

## BAĞLI LİSTELER

- Singly linked lists (Tek bağlı)
- Doubly linked lists (Çift bağlı)
- Circular lists
- Skip lists
- Self-organized lists
- Sparse tables

Yrd.Doç.Dr. M. Ali Akcayol

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

### Singly linked lists

- Dizilerin dezavantajları.
  - Boyut değiştirme zordur
  - Yeni bir eleman ekleme zordur
  - Bir elemanı silme zordur
  - Diziler tüm elemanları için hafızada yer ayırırlar

Metin

Bağlı dizilerle bu problemler  
aşılabilir.

Mahmut	0
Mehmet	1
Melek	2
Melike	3
Murat	4
Mustafa	5
Mürsel	6
	7

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

### Singly linked lists

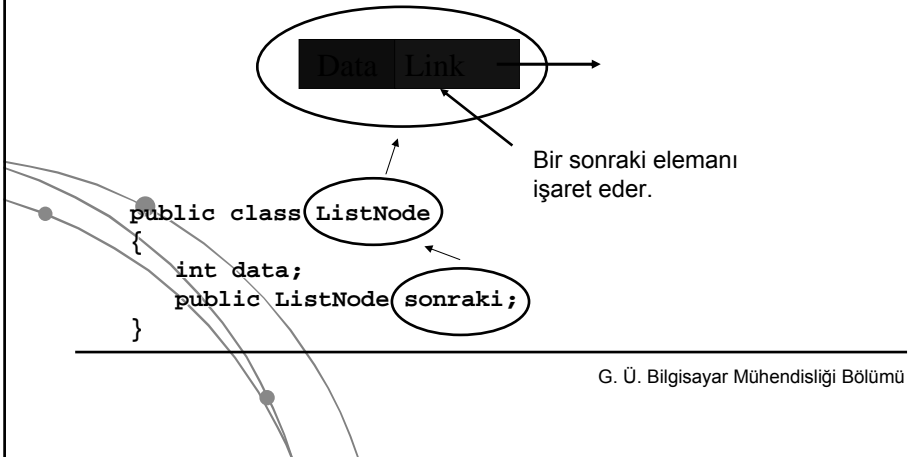
- Her dizi elemanı için ayrı hafıza alanı ayrılır.
- Bilgi kavramsal olarak sıralıdır ancak hafızada bulunduğu yer sıralı değildir.
- Herbir eleman (node) bir sonrakini gösterir.



## BAĞLI LİSTELER

### Singly linked lists

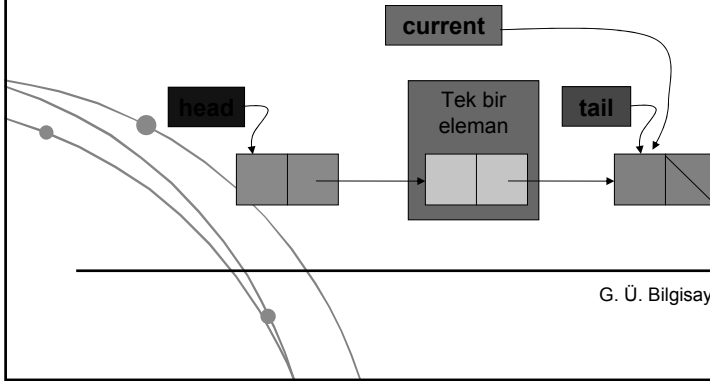
- Listedeki her bir eleman data ve link kısmından oluşur.
- Data kısmı içerisinde saklanan bilgiyi ifade eder.
- Link kısmı ise kendisinden sonraki elemanı işaret eder.



# BAĞLI LİSTELER

## Singly linked lists

- Listede bir başlangıç elemanı vardır (head).
- Listenin birde sonuncu elemanı vardır (tail).
- Listede aktif (current) eleman şu anda bilgilerine ulaşabileceğimiz elemandır.



