



# Bölüm 6. Karma

Olcay Taner Yıldız

2014



## Karma Tablosu

[Bağlı Liste ile Karma Tablosu Tanımı](#)

[Açık Adresleme](#)

[Tekrar Karma](#)

[Uygulama: Hedef Tahtası](#)

[Uygulama: Karma Dizini](#)

# Karma Tablosu



# 5 eleman içeren örnek bir karma tablosu

[Karma Tablosu](#)

[Bağlı Liste ile Karma Tablosu Tanımı](#)

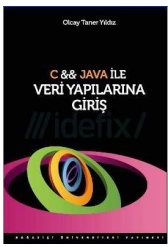
[Açık Adresleme](#)

[Tekrar Karma](#)

[Uygulama: Hedef Tahtası](#)

[Uygulama: Karma Dizini](#)

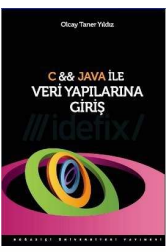
0	
1	71
2	9
3	423
4	11
5	
6	76



# Tam sayılar içeren karma tablosu tanımı

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8
	9
	10

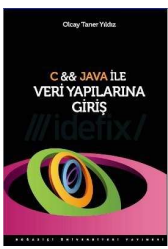
```
public class Karma{
    Ornek[] tablo;
    boolean[] silindi;
    int N;
    public Karma(int N){
        tablo = new Ornek[N];
        silindi = new boolean[N];
        this.N = N;
    }
}
```



# Tamsayılar için kullanılabilen örnek bir karma fonksiyonu

- [Karma Tablosu](#) 1
- [Bağlı Liste ile Karma Tablosu Tanımı](#) 2
- [Açık Adresleme](#) 3
- [Tekrar Karma](#)
- [Uygulama: Hedef Tahtası](#)
- [Uygulama: Karma Dizini](#)

```
int karma(int sayi){  
    return sayi % N;  
}
```



# Katarlar (karakter dizileri) için kullanılacak örnek bir karma fonksiyonu

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8

```
int karma(String katar){
    int i, pozisyon = 0;
    for (i = 0; i < katar.length; i++){
        pozisyon = 39 * pozisyon + katar.charAt(i);
    }
    pozisyon = pozisyon % N;
    return pozisyon;
}
```



[Karma Tablosu](#)

[Bağlı Liste ile Karma Tablosu Tanımı](#)

[Açık Adresleme](#)

[Tekrar Karma](#)

[Uygulama: Hedef Tahtası](#)

[Uygulama: Karma Dizini](#)

