indexDosyasiniOlustur
4:indexDosyasiniSil
5:indexDosyasiniGoster
6:kayitGuncelle
7:kayitSil
8:kayitBul
9:Exit

7

Silinecek kaydın öğrenci no'sunu giriniz : 12
Silinecek kaydın ders kodunu giriniz : 5

Öğrenci No:12
Ders Kodu:5
Puan:76
```

### Kayıt Bul:

```
1:kayitEkle
2:veriDosyasiniGoster
3:Yeniden indexDosyasiniOlustur
4:indexDosyasiniSil
5:indexDosyasiniGoster
6:kayitGuncelle
7:kayitSil
8:kayitBul
9:Exit

8

Bulunacak kaydın öğrenci no'sunu giriniz : 1

Öğrenci No:1
Ders Kodu:1
Puan:1
Öğrenci No:1
Ders Kodu:2
Puan:2
```

### Exit:

```
1:kayitEkle
2:veriDosyasiniGoster
3:Yeniden indexDosyasiniOlustur
4:indexDosyasiniSil
5:indexDosyasiniGoster
6:kayitGuncelle
7:kayitSil
8:kayitBul
9:Exit

9

Process returned 0 (0x0)   execution time : 31.781 s
Press any key to continue.
```