G. Ü. Bilgisayar Mühendisliği Bölümü

# BAĞLI LİSTELER

## Singly linked lists (işlemler)

- Listeye eleman ekleme
  - Başa
  - Sona
  - Sıralı
- Listeden eleman silme
  - Baştan
  - Sondan
  - Tümünü
  - Belirlenen bilgiye sahip elemanı
  - İstenen sıradaki elemanı
- Arama
- Listeleme
  - İstenen bilgiye göre
- Kontrol
  - Boş liste
  - Liste boyutu

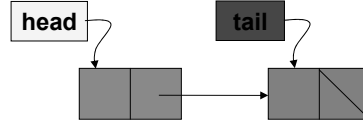
G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Singly linked lists (Liste oluşturma)

```
public class ListNode
{
    public string adSoyad;
    public ListNode sonraki;
    public ListNode(string adSoyad)
    {
        this.adSoyad = adSoyad;
    }
}

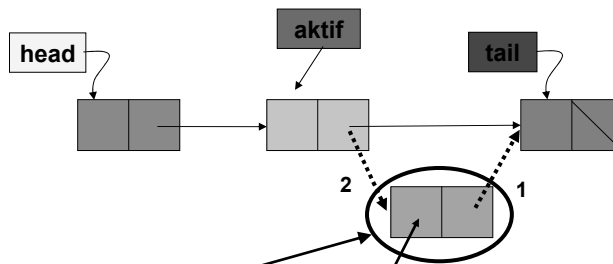
public class LinkedList
{
    public ListNode headNode, tailNode;
    public LinkedList()
    {
        headNode = new ListNode("head");
        tailNode = new ListNode("tail");
        headNode.sonraki = tailNode;
        tailNode.sonraki = tailNode;
    }
}
```



G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Singly linked lists (Eleman ekleme)

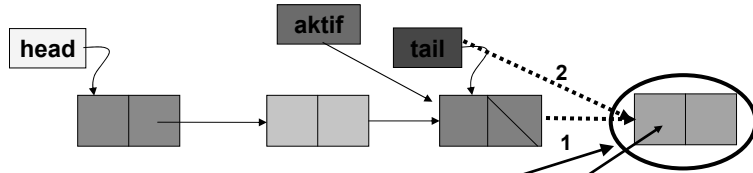


```
ListNode yeniNode = new ListNode("Mustafa");
1 yeniNode.sonraki = aktif.sonraki;
2 aktif.sonraki = yeniNode;
```

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Singly linked lists (Eleman ekleme)

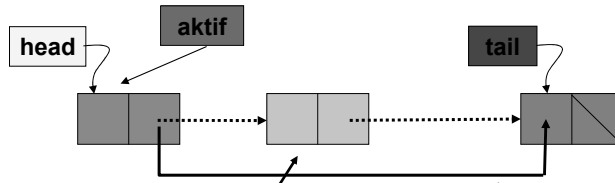


```
ListNode yeniNode = new ListNode("Mustafa");  
1 aktif.sonraki = yeniNode;  
2 tailNode = yeniNode;
```

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Singly linked lists (Eleman silme)

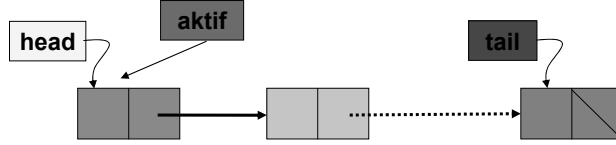


```
while ((aktif.sonraki != bagliListe.tailNode) && (aktif.sonraki != silinecekNode))  
{  
    aktif = aktif.sonraki;  
}  
aktif.sonraki = aktif.sonraki.sonraki;
```

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Singly linked lists (Eleman arama)



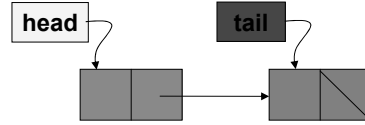
```
while ((aktif.sonraki != bagliListe.tailNode) && (aktif.sonraki != arananNode))
{
    aktif = aktif.sonraki;
}
```

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Singly linked lists (Boş liste)

```
if (head.sonraki == tail) listeBos = true
else listeBos = False;
```



Eleman sayısı:

```
int i = 0;
ListNode aktif = bagliListe.headNode;
while (aktif.sonraki != bagliListe.tailNode)
{
    aktif = aktif.sonraki;
    i++;
}
listeBoyutu = i;
```

