

Hashing (Kırpma)

- Hash Fonksiyonları
 - Selecting Digits
 - Folding (shift folding, boundary folding)
 - Division
 - Mid-Square
 - Extraction
 - Radix Transformation
- Çakışma (Collision) ve çözümler
 - Linear Probing
 - Double Hashing
 - Quadratic Probing
 - Chaining

Yrd.Doç.Dr. M. Ali Akcayol

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

- Arama metodlarında temel işlem anahtarları karşılaştırmaktır.
- Bir anahtarın tablo içerisinde bulunduğu pozisyona ulaşıncaya kadar arama işlemine devam edilir.
- Hash fonksiyonuyla aranan anahtar elemana doğrudan erişilebilmektedir.
- Hash fonksiyonu bir anahtar bilgisinin tabloda bulunduğu indeksi hesaplamaktadır.
- Open hashing (Açık kırpma): Potansiyel olarak limitsiz alan kullanır.
- Closed hashing (Kapalı kırpma): Bilgi kaydı için sabit alan kullanır.

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

- Boyutu N olan bir tabloda, hash fonksiyonu ($h(x)$) bir x anahtarını 0 ile N-1 arasında bir deęerle eşleřtirir.

Örnek:

N=15 olan bir tablo için $h(x) = x \% 15$ (% -> modlu bölüm) olarak belirlenebilir.

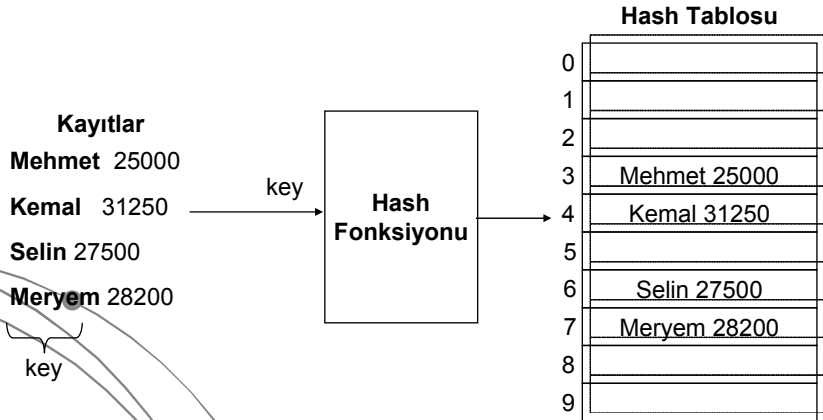
Eđer $x =$ 25 129 35 2501 47 36
 $h(x) =$ 10 9 5 11 2 6

Anahtarların tablo içerisindeki yerleri ise ařağıdaki gibidir:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
-	-	47	-	-	35	36	-	-	129	25	2501	-	-	-

G. Ü. Bilgisayar Mühendislięi Bölümü

Hashing

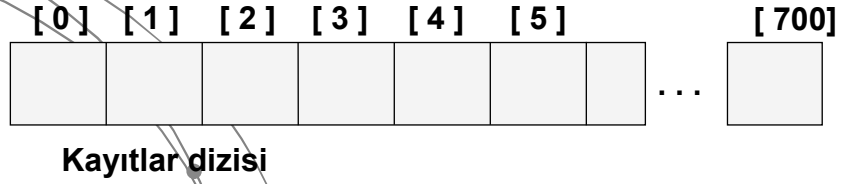


G. Ü. Bilgisayar Mühendislięi Bölümü

Hashing

Örnek:

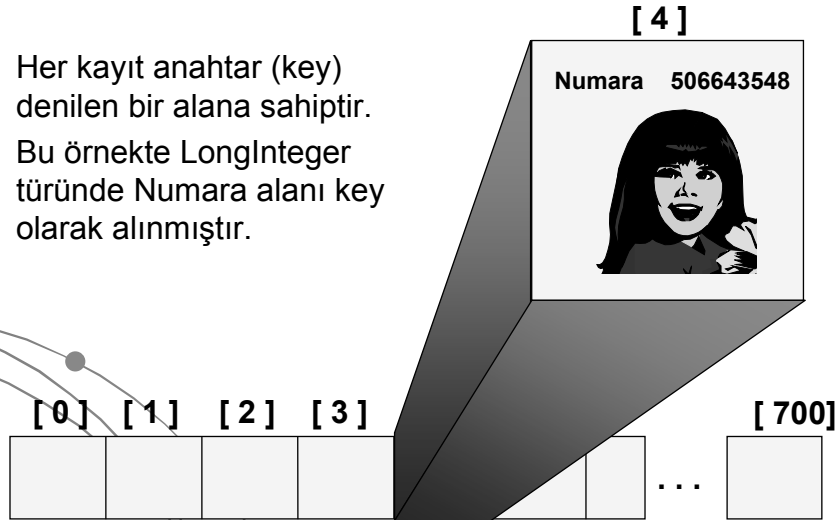
Aşağıdaki tablo 701 kayıt içermektedir.



