



## **Algoritma Analizi - Ödev 3**

### **Hash İşlemleri**

**Öğrenci Adı: Fatih ALTINCI**

**Öğrenci No: 20011610**

**Dersin Öğretmeni: M. Elif Karşılıgil**

### Problem Tanımı:

Bu ödevde konusu Hash işlemleri olan ve bir sistemde bulunan kullanıcı isimlerini saklayan, ekleme, arama ve silme işlemleri yapılabilen bir hash tablosu hazırlanması istenmektedir.

Hash tablosu için kullanılacak hash fonksiyonlarının tanımları ödev açıklamasında verilmiştir.

Ödevin çalıştırma türü olarak 2 mod şeklinde ele alınması istenmiştir: Normal ve Debug. Normal modda yapılan işlemlerin sonuçları başarılı/başarısız şeklinde sonuç odaklı, Debug modunda ise ara adımlar hakkında da bilgiler içerecek şekilde sonuçların bastırılması istenmiştir.

Proje detayında tabloda hangi verilerin tutulacağı bilgisi verilmiştir.

Hashing mantığına uygun bir kod geliştirerek problem tanımı yerine geliştirilmiştir.

### Problemin Çözümü:

Ödevde bahsedilen şekilde veriler Struct yapısında tutularak bir Hash tablosu yaratılmıştır. Daha sonra kullanılacak fonksiyonlar yazılmıştır. Ödevin problem tanımında Arama, Ekleme, Silme, ve Düzenleme (Yeni tabloya geçme) işlemleri istenmiş ve bunlar bir menü tasarlanarak kullanıcının kullanması için gerçekleştirilmiştir.

Ana şema akışı şu şekildedir: Kullanıcıdan Load Faktör bilgisi ve eklenecek Eleman Sayısı bilgisi alınır. Daha sonra uygun bir Tablo boyutu seçilerek ve kullanıcıdan alınan elemanlar ile Hash tablosu oluşturulur. Daha sonrasında kullanıcı hangi işlemleri yapmak istediğine dair menüye yönlendirilir.

Burada ekleme şeması -> Kullanıcıdan tabloya eklenecek isim alınır -> Horner Kuralı kullanılarak bir anahtar oluşturulur -> Bu anahtar ödev tanımında belirtilen Hash fonksiyonlarından geçirilerek yerleştirilmesi gereken index belirlenir -> Index kontrolü yapılır -> (Boşsa) Yerleştirilir -> (Doluysa) Diğer Hash fonksiyonundaki mod değeri ile yeni index belirlenerek yerleştirilir.

Silme şeması -> Kullanıcıdan tablodan silinecek isim alınır -> Ekleme aşamasındaki işlemler uygulanarak index bulunur -> O index'teki kullanıcı Struct'ının Deleted parametresi 1 yapılır. Böylelikle fiziksel silinme yapılmadığı için listelenme aşamasında 0 listelenirse o eleman görülmez (güncel veritabanları bu şekildedir).

Arama şeması -> Kullanıcıdan aranacak isim alınır -> Aynı aşamalar gerçekleştirilir ve index bulunur -> Bu kontroller sırasında ekstra Deleted parametresi de kontrol edilir -> Kullanıcı silinmişse veya yoksa bulunmaz -> Varsa bulunduğu ibaresi bastırılır.

Yeni tabloya geçme şeması -> Kullanıcı tabloda çok fazla ekleme, silme vb. işlemleri yapmış olabilir. Bu durumda parametreleri sıfırlayarak sadece silinmemiş kullanıcıların Deleted = 0 olanların içerdiği isimleri yeni bir Hash tablosu oluşturarak oraya aktarır -> Daha sonrasında eski tabloyu bellekten boşaltır.

Çıkış -> Programı sonlandırır.