# Bağlı Liste ile Karma Tablosu Tanımı



# Her elemanı bir bağlı liste olan örnek bir karma tablosu

Karma Tablosu

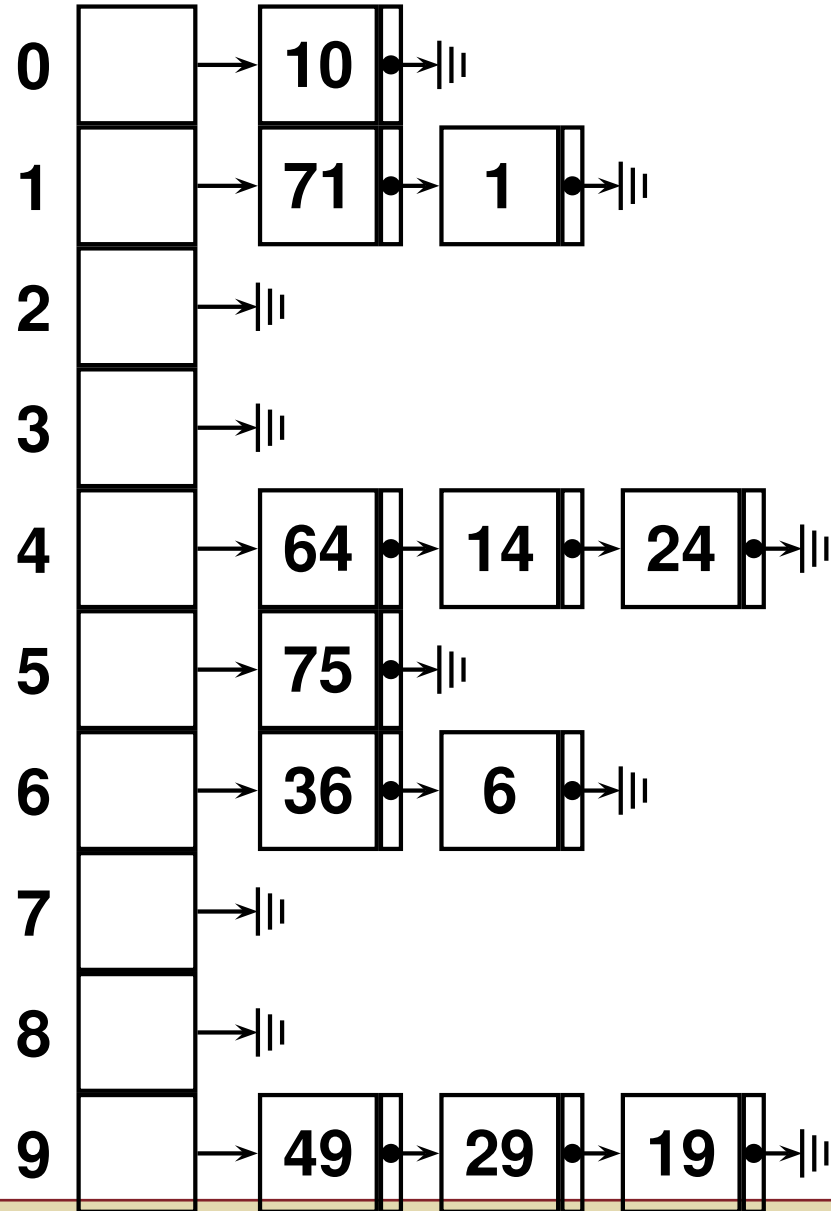
Bağlı Liste ile Karma Tablosu Tanımı

Açık Adresleme

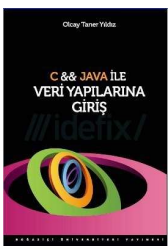
Tekrar Karma

Uygulama: Hedef Tahtası

Uygulama: Karma Dizini



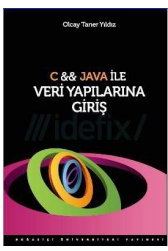




# Her elemanı bir bağlı liste olan karma tablosu tanımı

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8
	9
	10
	11

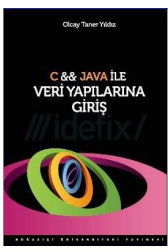
```
public class Karma{
    Liste [] tablo;
    int N;
    public Karma(int N){
        int i;
        tablo = new Liste[N];
        for (i = 0; i < N; i++)
            tablo[i] = new Liste ();
        this.N = N;
    }
}
```



# Her elemanı bağlı liste olan bir karma tablosunda bir sayı arama

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	

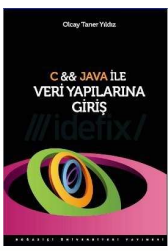
```
Eleman karmaAra(int icerik){  
    int adres;  
    adres = karma(icerik);  
    return tablo[adres].listeAra( icerik );  
}
```



# Her elemanı bağlı liste olan bir karma tablosuna yeni bir eleman ekleme

- [Karma Tablosu](#) 1
- [Bağlı Liste ile Karma Tablosu Tanımı](#) 2
- [Açık Adresleme](#) 3
- [Tekrar Karma](#) 5
- [Uygulama: Hedef Tahtası](#)
- [Uygulama: Karma Dizini](#)

```
void karmaEkle(Eleman eleman){
    int adres;
    adres = karma(eleman.icerik);
    tablo [adres]. listeyeEkle (eleman);
}
```



# Her elemanı bağlı liste olan bir karma tablosundan bir elemanı silme

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8

```
void karmaSil(int icerik){
    Eleman eleman;
    int adres;
    adres = karma(icerik);
    eleman = tablo[adres].listeAra ( icerik );
    if (eleman != null)
        tablo [adres]. listedenSil (eleman);
}
```



# *Karma Tablosu İşlemleri (Bağlı Liste)*

Karma Tablosu

Bağlı Liste ile Karma Tablosu Tanımı

Açık Adresleme

Tekrar Karma

Uygulama: Hedef Tahtası

Uygulama: Karma Dizini

- Arama:  $O(1)$
- Ekleme:  $O(1)$
- Silme:  $O(1)$



[Karma Tablosu](#)

[Bağlı Liste ile Karma Tablosu Tanımı](#)

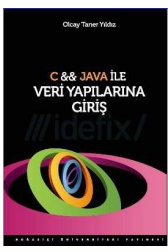
[Açık Adresleme](#)

[Tekrar Karma](#)

[Uygulama: Hedef Tahtası](#)

[Uygulama: Karma Dizini](#)

# Açık Adresleme



# Doğrusal strateji ile düzenlenmiş açık adreslemeli bir karma tablosu

[Karma Tablosu](#)

[Bağlı Liste ile Karma Tablosu Tanımı](#)

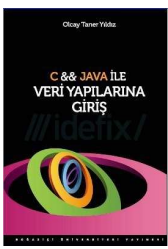
[Açık Adresleme](#)

