

# DOSYALAMA FONKSİYONLARI VE YOĞUN INDEX YAPISI

## Özet

*Proje kapsamında, ikili ve metin dosyaların kullanıldığı dosyalama işlemleri ile yoğun index yapısı gerçekleştirilmiştir. Öncelikle ikili bir veri dosyasına kayıtlar eklenmiş, ardından bir metin dosyasına bu kayıtlar bubble sort algoritması ile yazılarak indexlenmiştir. Bununla birlikte çeşitli dosyalama fonksiyonları ve string kütüphaneleri bir arada kullanılmış olup, index yapısında özellikle binary search algoritmasının kullanıldığı arama işlemlerinin performansı gözlemlenmiştir.*

### 1. Giriş

İndeks dosyasındaki kayıtlar anahtara göre sıralı iken veri dosyasındaki kayıtlar sıralı değildir. Binary search yardımıyla aradığımız kayda hızlıca ulaşabilmek için veri dosyasındaki sıralı olmayan kayıtları index dosyasına sıralı şekilde eklemeyi ve çeşitli dosyalama işlemleri gerçekleştirmeyi amaçlamaktayız.

### 2. Yöntem

Projede öncelikle öğrenci bilgilerini tutması için ogrNo, dersKodu, puan değişkenlerinden oluşan bir struct yapısı tanımlanır.

Daha sonra ilgili işlemleri gerçekleştirmek üzere fonksiyonlar oluşturulur. Menü açıldığında seçilen girişe göre çeşitli dosyalama işlemleri yapılır:

- addNew: Kendisine gönderilen ogrNo, dersKodu, puan parametrelerine göre ikili veri dosyasına yeni öğrenci kaydı ekler.

- createIndex: İkili veri dosyasından okunan anahtar değeri (ogrNo) ile anahtara ait bilginin offset değeri; anahtar değeri küçükten büyüğe sıralı bir şekilde (bubble sort) metin dosyasına yazılarak indexlenir.

- findInIndex: Kendisine gönderilen ogrNo parametresini alarak binary search (ikili arama) algoritması ile ilgili öğrenci numarasına sahip öğrenciyi index dosyasında bulur ve buna bağlı olarak veri dosyasından öğrenciye ait yapılan ilk kaydı gösterir.

- deleteLine: Kendisine gönderilen ogrNo parametresini alarak veri ve index dosyasında bu kayda ait tüm kayıtları siler.

- updateLine: Kendisine gönderilen ogrNo, dersKodu, puan parametrelerini alarak daha önce girilmiş olan bir kayıt üzerinde öğrenciye ait puanı günceller.

- listData: Veri dosyasındaki tüm kayıtları listeler.

- listIndex: Index dosyasının içeriğini ekrana basar.

- destroyIndex: Çağırıldığında index dosyasını tamamen diskten siler.

- mainMenu: Yapılmak istenen işlemi seçmek için hangi tuşa basılması gerektiği bilgisini içerir.