### Karşılaşılan Sorunlar:

Ne kadar istemesek de çakışma problemleri ile karşılaşabiliyoruz. Farklı keyler sonucunda aynı index'e yerleşmesi gereken isimler yer alabiliyor. Örneğin: Load Factor: 0.5, Eleman Sayısı 6 olacak şekilde bir tablo oluşturalım.

Daha önce sıralı işlemlerimiz sonucu eklemiş olduğumuz "Eren, Fatih, Talha, Ilker, Esat, Ugur" şeklinde isimlerimiz olsun. Bunlar yerleşmesi gereken index'lere yerleştirdikten sonra kullanıcı "BC"

değerini eklemek istesin bunun yerleşmesi gereken index'de 0'dır. O Index dolu olduğu için probing yaparak başka bir index'e yerleşir.

Parametreleri ben int şeklinde kodladım. Eğer key'lerin değer max int değeri geçerse negatif değer döndürüp yanlış index'e yerleşebilir. Long Int vb. tanımlar yapılarak çözülebilir.

Daha önce silinmiş olan bir kullanıcının olduğu index'e kullanıcı yerleşmeye çalışabilir bu durumda Deleted parametresi de kontrol edilmeli ve probing işlemine devam edilerek uygun index'e yerleştirilmeli.

### **Karmaşıklık Analizi:**

int calculateKey(char\* word, int R, int modSecimi):

- En iyi:  $O(1)$
- En kötü:  $O(N)$
- Ortalama:  $O(N)$

int h1(int key, int m, int modSecimi) –  $O(1)$

int h2(int key, int m, int modSecimi) –  $O(1)$

int h(int key, int i, int m, int modSecimi) –  $O(1)$

void insert(HASHTABLE \*hashTable, char \*userName, int m, int modSecimi):

- En iyi:  $O(1)$
- En kötü:  $O(N)$
- Ortalama:  $O(1)$

int search(HASHTABLE \*hashTable, char \*userName, int m, int modSecimi):

- En iyi:  $O(1)$
- En kötü:  $O(N)$
- Ortalama:  $O(N)$

void removeUser(HASHTABLE \*hashTable, char \*userName, int m, int modSecimi):

- En iyi:  $O(1)$
- En kötü:  $O(N)$
- Ortalama:  $O(N)$

HASHTABLE\* editTable(HASHTABLE \*hashTable, int m, int modSecimi):

- En iyi:  $O(N)$
- En kötü:  $O(N^2)$
- Ortalama:  $O(N)$

void printTable(HASHTABLE \*hashTable, int m) –  $O(N)$

HASHTABLE\* resizeTable(HASHTABLE \*hashTable, int \*M, int modSecimi):

- En iyi:  $O(N)$
- En kötü:  $O(N^2)$
- Ortalama:  $O(N)$

int isPrime(int num):

- En iyi:  $O(1)$
- En kötü:  $O(\sqrt{N})$
- Ortalama:  $O(\sqrt{N})$

int findNextPrime(int num):

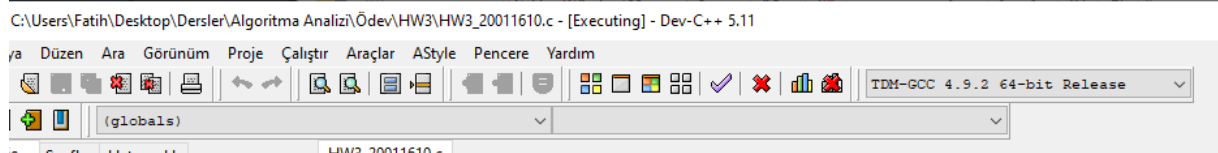
- En iyi:  $O(1)$
- En kötü:  $O(\sqrt{N})$
- Ortalama:  $O(\sqrt{N})$

**Toplam Karmaşıklık:**

- En iyi:  $O(1)$
- En kötü:  $O(N)$
- Ortalama:  $O(1)$ , nadiren  $O(N)$  olur.