G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

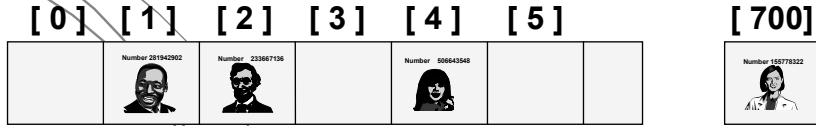
- Her kayıt anahtar (key) denilen bir alana sahiptir.
- Bu örnekte LongInteger türünde Numara alanı key olarak alınmıştır.



G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

- Hash tablosu kullanıldığında bazı pozisyonlar geçerli kayıtlara sahiptir, bazıları ise boştur.



G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Hash fonksiyonları
Bir anahtarın tabloda bulunduğu indeks sırasını verir.

Perfect hash fonksiyonu:
Herbir anahtara sadece bir pozisyonu eşleştiren fonksiyona denir.

Simple perfect hash fonksiyonu:
Tablo boyutu ile toplam anahtar sayısı eşit olduğunda (tabloda boş yer yoksa) herbir anahtara sadece bir pozisyonu eşleştiren fonksiyona denir.

İyi bir hash fonksiyonu:

- kolay ve hızlı hesaplanabilir olmalıdır.
- tablodaki herbir pozisyon için sadece bir anahtar atmalıdır.

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Hash fonksiyonları

- Hash fonksiyonları integer sayılarla işlem yaparlar.
- Integer olmayan anahtarlarda integer değere dönüştürme işlemi yapılır.
- Örneğin kişilere ait sağlık numarası 9635-8904 şeklinde ise aradaki tire işareti kaldırılarak 96358904 olarak alınır.
- Eğer anahtar karakterlerden oluşuyorsa karakterlerin ASCII kodları kullanılır.

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Hash fonksiyonları (Selecting Digits – Rakam Seçme)

- Anahtar üzerindeki belirlenmiş bazı haneleri seçip birleştirerek tablodaki pozisyon bulunur.
- 2. ve 5. hanelerin seçimiyle oluşturulan değer aşağıdaki gibidir.
 $h(033475678) = 37$
 $h(023455678) = 25$

Artıları ve Eksileri (Pros and Cons)

- Yapısı basittir.
- Anahtarları tablonun tamamına düzgün bir şekilde dağıtamaz.
- Çakışma çok sık olabilir.

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Hash fonksiyonları (Folding)

- Anahtar birkaç parçaya bölünür ve bu parçalar kendi arasında toplanarak tablodaki pozisyon bulunur.
- Shift folding metodunda anahtarın herbir parçası değiştirilmeden tablo boyutuna göre mod ile toplanır.

Örnek: SSN = 123-45-6789 olarak verilsin. SSN numarası 123, 456, 789 olarak üç parçaya ayrılıp toplandığında $123+456+789 = 1368$ olarak pozisyon numarası elde edilir.

- Boundary folding metodunda anahtarın parçalarının sırası değiştirilerek tablo boyutuna göre mod ile toplanır.

Örnek: SSN numarası 123, 456, 789 olarak üç parçaya ayrılır. Birinci parça aynı sırada kalır ve ikinci parça ters sıralanır. Daha sonra üçüncü parça aynı sırada alınır ve tablo boyutuna göre mod ile toplanır. ($123+654+789 = 1566$)

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Hash fonksiyonları (Division - Bölme)

- Anahtar değeri tablo boyutuna göre mod ile bölünür.

Örnek: SSN = 123456789 olarak verilsin. Tablo boyutu 1000 olursa, hash fonksyonu sonucu aşağıdaki gibi bulur;

$$\text{hash}(h) = 123456789 \% 1000 = 789$$

- Yapısı basittir ancak çakışma olur.