[Tekrar Karma](#)

[Uygulama: Hedef Tahtası](#)

[Uygulama: Karma Dizini](#)

0			49	49	49
1				58	58
2					69
3					
4					
5					
6					
7					
8		18	18	18	18
9	89	89	89	89	89

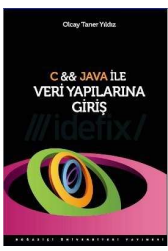


# Karma tablosunda bir sayı arama (doğrusal strateji)

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8
	9
	10

```
Ornek karmaAra(int icerik){
    int adres;
    adres = karma(icerik);
    while (tablo[adres] != null){
        if (!( silindi [adres]) && tablo[adres]. icerik == icerik )
            break;
        adres = (adres + 1) % N;
    }
    return tablo[adres];
}
```

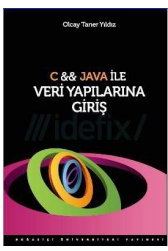




# Karma tablosunda yeni bir eleman ekleme (doğrusal strateji)

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8
	9

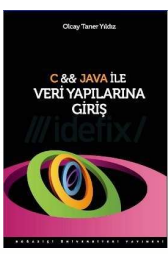
```
void karmaEkle(Ornek ornek){
    int adres;
    adres = karma(ornek.icerik);
    while (tablo[adres] != null && !(silindi [adres]))
        adres = (adres + 1) % N;
    if (tablo[adres] != null)
        silindi [adres] = false;
    tablo[adres] = ornek;
}
```



# Karma tablosundan eleman silme (doğrusal strateji)

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8
	9
	10

```
void karmaSil(int icerik){
    int adres;
    adres = karma(icerik);
    while (tablo[adres] != null){
        if (!( silindi [adres]) && tablo[adres]. icerik == icerik )
            break;
        adres = (adres + 1) % N;
    }
    silindi [adres] = true;
}
```



# İkinci derece strateji ile düzenlenmiş açık adreslemeli bir karma tablosu

Karma Tablosu

Bağlı Liste ile Karma Tablosu Tanımı

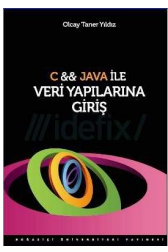
Açık Adresleme

Tekrar Karma

Uygulama: Hedef Tahtası

Uygulama: Karma Dizini

0				49	49	49
1						
2					58	58
3						69
4						
5						
6						
7						
8			18	18	18	18
9	89	89	89	89	89	89



# Çift karma strateji ile düzenlenmiş açık adreslemeli bir karma tablosu

[Karma Tablosu](#)

[Bağlı Liste ile Karma Tablosu Tanımı](#)

[Açık Adresleme](#)

[Tekrar Karma](#)

[Uygulama: Hedef Tahtası](#)

[Uygulama: Karma Dizini](#)

0					69
1					
2					
3				58	58
4					
5					
6			49	49	49
7					
8		18	18	18	18
9	89	89	89	89	89



# Karma Tablosu İşlemleri (Dizi)

Karma Tablosu

Bağlı Liste ile Karma Tablosu Tanımı

Açık Adresleme

Tekrar Karma

Uygulama: Hedef Tahtası

Uygulama: Karma Dizini

- Arama:  $O(1)$
- Ekleme:  $O(1)$
- Silme:  $O(1)$



[Karma Tablosu](#)

[Bağlı Liste ile Karma Tablosu Tanımı](#)

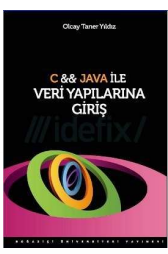
[Açık Adresleme](#)

[Tekrar Karma](#)

[Uygulama: Hedef Tahtası](#)

[Uygulama: Karma Dizini](#)

# Tekrar Karma



# Önceki şekildeki karma tablosunun tekrar karılması

[Karma Tablosu](#)

[Bağlı Liste ile Karma Tablosu Tanımı](#)

[Açık Adresleme](#)

[Tekrar Karma](#)

[Uygulama: Hedef Tahtası](#)

[Uygulama: Karma Dizini](#)

0	
1	71
2	
3	423
4	
5	
6	76
7	
8	
9	9
10	
11	11
12	
13	



# Bir karma tablosunu tekrar karma

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8
	9
	10
	11
	12
	13
	14
	15
	16
	17

```
void tekrarKarma(){
    int i;
    Ornek[] tablo;
    boolean[] silindi;
    tablo = new Ornek[N];
    silindi = new boolean[N];
    for (i = 0; i < N; i++){
        tablo[i] = this.tablo[i];
        silindi[i] = this.silindi[i];
    }
    this.tablo = new Ornek[2 * N];
    this.silindi = new boolean[2 * N];
    N = 2 * N;
    for (i = 0; i < N / 2; i++)
        if (tablo[i] != null && !silindi[i])
            karmaEkle(tablo[i]);
}
```





