BAĞLI LİSTELER (LINKED LISTS)

Aynı kümeye ait verilerin bellek üzerinde bir gösterici (pointer) yardımı ile birbirine bağlanması sonucu oluşturulan veri yapısıdır. Bu veri yapısındaki bütün elemanların ortak noktası kendi içlerinde bağlantı bilgisi içeren kendi türünden bir ya da daha fazla göstericiye sahip olmasıdır. Veri yapısı, kümedeki herhangi bir elemanın, kendisinden önce ve kendisinden sonra hangi elemanın bulunduğu bilgisi bu göstericilere değer vererek kurulur. Bağlı listeler dinamik veri yapılarıdır. Bellek üzerinde tanımlanan elemanlar göstericiler yardımıyla birbirlerine bağlanarak bağlı liste yapısı oluşturulur. Listeye istenirse dizilerde olduğu gibi eleman ekleme ve silme işlemleri uygulanabilir.

BAĞLI LİSTELER (LINKED LISTS) (DEVAM)

Bağlı liste kullanımının dizi kullanımından farkı programcının, başta kaç adet elemanla işlem yapılacağını bilmek zorunda olmamasıdır. Listenin boyutu dizilerde olduğu gibi önceden belirlenmez ve dinamik olarak yaratılır. Kaç adet elemanla çalışacağımızı bilmediğimiz durumlarda bağlı liste kullanmak bize büyük avantajlar sağlar çünkü sadece birkaç elemanın kullanılacağı bir diziyi, ne kadarlık bir bellek alanı ayıracağımızı bilmeden -örneğin ***1000*** elemanı barındıracak şekilde- yaratmak bize bellek kullanım miktarı açısından büyük çaplı bir kayba mal olacaktır.

BAĞLI LİSTELER (LINKED LISTS) (DEVAM)

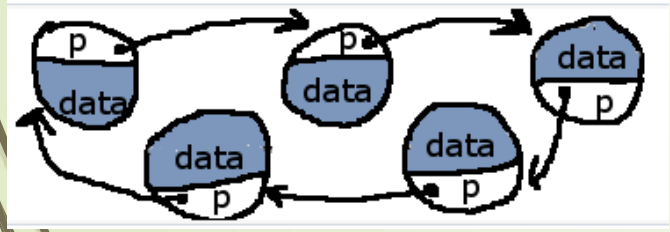
Tek Yönlü Bağlı Listeler : Listedeki elemanlar arasında sadece tek bir bağ vardır. Bu tür bağlı listelerde hareket yönü sadece listenin başından sonuna doğrudur. (Şekil - 1)

Çift Yönlü Bağlı Listeler : Listedeki elemanlar arasında iki yönlü bağ vardır. Elemanın bağlantı bilgisi bölümünde iki gösterici bulunur. Bu göstericinin biri kendisinden sonra gelen elemanı diğeri ise kendisinden önce gelen elamanın adres bilgisini tutar. Bu sayede listenin hem başından sonuna hem de listenin sonundan başına doğru hareket edilebilir. Bu yöntem daha esnek bir yapıya sahip olduğundan bazı problemlerin çözümünde daha işlevsel olabilmektedir. (Şekil - 2)

Dairesel Bağlı Listeler : Listedeki elemanlar arasında tek yönlü bağ vardır. Tek yönlü bağlı listelerden tek farkı ise son elemanın göstericisi ilk listenin ilk elamanının adresini göstermesidir. Bu sayade eğer listedeki elemanlardan birinin adresini biliyorsak listedeki bütün elemanlara erişebiliriz.(Şekil - 3)