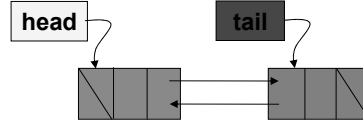
G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Doubly linked lists (Liste oluşturma)

```
public class ListNode {
    public string adSoyad;
    public ListNode oncesi, sonraki;
    public ListNode(string adSoyad)
    {
        this.adSoyad = adSoyad;
    }
}

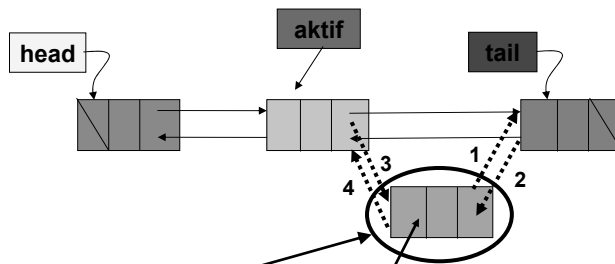
public class LinkedList {
    public ListNode headNode, tailNode;
    public LinkedList()
    {
        headNode = new ListNode("head");
        tailNode = new ListNode("tail");
        headNode.oncesi = headNode;
        headNode.sonraki = tailNode;
        tailNode.oncesi = headNode;
        tailNode.sonraki = tailNode;
    }
}
```



G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Doubly linked lists (Eleman ekleme)

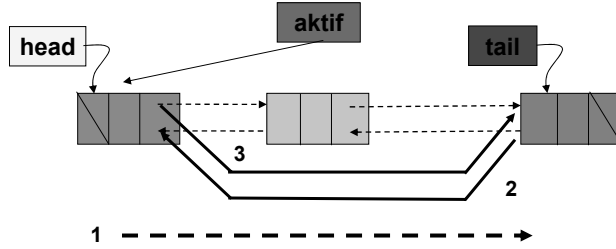


```
ListNode yeniNode = new ListNode("Mustafa");
1 yeniNode.sonraki = aktif.sonraki;
2 aktif.sonraki.oncesi = yeniNode;
3 aktif.sonraki = yeniNode;
4 yeniNode.oncesi = aktif;
```

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Doubly linked lists (Eleman silme)

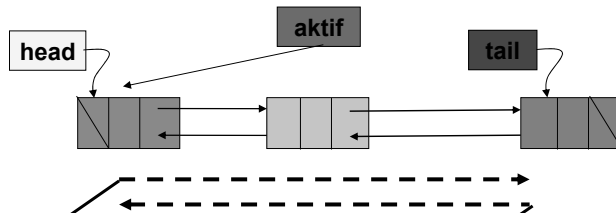


```
while ((aktif.sonraki != bagliListe.tailNode) && (aktif.sonraki != silinecekNode))
{
1   aktif = aktif.sonraki;
}
2 aktif.sonraki.sonraki.onceki = aktif;
3 aktif.sonraki = aktif.sonraki.sonraki;
```

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Doubly linked lists (Eleman arama)



```
while ((aktif.sonraki != bagliListe.tailNode) && (aktif.sonraki != arananNode))
{
1   aktif = aktif.sonraki;
}
while ((aktif.onceki != bagliListe.headNode) && (aktif.onceki != arananNode))
{
2   aktif = aktif.onceki;
}
```

