



YZM 2116- VERİ YAPILARI

LAB#9: HASH TABLOLARI

Hash Tabloları

- İdeal bir hash tablo veri yapısı, içerisinde elemanlar barındıran **sabit bir diziden oluşur.**
- Dizi içerisindeki elemanlar, **index hesaplamasında kullanılacak anahtar (key)** isimli **özel bir üye** bulundurmamak **zorundadırlar.**
 - Anahtar; integer veya **string** bir değer olabilir.
 - **Örneğin:** Bir Öğrenci nesnesindeki Öğrenci No veya Öğrenci TC Kimlik No olabilir.
- Dizinin boyutu **TabloBoyutu** olup,
- Diziye eklenecek elemanlar 0'dan **(TabloBoyutu-1)**'e kadar olan **indekslerde** saklanırlar.
- **Anahtardan→Indekse** dönüştürme işlemine **Hashing**, bu işi yapan fonksiyona **Hashing fonksiyonu** denir.

Örnek


Elemanlar

Cem 1111

Ada 2222

Sibel 3333

İpek 4444


anahtar

anahtar →



Hash Tablosu

0	
1	
2	
3	Cem 1111
4	Ada 2222
5	
6	Sibel 3333
7	İpek 4444
8	
9	

Hash Fonksiyonu (devam...)

$$f(x) = x \% 15$$

ise

$$f(x) ?$$

if x = 25 129 35 2501 47 36

f(x) = 10 9 5 11 2 6

Anahtarların diziye yerleşimi aşağıdaki gibidir:

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

- Silme, ekleme ve arama işlem karmaşıklığı $O(1)$,
ancak **bir problem var???**

Hash Fonksiyonu (devam...)

Soru: Eğer $x = 65$ anahtarını eklemek istersek ne olur?

$$x = 65$$

$$f(x) = 5$$

Anahtarların diziye yerleşimi:

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

- 35 ve 65 anahtarları için $f(x)$ fonksiyonundan **aynı indis değeri** döndü.

Çakışma (*collision*) oldu...

Çakışma (Collision)

- Hashing işlemi sonucunda farklı anahtarlara sahip iki eleman, **aynı dizi indeks değeri** üretilebilir.
- Eğer **en az iki eleman** için aynı indeks değeri üretilirse bu duruma **çarpışma** denir.
- Çarpışma istenmeyen bir durumdur.
- **Çarpışmayı çözmek** için **iki yöntem** vardır:
 1. Ayrık zincirleme (Separate chaining)
 2. Açık adresleme (Open addressing)
 - I. Doğrusal ölçüm (Linear probing)
 - II. Karesel ölçüm (Quadratic probing)

Doğrusal Ölçüm (Linear Probing)

- Çakışma meydana geldiğinde, **doğrusal arama mantığıyla**, **uygun boş yerler** sırayla aranırlar.
- f doğrusal bir fonksiyon olup, $f(i) = i$
- Sırayla deneme işlemi gerçekleştirilir.
- Hash tablosunun sonuna gelindiyese, başa dönülür.
 - Döngüsel Kuyruk mantığında veya döngüsel dizi

Doğrusal Ölçüm (Linear Probing) (devam...)

- *TabloBoyutu = 15.*
- **65 elemanını** eklemek istiyoruz, ancak **5 numaralı indiste** zaten 35 elemanı var.
- *Çakışma* gerçekleşti, ne yapacağız?

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

Doğrusal Ölçüm (Linear Probing) (devam...)

- **Çözüm:** 65 Ekle

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

- Toplam **3 deneme** yaptık ve uygun yeri bulduk.



Uygulamayı indirip çalıştırın....

Ayrık Zincirleme Çözümü

- Aynı indis pozisyonuna gelen kayıtlar **bağlı listelerle** gösterilir.
- Çarpışma meydana gelirse ikinci eleman bir bağlı liste ile **birinci elemana bağlanır**.
- Bağlı listeler tek veya çift yönlü olabilir.

1. Ayrık Zincirleme Çözümü (devam...)

Anahtarlar: 0, 1, 4, 9, 16, 25, 36, 49, 64, 81 $\text{hash(anahtar)} = \text{anahtar} \% 10$