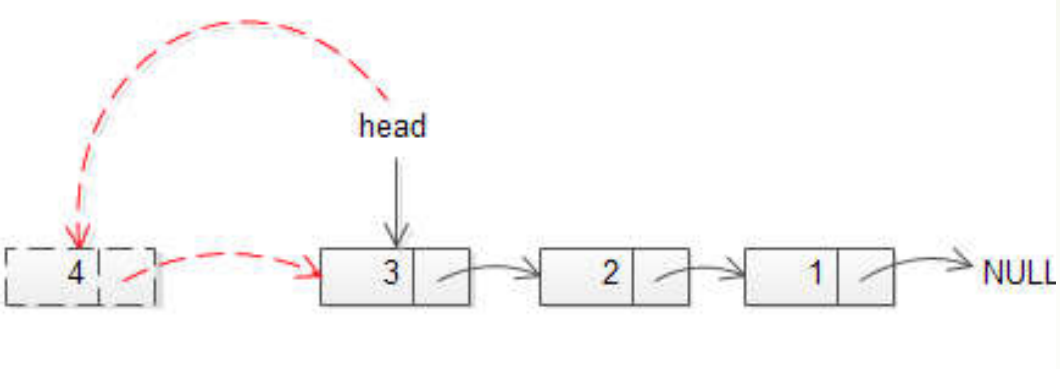
Şekil 1-Tek Yönlü Bağlı Listeler

Şekil 2-Çift Yönlü Bağlı Listeler

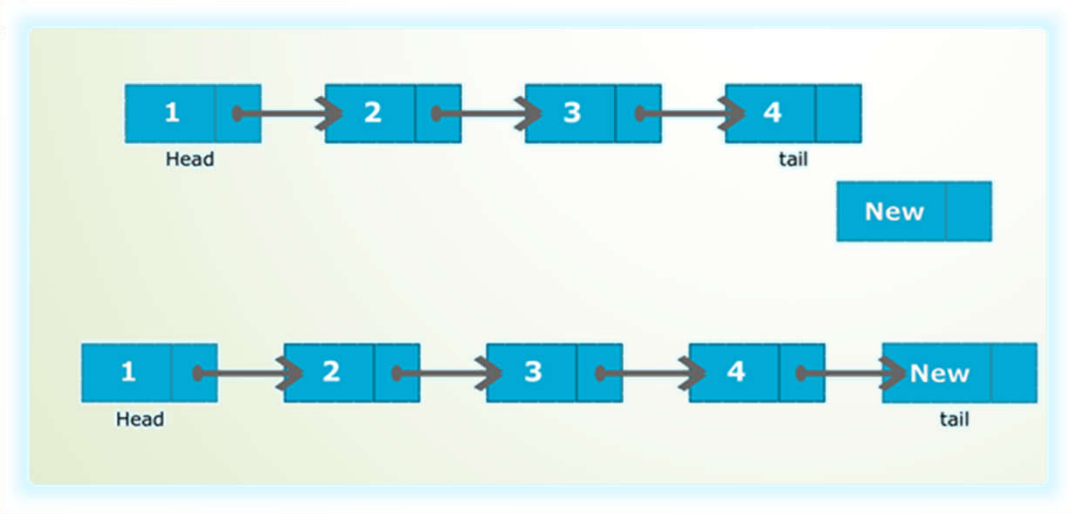


Şekil 3-Dairesel Bağlı Listeler

BAĞLI LİSTELER (LINKED LISTS) (KOD- Öne Ekle, Sona Ekle, Araya Ekle,Silme, Yazdırma)



Öne Ekle



Sona Ekle

BAĞLI LİSTELER (LINKED LISTS) (KOD- Öne Ekle, Sona Ekle, Araya Ekle,Silme, Yazdırma)

```
public class Dugum
    {
        public Dugum next;
        public Object Veri;
    }

    public class LinkedList
    {
        private Dugum head;

        public void dugumYazdir()
        {
            Dugum aktif = head;
            while (aktif != null)
            {
                Console.WriteLine(aktif.Veri);
                aktif = aktif.next;
            }
        }
```

BAĞLI LİSTELER (LINKED LISTS) (KOD- Öne Ekle, Sona Ekle, Araya Ekle,Silme, Yazdırma)

```
public void OneEkle(Object Veri)
        {
            Dugum eklenecek = new Dugum();

            eklenecek.Veri = Veri;
            eklenecek.next = head;

            head = eklenecek;
        }
```

```
public void SonaEkle(Object Veri)
        {
            if (head == null)
            {
             Dugum head = new Dugum();

                head.Veri = Veri;
                head.next = null;
            }
            else
            {
                Dugum eklenecek = new Dugum();
                eklenecek.Veri = Veri;

                Dugum aktif = head;
                while (aktif.next != null)
                {
                    aktif = aktif.next;
                }

                aktif.next = eklenecek;
            }
        }
    }
```

BAĞLI LİSTELER (LINKED LISTS) (KOD- Öne Ekle, Sona Ekle, Araya Ekle,Silme, Yazdırma)

Bağlı Liste (Linked List) – Araya Eleman Ekleme

Bağlı listelerde araya eleman ekleme işlemi hem kolay hem zordur. Dizilerle karşılaştıracak olursak araya eleman ekleme işleminde kaydırma yapmak zorunda kalmayız. Ancak dizilerdeki gibi de Random Access, daha doğrusu direkt erişim yapamayız. Yani 3. indise eleman eklemek istiyorsak, direkt olarak 3. indise varamayız.

Peki Bağlı liste araya eleman ekleme işlemi nasıl oluyor? bunu **Bağlı Liste sona eleman ekleme** işlemi gibi düşünebilirsiniz. Tek fark, gidilecek yer listenin sonu değil, her hangi bir yer olmalıdır.