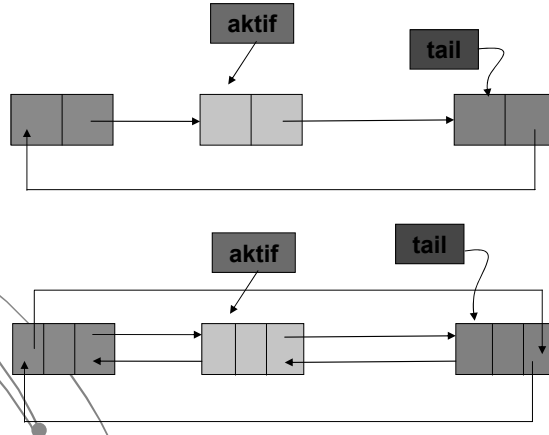
G. Ü. Bilgisayar Mühendisliği Bölümü



# BAĞLI LİSTELER

Circular lists

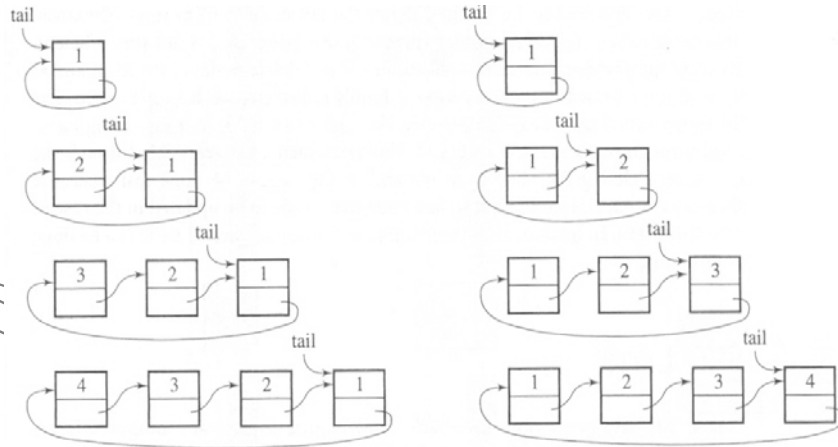
Sadece 1 tane sabit node vardır (tailNode)



G. Ü. Bilgisayar Mühendisliği Bölümü

# BAĞLI LİSTELER

Circular lists

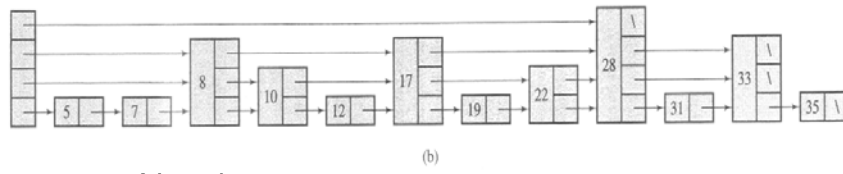
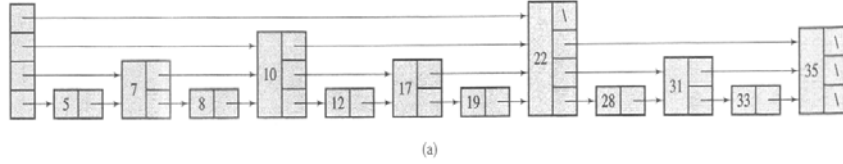


Listenin başına ve sonuna eleman ekleme

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

### Skip lists

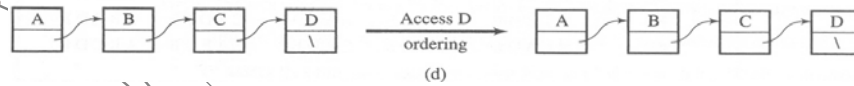
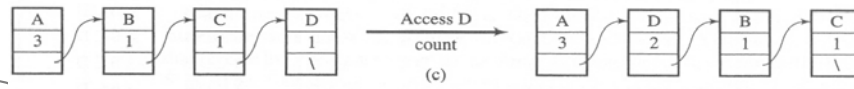
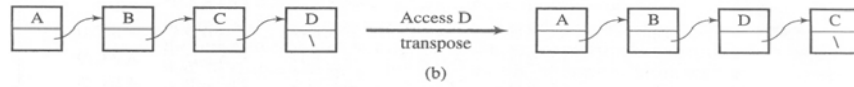
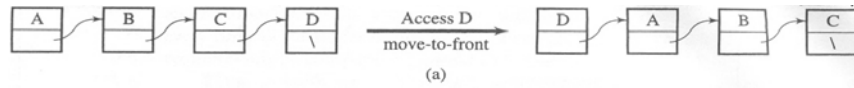


Eleman ekleme ve silme zordur. Listenin yeniden seviyelendirilmesi gerekir.  
Arama işlemi çok hızlıdır.