[Karma Tablosu](#)

[Bağlı Liste ile Karma Tablosu Tanımı](#)

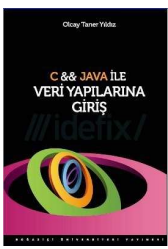
[Açık Adresleme](#)

[Tekrar Karma](#)

[Uygulama: Hedef Tahtası](#)

[Uygulama: Karma Dizini](#)

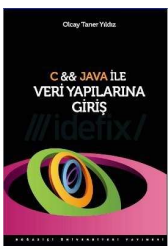
## Uygulama: Hedef Tahtası



# Hedef tahtası probleminin geniş arama yöntemiyle çözümü (1)

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8
	9
	10
	11
	12
	13

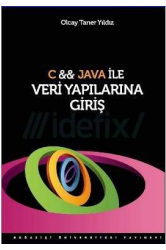
```
String hedefTahtasi(int[] tahta){
    int i, t;
    String a;
    Eleman e;
    Kuyruk k;
    Ornek o;
    Karma kt;
    e = new Eleman(0, "");
    k = new Kuyruk();
    k.kuyrugayaEkle(e);
    o = new Ornek(0);
    kt = new Karma();
    kt.karmaEkle(o);
}
```



# Hedef tahtası probleminin geniş arama yöntemiyle çözümü (2)

Karma Tablosu	14
Bağlı Liste ile Karma Tablosu Tanımı	15
Açık Adresleme	16
Tekrar Karma	17
Uygulama: Hedef Tahtası	18
Uygulama: Karma Dizini	19
	20
	21
	22
	23
	24
	25
	26
	27
	28
	29
	30
	31
	32

```
while (!k.kuyrukBos()){
    e = k.kuyrukSil ();
    if (e.toplam == 100)
        return e.atis;
    for (i = 0; i < 5; i++){
        t = e.toplam + tahta[i ];
        if (t <=100)
            if (kt.karmaAra(t) != null){
                a = e.atis ;
                a = a + "□" + tahta[i ];
                e = new Eleman(t, a);
                k.kuyrugaEkle(e);
                o = new Ornek(t);
                kt.karmaEkle(o);
            }
        }
    }
    return null;
}
```



[Karma Tablosu](#)

[Bağlı Liste ile Karma Tablosu Tanımı](#)

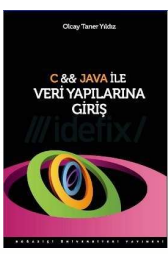
[Açık Adresleme](#)

[Tekrar Karma](#)

[Uygulama: Hedef Tahtası](#)

[Uygulama: Karma Dizini](#)

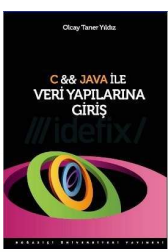
## Uygulama: Karma Dizini



# Öğrenci bilgilerini (no, ad, soyad) içeren örnek yapısı

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8
	9
	10

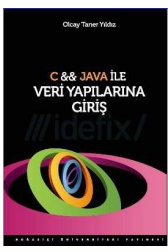
```
public class Ornek{
    String ad;
    String soyad;
    int icerik;
    public Ornek(int no, String ad, String soyad){
        this.icerik = no;
        this.ad = ad;
        this.soyad = soyad;
    }
}
```



# Öğrenci dosyasındaki bilgilerle karma tablosunu doldurmak

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6
	7
	8
	9
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19

```
Karma dosyaOku(){
    Scanner dosya;
    Ornek e;
    String ad;
    String soyad;
    int no, yas, i, sayi;
    Karma k;
    dosya = new Scanner(new File("ogrenci.txt"));
    sayi = dosya.nextInt();
    k = new Karma();
    for (i = 0; i < sayi; i++){
        no = dosya.nextInt();
        ad = dosya.next();
        soyad = dosya.next();
        e = new Ornek(no, ad, soyad);
        k.karmaEkle(e);
    }
    return k;
}
```



# Numarası 18 olan öğrencinin ad ve soyadını bulan karma dizini kullanarak fonksiyon

Karma Tablosu	1
Bağlı Liste ile Karma Tablosu Tanımı	2
Açık Adresleme	3
Tekrar Karma	4
Uygulama: Hedef Tahtası	5
Uygulama: Karma Dizini	6

```
void sorgu(Karma k){  
    Ornek o;  
    o = k.karmaAra(18);  
    System.out.print(o.ad);  
    System.out.print(o.soyad);  
}
```