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Hash fonksiyonları (Mid-square, Orta kare)

- Anahtar değerin karesi alınır ve sonucun orta kısmı seçilerek tablodaki pozisyon değeri bulunur.

Örnek: anahtar = 3121 olarak verilsin. Hash fonksyonu sonucu aşağıdaki gibi bulur;

$$3121^2 = 9740641$$
$$\text{hash}(3121) = 406$$

- Anahtarın karesi binary olarak gösterilebilir.

$$3121^2 = 100101001010000101100001$$
$$\text{hash}(3121) = 0101000010 = 322$$

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Hash fonksiyonları (Extraction)

- Anahtar değerinin sadece bazı kısımları seçilerek tablodaki pozisyon değeri bulunur.

Örnek: anahtar = 123-45-6789 olarak verilsin. Hash fonksyonu aşağıdakilerden herhangi birisi olabilir;

- $\text{hash}(123-45-6789) = 123456789 = 1234$

- $\text{hash}(123-45-6789) = 123456789 = 6789$

- $\text{hash}(123-45-6789) = 123456789 = 1289$

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Hash fonksiyonları (Radix Transformation)

- Anahtar değeri başka bir sayı tabanına dönüştürülür.

Örnek: anahtar = 1238 olarak verilsin. Tablo boyutu 1000 olarak alındığında,

$$1238_{10} = 2326_8$$

Hesaplanan değer tablo boyutuna mod ile bölünerek pozisyon değeri bulunur.

$$\text{Hash}(1238) = 2326 \% 1000 = 326$$

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışma

$x = 65$ değerini aşağıdaki tabloya ekleyim.

$$\begin{aligned}x &= 65 \\ h(x) &= 5\end{aligned}$$

Aynı pozisyona birden fazla kayıt gelirse çakışma meydana gelir.

0	-
1	-
2	47
3	-
4	-
5	35
6	36
7	-
8	-
9	129
10	25
11	2501
12	-
13	-
14	-

65

G. Ü. Bilgisayar Mühendisliği Bölümü

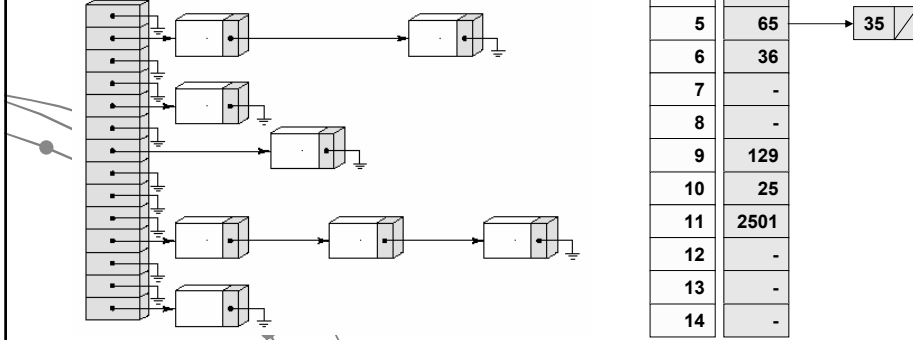
Hashing

Çakışmanın giderilmesi (Chaining)

Aynı pozisyona gelen kayıtlar bağlı listelerle gösterilir.

Ekleme: Listenin başına eklenir

Silme/Erişim: Uygun listede arama yapılır



G. Ü. Bilgisayar Mühendisliği Bölümü

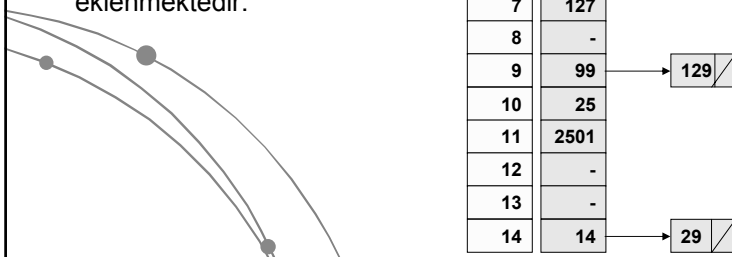
Hashing

Çakışmanın giderilmesi (Chaining)

Örnek:

29, 16, 14, 99, 127 sayılarının eklenmesi

Aynı pozisyona gelen diğer anahtarlar bağlı listenin başına eklenmektedir.