## Ekran Çıktıları:

Dev-C++ - TDM-GCC 4.9.2 64 bitte derlenmiş ve kontrol edilmiştir, sıkıntı yoktur.



N=6, Load Faktör = 0.5, Mod: Normal

```
Mod secimi yapiniz (1: NORMAL, 2: DEBUG): 1
Eklenebilecek eleman sayisi (N) giriniz: 6
Load factor (lambda) giriniz: 0.5
Hash tablosu uzunlugu (M): 13
1. kullanıcı adini giriniz: Ugur
Ugur kullanıcı adi hash tablosuna eklendi.
2. kullanıcı adini giriniz: Eren
Eren kullanıcı adi hash tablosuna eklendi.
3. kullanıcı adini giriniz: Esat
Esat kullanıcı adi hash tablosuna eklendi.
4. kullanıcı adini giriniz: Fatih
Fatih kullanıcı adi hash tablosuna eklendi.
5. kullanıcı adini giriniz: Ilker
Ilker kullanıcı adi hash tablosuna eklendi.
6. kullanıcı adini giriniz: Talha
Talha kullanıcı adi hash tablosuna eklendi.

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdir
5. Yeni Tabloya Gec
6. Cikis
Seciminizi yapiniz: _
```

Tabloyu yazdırma:

```
Seciminizi yapiniz: 4

Hash tablosu:
0. index: Eren (Silinme Durumu: 0)
1. index: NULL
2. index: Ugur (Silinme Durumu: 0)
3. index: NULL
4. index: NULL
5. index: Talha (Silinme Durumu: 0)
6. index: NULL
7. index: Ilker (Silinme Durumu: 0)
8. index: Fatih (Silinme Durumu: 0)
9. index: NULL
10. index: NULL
11. index: Esat (Silinme Durumu: 0)
12. index: NULL
```

Kullanıcı ekleme:

```
Seciminizi yapiniz: 1
Eklemek istediginiz kullanıcı adini giriniz: Mehmet
Mehmet kullanıcı adi hash tablosuna eklendi.

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdir
5. Yeni Tabloya Gec
6. Cikis
Seciminizi yapiniz: 4

Hash tablosu:
0. index: Eren (Silinme Durumu: 0)
1. index: NULL
2. index: Ugur (Silinme Durumu: 0)
3. index: Mehmet (Silinme Durumu: 0)
4. index: NULL
5. index: Talha (Silinme Durumu: 0)
6. index: NULL
7. index: Ilker (Silinme Durumu: 0)
8. index: Fatih (Silinme Durumu: 0)
9. index: NULL
10. index: NULL
11. index: Esat (Silinme Durumu: 0)
12. index: NULL
```

Kullanıcı silme:

```
Seciminizi yapiniz: 3
Silmek istediginiz kullanıcı adini giriniz: Eren
Eren kullanıcı adi silindi.

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdir
5. Yeni Tabloya Gec
6. Cikis
Seciminizi yapiniz: 4

Hash tablosu:
0. index: Eren (Silinme Durumu: 1)
1. index: NULL
2. index: Ugur (Silinme Durumu: 0)
3. index: Mehmet (Silinme Durumu: 0)
4. index: NULL
5. index: Talha (Silinme Durumu: 0)
6. index: NULL
7. index: Ilker (Silinme Durumu: 0)
8. index: Fatih (Silinme Durumu: 0)
9. index: NULL
10. index: NULL
11. index: Esat (Silinme Durumu: 0)
12. index: NULL
```

Kullanıcı arama (Hem olan hem silineni arayalım):

```
Seciminizi yapiniz: 2
Aramak istediginiz kullanıcı adini giriniz: Eren
Eren kullanıcı adi hash tablosunda bulunamadi.

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdir
5. Yeni Tabloya Gec
6. Cikis
Seciminizi yapiniz: 2
Aramak istediginiz kullanıcı adini giriniz: Fatih
Fatih kullanıcı adi 8. adreste bulunmaktadır.
```