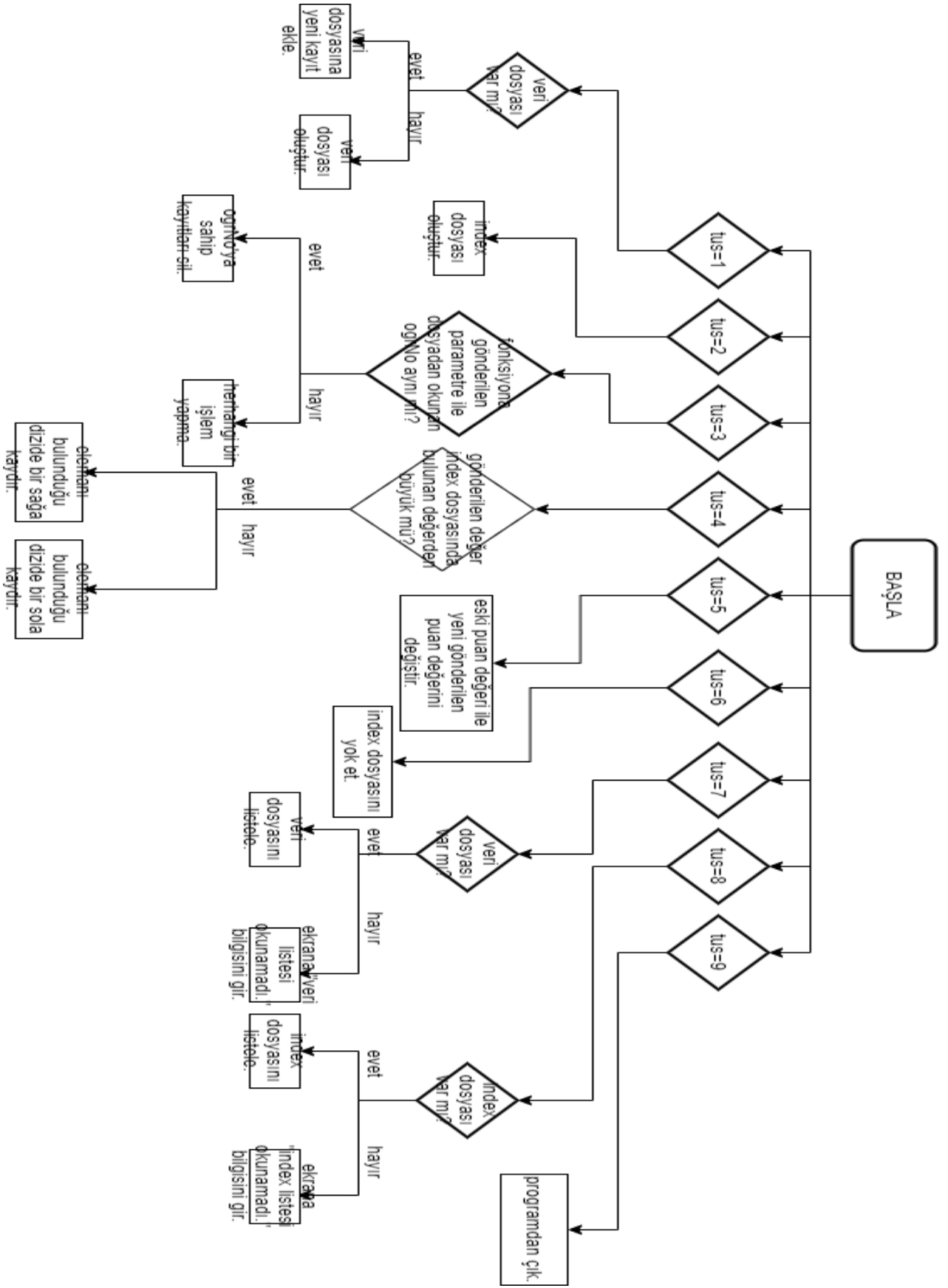
### **3. Sonuç**

Proje C programlama dili ile yazılmış olup, MacOS ve Windows İşletim sistemlerinde, Xcode, CodeBlocks ve VSCode IDE'leri ile testleri yapılmıştır.

Derleme esnasında IDE'lerin farklılığında kaynaklı basit sözdizim uyarı mesajları dışında herhangi bir compiler veya runtime hatası alınmamıştır.

İkili ve metin dosyaların kullanıldığı dosyalama işlemleri ve yoğun index yapısı gerçekleştirilmiş, index yapısında binary search algoritmasının kullanıldığı arama işlemlerinin performansı gözlemlenmiştir.

#### 4. Akış Diyagramı



## 5. Ekran Çıktılar

**DATA FILE**  
(data.bin)

ogrNo	dersKodu	puan
1008	1	10
1006	3	30
1010	2	20
1009	8	80
1001	4	40
1007	6	60
1001	5	50
1001	2	20
1004	4	40
1006	6	60
1005	2	20
1005	9	90
1008	6	60
1001	7	70
1007	3	30
1007	7	70
1009	9	90
1004	7	70
1002	8	80
1010	3	30
1010	4	40
1003	7	70
1001	1	10
1007	1	10
1009	7	70
1003	1	10
1010	5	50
1003	5	50
1004	1	10
1002	1	10
1005	1	10
1008	7	70
1009	1	10
1008	5	50
1010	1	10
1002	3	30
1003	2	20
1005	4	40
1007	8	80
1006	5	50
1004	9	90
1009	3	30
1008	9	90
1002	4	40
1002	5	50
1006	8	80
1003	9	90
1004	5	50
1005	6	60
1006	7	70