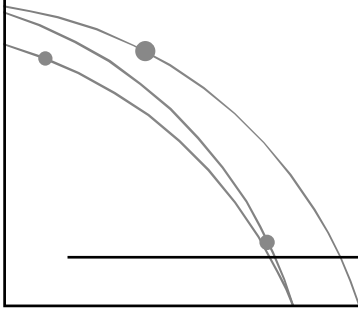


G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Chaining metodunun dezavantajları

- Tablonun bazı kısımları hiç kullanılmamaktadır.
- Bağlı listeler uzadıkça arama ve silme işlemleri için gereken zaman uzamaktadır.



G. Ü. Bilgisayar Mühendisliği Bölümü

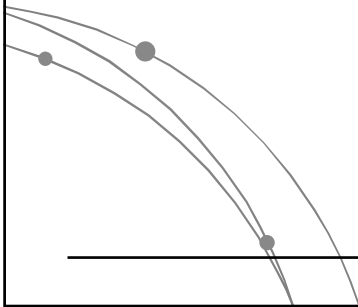
Hashing

Çakışmanın giderilmesi (Linear Probing)

Aynı pozisyona gelen ikinci kayıt ilgili pozisyondan sonraki ilk boş pozisyona yerleştirilir.

Ekleme: Boş bir alan bulunarak yapılır.

Silme/Erişim: İlk boş alan bulunana kadar devam edebilir.



0	-
1	-
2	47
3	-
4	-
5	35
6	36
7	65
8	-
9	129
10	25
11	2501
12	-
13	-
14	-

65

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Linear Probing)

Örnek :

Numara 701466868



Hash değeri 2

[0] [1] [2] [3] [4] [5] [700]

	Number 281942902	Number 233687136	Number 606026865	Number 506643548		

Number 15279322

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Linear Probing)

Örnek : (devam)

Numara 701466868



Hash değeri 2

Çakışma olunca ilk boşluğa kadar hareket edilir.

[0] [1] [2] [3] [4] [5] [700]

	Number 281942902	Number 233687136	Number 606026865	Number 506643548		

Number 15279322

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Linear Probing)

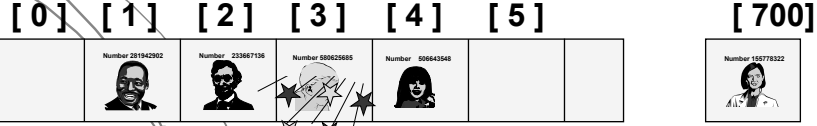
Örnek : (devam)

Numara 701466868



Çakışma olunca ilk boşluğa kadar hareket edilir.

Hash değeri 2



G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Linear Probing)

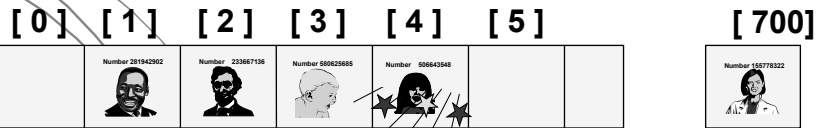
Örnek : (devam)

Numara 701466868



Çakışma olunca ilk boşluğa kadar hareket edilir.

Hash değeri 2



G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Linear Probing)

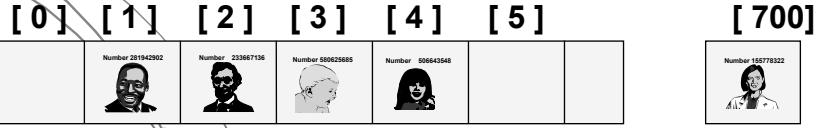
Örnek : (devam)

Çakışma olunca ilk boşluğa kadar hareket edilir.

Numara 701466868



Hash değeri 2



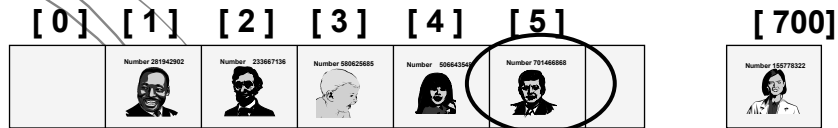
G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Linear Probing)

Örnek : (devam)

Yeni kayıt ilk boşluğa yerleştirilir.



G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Linear Probing)

- Örnek :
- $h(x) = x \text{ mod } 13$
 - 18, 41, 22, 44, 59, 32, 31, 73 değerlerini verilen sırada giriniz.