Şimdi halihazırda Eren'in indexine eşit 'BC' eklemesi yapalım. (0. index)

```
Seciminizi yapiniz: 1
Eklemek istediginiz kullanıcı adini giriniz: BC
BC kullanıcı adi hash tablosuna eklendi.

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdir
5. Yeni Tabloya Gec
6. Cikis
Seciminizi yapiniz: 4

Hash tablosu:
0. index: Eren (Silinme Durumu: 1)
1. index: NULL
2. index: Ugur (Silinme Durumu: 0)
3. index: Mehmet (Silinme Durumu: 0)
4. index: NULL
5. index: Talha (Silinme Durumu: 0)
6. index: NULL
7. index: Ilker (Silinme Durumu: 0)
8. index: Fatih (Silinme Durumu: 0)
9. index: BC (Silinme Durumu: 0)
10. index: NULL
11. index: Esat (Silinme Durumu: 0)
12. index: NULL
```

Görüldüğü gibi uygun olan diğer index'e yerleşti. Eren silinmiş olmasına rağmen kontrol edildi ve oraya yerleşmedi.

Şimdi Ugur'u da silip yeni bir tabloya geçelim.

```
Seciminizi yapiniz: 3
Silmek istediginiz kullanıcı adini giriniz: Ugur
Ugur kullanıcı adi silindi.

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdir
5. Yeni Tabloya Gec
6. Cikis
Seciminizi yapiniz: 4

Hash tablosu:
0. index: Eren (Silinme Durumu: 1)
1. index: NULL
2. index: Ugur (Silinme Durumu: 1)
3. index: Mehmet (Silinme Durumu: 0)
4. index: NULL
5. index: Talha (Silinme Durumu: 0)
6. index: NULL
7. index: Ilker (Silinme Durumu: 0)
8. index: Fatih (Silinme Durumu: 0)
9. index: BC (Silinme Durumu: 0)
10. index: NULL
11. index: Esat (Silinme Durumu: 0)
12. index: NULL

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdir
5. Yeni Tabloya Gec
6. Cikis
Seciminizi yapiniz: 5
Yeni tabloya geciliyor...

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdir
5. Yeni Tabloya Gec
6. Cikis
Seciminizi yapiniz: 4

Hash tablosu:
0. index: BC (Silinme Durumu: 0)
1. index: NULL
2. index: NULL
3. index: Mehmet (Silinme Durumu: 0)
4. index: NULL
5. index: Talha (Silinme Durumu: 0)
6. index: NULL
7. index: Ilker (Silinme Durumu: 0)
8. index: Fatih (Silinme Durumu: 0)
9. index: NULL
10. index: NULL
11. index: Esat (Silinme Durumu: 0)
12. index: NULL
```

Görüldüğü gibi önce Ugur silindi ve ardından yeni tabloya geçilerek sadece silinmeyen kullanıcılar yeni tabloya taşındı ve uygun indexlere yerleştirildiler.



Şimdi bir de Debug modda çalıştıralım:

N=3, Load Faktör = 0.5, Mod: Debug

```
Mod secimi yapınız (1: NORMAL, 2: DEBUG): 2
Eklenebilecek eleman sayısı (N) giriniz: 3
Load factor (lambda) giriniz: 0.5
Hash tablosu uzunlugu (M): 7
1. kullanıcı adini giriniz: Ugur
----- Ugur KULLANICISI EKLENİYOR -----
Kullanici adi: Ugur
word[0] = U, result = 21
word[1] = g, result = 690
word[2] = u, result = 21443
word[3] = r, result = 664783
Key: 664783
h1(664783) = 0
h2(664783) = 4
h2(664783) = 4
h1(664783) = 0
h(664783, 0) = (0 + 0 * 4) % 7 = 0
h1(664783) = 0
h2(664783) = 4
Index: 0
Ugur kullanıcı adi hash tablosuna eklendi.
2. kullanıcı adini giriniz: Eren
----- Eren KULLANICISI EKLENİYOR -----
Kullanici adi: Eren
word[0] = E, result = 5
word[1] = r, result = 205
word[2] = e, result = 6392
word[3] = n, result = 198198
Key: 198198
h1(198198) = 0
h2(198198) = 4
h2(198198) = 4
h1(198198) = 0
h(198198, 0) = (0 + 0 * 4) % 7 = 0
h1(198198) = 0
h2(198198) = 4
Index: 0
h1(198198) = 0
h2(198198) = 4
h2(198198) = 4
h1(198198) = 0
h(198198, 1) = (0 + 1 * 4) % 7 = 4
h1(198198) = 0
h2(198198) = 4
Index: 4
Eren kullanıcı adi hash tablosuna eklendi.
3. kullanıcı adini giriniz: Fatih
----- Fatih KULLANICISI EKLENİYOR -----
Kullanici adi: Fatih
word[0] = F, result = 6
word[1] = a, result = 219
word[2] = t, result = 6841
word[3] = i, result = 212112
word[4] = h, result = 6575512
Key: 6575512
h1(6575512) = 6
h2(6575512) = 3
h2(6575512) = 3
h1(6575512) = 6
h(6575512, 0) = (6 + 0 * 3) % 7 = 6
h1(6575512) = 6
h2(6575512) = 3
Index: 6
Fatih kullanıcı adi hash tablosuna eklendi.
```

Tabloyu yazdırma:

```
Seciminizi yapiniz: 4

Hash tablosu:
0. index: Ugur (Silinme Durumu: 0)
1. index: NULL
2. index: NULL
3. index: NULL
4. index: Eren (Silinme Durumu: 0)
5. index: NULL
6. index: Fatih (Silinme Durumu: 0)
```

Kullanıcı ekleme:

```
----- Ilker KULLANICISI EKLENİYOR -----
Kullanici adi: Ilker
word[0] = I, result = 9
word[1] = l, result = 323
word[2] = k, result = 10056
word[3] = e, result = 311773
word[4] = r, result = 9665013
Key: 9665013
h1(9665013) = 1
h2(9665013) = 4
h2(9665013) = 4
h1(9665013) = 1
h(9665013, 0) = (1 + 0 * 4) % 7 = 1
h1(9665013) = 1
h2(9665013) = 4
Index: 1
Ilker kullanıcı adi hash tablosuna eklendi.

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdır
5. Yeni Tabloya Geç
6. Çıkış
Seciminizi yapiniz: 4

Hash tablosu:
0. index: Ugur (Silinme Durumu: 0)
1. index: Ilker (Silinme Durumu: 0)
2. index: NULL
3. index: NULL
4. index: Eren (Silinme Durumu: 0)
5. index: NULL
6. index: Fatih (Silinme Durumu: 0)
```

Kullanıcı silme:

```
Silmek istediginiz kullanıcı adını giriniz: Ilker
word[0] = I, result = 9
word[1] = l, result = 323
word[2] = k, result = 10056
word[3] = e, result = 311773
word[4] = r, result = 9665013
h1(9665013) = 1
h2(9665013) = 4
h2(9665013) = 4
h1(9665013) = 1
h(9665013, 0) = (1 + 0 * 4) % 7 = 1
h1(9665013) = 1
h2(9665013) = 4
Adım 1 - Index 1: Ilker
Ilker kullanıcı adı silindi.

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdır
5. Yeni Tabloya Geç
6. Çıkış
Seçiminizi yapınız: 4

Hash tablosu:
0. index: Ugur (Silinme Durumu: 0)
1. index: Ilker (Silinme Durumu: 1)
2. index: NULL
3. index: NULL
4. index: Eren (Silinme Durumu: 0)
5. index: NULL
6. index: Fatih (Silinme Durumu: 0)
```