**INDEX FILE**  
(index.txt)

ogrNo = Offset
1001=48
1001=72
1001=84
1001=156
1001=264
1002=216
1002=348
1002=420
1002=516
1002=528
1003=252
1003=300
1003=324
1003=432
1003=552
1004=96
1004=204
1004=336
1004=480
1004=564
1005=120
1005=132
1005=360
1005=444
1005=576
1006=12
1006=108
1006=468
1006=540
1006=588
1007=60
1007=168
1007=180
1007=276
1007=456
1008=0
1008=144
1008=372
1008=396
1008=504
1009=36
1009=192
1009=288
1009=384
1009=492
1010=24
1010=228
1010=240
1010=312
1010=408

Index Dosyasi Yaratildi!

VERI DOSYASI ISLEMLERI	INDEX DOSYASI ISLEMLERI
Kayit Ekle [ 1 ]	Index Dosyasi Olustur [ 2 ]
Kayit Sil [ 3 ]	Kayit Bul/Binary Search [ 4 ]
Kayit Guncelle [ 5 ]	Index Dosyasini Sil [ 6 ]
Veri Dosyasini Goster [ 7 ]	Index Dosyasini Goster [ 8 ]

ISLEMINIZI SECINIZ [1]...[8], CIKIS ICIN [9]:

ISLEMINIZI SECINIZ [1]...[8], CIKIS ICIN [9]: 1

YENI KAYIT EKLE:

Ogrencinin Numarasini Giriniz: 200

Ders Kodu Bilgisini Giriniz : 201

Ders Puan Bilgisini Giriniz : 132

Index Dosyasi Yaratildi!

ISLEMINIZI SECINIZ [1]...[8], CIKIS ICIN [9]: 3

Silmek Istediginiz Ogrenci Numarasini Giriniz: 200

200 Numarali Ogrenciye Ait Tum Veriler (1 Adet) Silindi.

ISLEMINIZI SECINIZ [1]...[8], CIKIS ICIN [9]: 5

KAYIT GUNCELLE:

Ogrenci Numarasini Giriniz : 200

Ders Kodu Bilgisini Giriniz : 201

Ders Puan Bilgisini Giriniz : 100

Bulunan Eski Veri >>> Ogrenci No: 200 | Ders Kodu: 201 | Puan: 132

Guncelleme Sonrasi >>> Ogrenci No: 200 | Ders Kodu: 201 | Puan: 100

Bulunan Anahtar: 1006 = Offset: 468. Arama islemi devam ediyor...

INDEX Dosyasinda Aranan 1006 Anahtar Nolu Ogrencinin, DATA Dosyasindaki Ilk Kaydi

Ogrenci No: 1006 | Ders Kodu: 3 | Puan: 30

##### VERI DOSYASI ICERIGI #####

OGRENCI NO DERS KODU PUAN

1008	1	10
1006	3	30
1010	2	20
1009	8	80
1001	4	40
1007	6	60
1001	5	50
1001	2	20
1004	4	40
1006	6	60
1005	2	20
1005	9	90
1008	6	60
1001	7	70
1007	3	30
1007	7	70
1009	9	90
1004	7	70
1002	8	80
1010	3	30
1010	4	40
1003	7	70
1001	1	10
1007	1	10
1009	7	70

##### INDEX DOSYASI ICERIGI #####

OGRENCI NO ADRES

1001	48
1001	72
1001	84
1001	156
1001	264
1002	216
1002	348
1002	420
1002	516
1002	528
1003	252
1003	300
1003	324
1003	432
1003	552
1004	96
1004	204
1004	336
1004	480
1004	564
1005	120
1005	132
1005	360
1005	444
1005	576