0	1	2	3	4	5	6	7	8	9	10	11	12



		41			18	44	59	32	22	31	73	
0	1	2	3	4	5	6	7	8	9	10	11	12

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Linear Probing metodunun avantajları / dezavantajları

- Bağlı listeler gibi ayrı bir veri yapısına ihtiyaç duyulmaz.
- Kayıtların yığın şeklinde toplanmasına sebep olur.



- Silme ve arama işlemleri için gereken zaman aynı hash değeri sayısı arttıkça artar.

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Quadratic Probing)

Aynı pozisyona gelen ikinci kayıt quadratic Fonksiyonla yerleştirilir.

En çok kullanılan fonksiyon

$$t = h(t)$$
$$f(x) = t + x^2$$

Yeni pozisyon için sırasıyla $(t+1^2)$, $(t+2^2)$, ..., $(t+n^2)$ değerlerine karşılık gelen pozisyonlara bakılır ve ilk boş olana yerleştirilir.

0	-	
1	-	
2	47	
3	-	
4	-	
5	35	X
6	36	X
7	-	
8	-	
9	129	X
10	25	
11	2501	
12	-	
13	-	
14	65	X

$t = h(65) = 5$

$t+1^2$

$t+2^2$

$t+3^2$

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Quadratic Probing)

Örnek: 29, 16, 14, 99, 127 değerlerini hash tablosuna quadratic probing metoduyla sırayla yerleştiriniz.

$$h(x) = x \bmod 15$$

0	29	X
1	-	
2	47	
3	-	
4	-	
5	35	
6	36	
7	-	
8	-	
9	129	
10	25	
11	2501	
12	-	
13	-	
14	65	X

$t+1^2$

$t = h(29) = 14$

t

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Quadratic Probing)

Örnek: 29, 16, 14, 99, 127 değerlerini hash tablosuna quadratic probing metoduyla sırayla yerleştiriniz.

$$h(x) = x \bmod 15$$

0	29	$t = h(16) = 1$
1	16	t
2	47	
3	-	
4	-	
5	35	
6	36	
7	-	
8	-	
9	129	
10	25	
11	2501	
12	-	
13	-	
14	65	

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Quadratic Probing)

Örnek: 29, 16, 14, 99, 127 değerlerini hash tablosuna quadratic probing metoduyla sırayla yerleştiriniz.

$$h(x) = x \bmod 15$$

0	29	$t+1^2$
1	16	
2	47	
3	14	$t+2^2$
4	-	
5	35	
6	36	
7	-	
8	-	
9	129	
10	25	
11	2501	
12	-	
13	-	
14	65	$t = h(14) = 14$

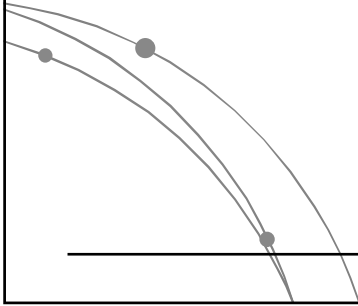
G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Quadratic Probing)

Örnek: 29, 16, 14, 99, 127 değerlerini hash tablosuna quadratic probing metoduyla sırayla yerleştiriniz.

$$h(x) = x \bmod 15$$



0	29
1	16
2	47
3	14
4	-
5	35
6	36
7	-
8	-
9	129
10	25
11	2501
12	-
13	99
14	65

Diagram illustrating quadratic probing for the value 99. The hash value $t = h(99) = 9$ is calculated. The probe sequence is t , $t+1^2$, and $t+2^2$. The value 99 is found at index 13, which is $t+2^2$. The values 129 and 2501 are marked with 'X' as they are not the target value.

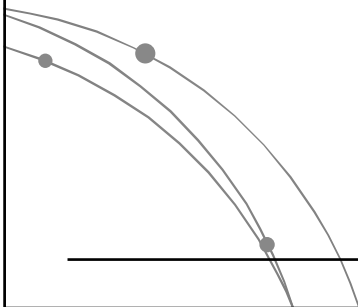
G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Quadratic Probing)

Örnek: 29, 16, 14, 99, 127 değerlerini hash tablosuna quadratic probing metoduyla sırayla yerleştiriniz.

$$h(x) = x \bmod 15$$



0	29
1	16
2	47
3	14
4	-
5	35
6	36
7	127
8	-
9	129
10	25
11	2501
12	-
13	99
14	65

Diagram illustrating quadratic probing for the value 127. The hash value $t = h(127) = 7$ is calculated. The probe sequence is t , $t+1^2$, and $t+2^2$. The value 127 is found at index 7, which is t . The values 129 and 2501 are marked with 'X' as they are not the target value.