Kullanıcı arama:

```
Seciminizi yapiniz: 2
Aramak istediginiz kullanıcı adini giriniz: Ilker
word[0] = I, result = 9
word[1] = l, result = 323
word[2] = k, result = 10056
word[3] = e, result = 311773
word[4] = r, result = 9665013
h1(9665013) = 1
h2(9665013) = 4
h2(9665013) = 4
h1(9665013) = 1
h(9665013, 0) = (1 + 0 * 4) % 7 = 1
h1(9665013) = 1
h2(9665013) = 4
Adim 1 - Index 1: Ilker
h1(9665013) = 1
h2(9665013) = 4
h2(9665013) = 4
h1(9665013) = 1
h(9665013, 1) = (1 + 1 * 4) % 7 = 5
h1(9665013) = 1
h2(9665013) = 4
Ilker kullanıcı adi hash tablosunda bulunamadi.
```

----- MENU -----

1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdir
5. Yeni Tabloya Gec
6. Cikis

```
Seciminizi yapiniz: 2
Aramak istediginiz kullanıcı adini giriniz: Fatih
word[0] = F, result = 6
word[1] = a, result = 219
word[2] = t, result = 6841
word[3] = i, result = 212112
word[4] = h, result = 6575512
h1(6575512) = 6
h2(6575512) = 3
h2(6575512) = 3
h1(6575512) = 6
h(6575512, 0) = (6 + 0 * 3) % 7 = 6
h1(6575512) = 6
h2(6575512) = 3
Adim 1 - Index 6: Fatih
Fatih kullanıcı adi 6. adreste bulunmaktadır.
```

MENU

Bir de yeni tabloya geçelim.

```
Yeni tabloya geciliyor...
----- Ugur KULLANICISI EKLENİYOR -----
Kullanici adi: Ugur
word[0] = U, result = 21
word[1] = g, result = 690
word[2] = u, result = 21443
word[3] = r, result = 664783
Key: 664783
h1(664783) = 0
h2(664783) = 4
h2(664783) = 4
h1(664783) = 0
h(664783, 0) = (0 + 0 * 4) % 7 = 0
h1(664783) = 0
h2(664783) = 4
Index: 0
----- Eren KULLANICISI EKLENİYOR -----
Kullanici adi: Eren
word[0] = E, result = 5
word[1] = r, result = 205
word[2] = e, result = 6392
word[3] = n, result = 198198
Key: 198198
h1(198198) = 0
h2(198198) = 4
h2(198198) = 4
h1(198198) = 0
h(198198, 0) = (0 + 0 * 4) % 7 = 0
h1(198198) = 0
h2(198198) = 4
Index: 0
h1(198198) = 0
h2(198198) = 4
h2(198198) = 4
h1(198198) = 0
h(198198, 1) = (0 + 1 * 4) % 7 = 4
h1(198198) = 0
h2(198198) = 4
Index: 4
----- Fatih KULLANICISI EKLENİYOR -----
Kullanici adi: Fatih
word[0] = F, result = 6
word[1] = a, result = 219
word[2] = t, result = 6841
word[3] = i, result = 212112
word[4] = h, result = 6575512
Key: 6575512
h1(6575512) = 6
h2(6575512) = 3
h2(6575512) = 3
h1(6575512) = 6
h(6575512, 0) = (6 + 0 * 3) % 7 = 6
h1(6575512) = 6
h2(6575512) = 3
Index: 6

----- MENU -----
1. Ekle
2. Ara
3. Sil
4. Tabloyu Yazdir
5. Yeni Tabloya Gec
6. Cikis
Seciminizi yapiniz: 4

Hash tablosu:
0. index: Ugur (Silinme Durumu: 0)
1. index: NULL
2. index: NULL
3. index: NULL
4. index: Eren (Silinme Durumu: 0)
5. index: NULL
6. index: Fatih (Silinme Durumu: 0)
```