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

### Sel-organizing lists



Move-to-front: Erişilen elemanı listenin başına alır  
Transpose: Erişilen elemanı önündekiyle yer değiştirir  
Count: Erişilme sayısına göre sıralar  
Ordering: İstenen kritere göre sıralar

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

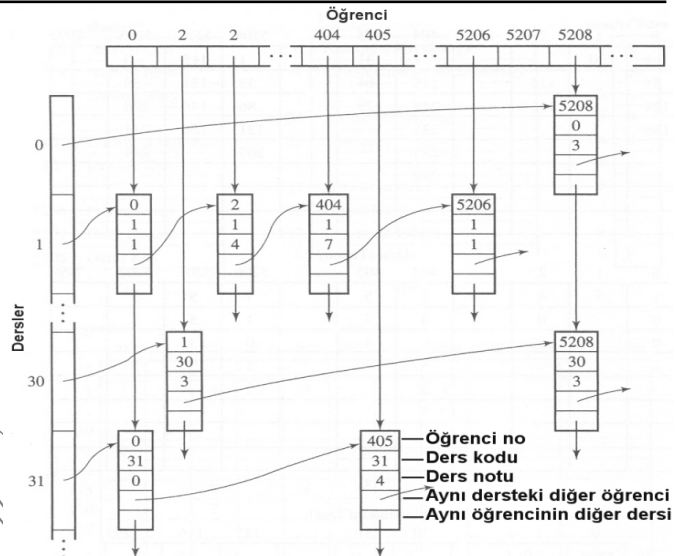
Sparse tables

		Öğrenci											
Notlar		0	1	2	...	404	405	...	5206	5207	5208	...	7999
Dersler	0										3		
	1	1		4		7			1				
	:												
	30		5								3		
	31	0					5						
	:												
	115			0		4				5			
	116			3									
	117												
	:												
299													

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Sparse tables  
Bağlı dizilerle  
gösterim



G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

Linked Lists (Gazi Üniversitesi Bilgisayar Mühendisliği Bölümü)

Linked List

1

head  
0429001 Kemal 80  
0429002 Mehmet 75  
0429003 Ahmet 85  
0429004 Pinar 82  
0429005 Betül 78  
tail

Numara 0429001 2

Ad Soyad Kemal

Ortalama 80 8

Liste Boş

Ekleme İşlemi

3

Başa  
 Sıraya  
 Sona

Ekle

Silme İşlemi

4

Baştakini  
 Sondakini  
 Tümünü  
 Girilen Numarayı  
 Girilen Sırayı

Sıra 0

Sil

Listeleme İşlemi

5

Numaraya Göre  
 Ada Soyada Göre  
 Ortalamaya Göre

Listele

Liste Boyutu

7

Liste Boyutu 0

Liste Boyutu

Yrd.Doç.Dr. M. Ali Akcayot  
Gazi Üniversitesi Bilgisayar Mühendisliği Bölümü

G. Ü. Bilgisayar Mühendisliği Bölümü

## BAĞLI LİSTELER

### Haftalık Ödev:

K adet ders için N adet öğrencinin numara ve harf ortalamalarını bağlı dizilerle gösteriniz.

- Herbir node öğrenci numarası, dersin kodu, dersin harf ortalaması bilgilerini bulunduracak.
- Her öğrencinin numarası headNode içindeki bilgi olacak.
- Her dersin kodu headNode içindeki bilgi olacak.
- Herbir node aynı derste bir sonraki öğrenciyi gösterecek.
- Herbir node aynı kişinin diğer dersini gösterecek.
- Program Windows application olarak hazırlanacak ve aşağıdaki işlemleri butonlarla yapacak.

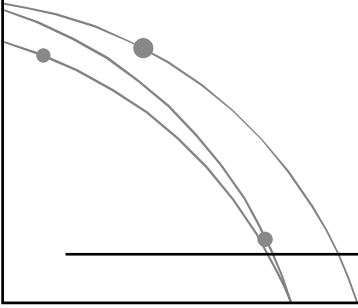
- 1- Bir öğrenciye yeni bir ders ekleme
- 2- Bir derse yeni bir öğrenci ekleme
- 3- Bir öğrencinin bir dersini silme
- 4- Bir derste bir öğrenciyi silme
- 5- Bir derste tüm öğrencileri numara sırasına göre sıralı listeleme
- 6- Bir öğrencinin aldığı tüm dersleri ders koduna göre sıralı listeleme

G. Ü. Bilgisayar Mühendisliği Bölümü

## **BAĞLI LİSTELER**

---

### **Gelecek Hafta Yığınlar (Stacks)**



G. Ü. Bilgisayar Mühendisliği Bölümü