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Quadratic Probing metodunun avantajları / dezavantajları

- Anahtar değerlerini linear probing metoduna göre daha düzgün dağıtır.
- Yeni eleman eklemeye tablo boyutuna dikkat edilmezse sonsuza kadar çalışma riski vardır. (Örn.: Boyutu 16 (0-15) olan bir tabloda 0, 1, 4 ve 9 pozisyonlarının dolu olduğu durumda 16 değerini eklemeye çalıştığımız zaman sonsuz döngüye girer.)

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Double Hashing)

- Aynı pozisyona gelen ikinci kayıt için ikinci bir hash fonksiyonu kullanılır.
- İkinci hash fonksiyonu 0 değerini alamaz.
- En çok kullanılan fonksiyon:

$$\text{hash}(x) = \text{hash}_1(x) + i * \text{hash}_2(x)$$

Örn.: $\text{hash}_2(x) = R - (x \% R)$, $R < \text{TableSize}$

$$\text{hash}(x) = \text{hash}_1(x)$$

$$\text{hash}(x) = \text{hash}_1(x) + 1 * \text{hash}_2(x)$$

$$\text{hash}(x) = \text{hash}_1(x) + 2 * \text{hash}_2(x)$$

$$\text{hash}(x) = \text{hash}_1(x) + 3 * \text{hash}_2(x)$$

...

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Çakışmanın giderilmesi (Double Hashing)

Örnek: 65 değerinin eklenmesi

$$\text{hash}_1(x) = x \% 15$$
$$\text{hash}_2(x) = 11 - (x \% 11)$$

$$\text{hash}(65) = \text{hash}_1(65)$$
$$\text{hash}(65) = 5 \text{ (Dolu)}$$

$$\text{hash}(65) = \text{hash}_1(65) + 1 * \text{hash}_2(65)$$
$$\text{hash}(65) = 5 + 1 = 6$$

$$\text{hash}(65) = \text{hash}_1(65) + 2 * \text{hash}_2(65)$$
$$\text{hash}(65) = 5 + 2 = 7$$

0	-
1	-
2	47
3	-
4	-
5	35
6	36
7	65
8	-
9	129
10	25
11	2501
12	-
13	-
14	-

$t = h_1(65) = 5$

X

X

$t+1 * h_2(65)$

$t+2 * h_2(65)$

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

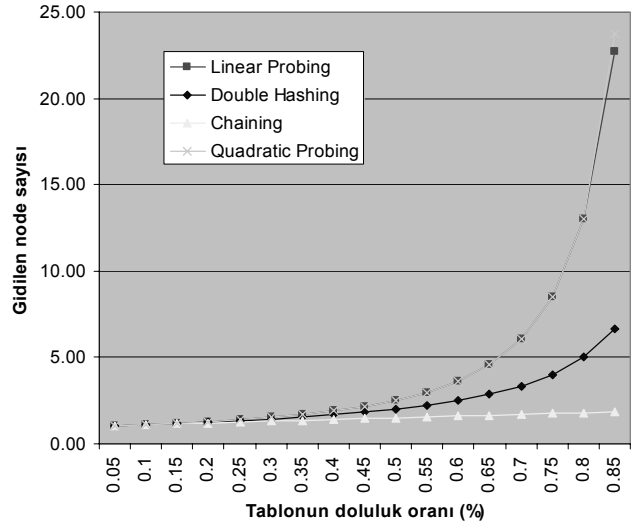
Double Hashing metodunun avantajları / dezavantajları

- Anahtar değerlerini linear probing metoduna göre daha düzgün dağıtır ve gruplar oluşmaz.
- Quadratic probing metoduna göre daha yavaştır çünkü ikinci bir hash fonksiyonu hesaplanır.

G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Performans



G. Ü. Bilgisayar Mühendisliği Bölümü

Hashing

Haftalık Ödev:

• 100 tane anahtar değerini rastgele üreten ve bu değerleri boyutu 100 olan bir hash tablosuna yerleştiren programı yazınız. Programda hash fonksiyonu olarak division metodunu, çakışma çözümü için linear probing ile quadratic probing metodlarını ayrı ayrı kullanınız.

- Anahtar değerlerini integer olarak alınız.
- Hash tablosu için dizi yapısını kullanınız.

G. Ü. Bilgisayar Mühendisliği Bölümü