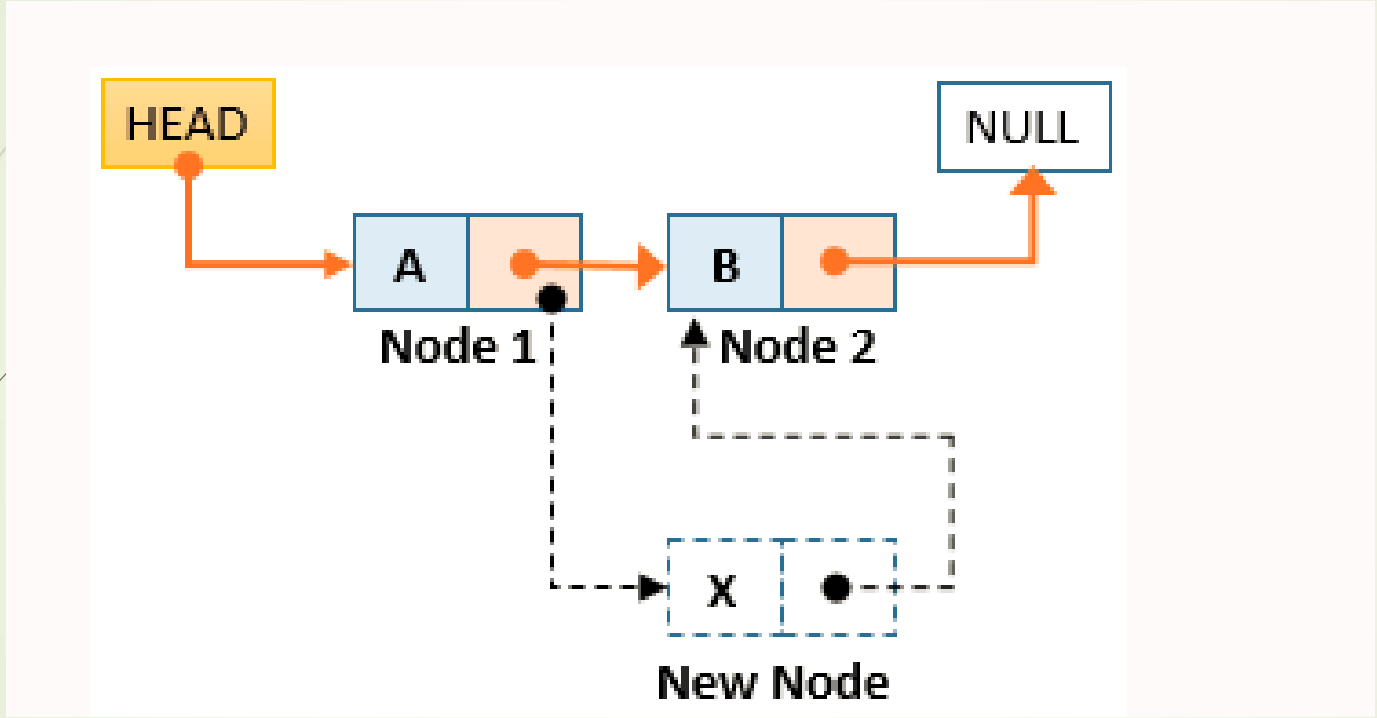
3 => 6 => 9 => 15 =>18

Yukarıdaki gibi bir bağlı listemiz olsun. Ve biz ikinci indisin (yani 9) sonuna 12 sayısını eklemek isteyelim. Bu işlemi yapmak için 9 sayısına kadar traverse yapmalıyız. ve 9'un next değerine eklemek istediğimiz değeri eklemeliyiz.
Yani 9 sayısını tutan düğüm "**aranan**" olsun. Mecut durumdaki (sayımızı eklemeden önce) 9 sayısının next değerini(15) de **eklenecek** düğümünü atayalım.

```
eklenecek.sonraki = aktif.sonraki;
        aktif.sonraki = eklenecek;
```

diyerek eklemeyi yapıyoruz.

BAĞLI LİSTELER (LINKED LISTS) (KOD- Öne Ekle, Sona Ekle, Araya Ekle,Silme, Yazdırma)



Araya Ekle

BAĞLI LİSTELER (LINKED LISTS) (KOD- Öne Ekle, Sona Ekle, Araya Ekle,Silme, Yazdırma)

```
public void arayaEkle(object v,int konum)
        {
            int sayac = -1;
            Dugum eklenecek = new Dugum();
            eklenecek.veri = v;

            Dugum aktif = head;
            do
            {
                aktif = aktif.sonraki;
                sayac++;
            } while (sayac != konum-1);
            //ekleneceği aktifin sonrakine ata
            eklenecek.sonraki = aktif.sonraki;
            aktif.sonraki = eklenecek;

        }
    }
```

BAĞLI LİSTELER (LINKED LISTS) (KOD- Öne Ekle, Sona Ekle, Araya Ekle,Silme, Yazdırma)

Bağlı Liste (Linked List) – Aradan Eleman Silme

Bağlı Listelerde Aradan eleman silme işlemi sondan ve baştan eleman silme işlemi gerçekleşmediğinde yapılan tüm silme işlemlerinde geçerlidir. Aradan eleman silme işleminde kullanıcı bir değer girer ve liste taranarak silme işlemi gerçekleştirilir. Ancak biz dizilerdeki gibi direkt silme işlemi gerçekleştiremeyiz. Çünkü Bağlı listelerde tüm elemanlar birbirleri ile ilişkilidir. Eğer ilişkiyi bozarsak listemiz de bozulur. Şimdi aşağıya bir bakalım

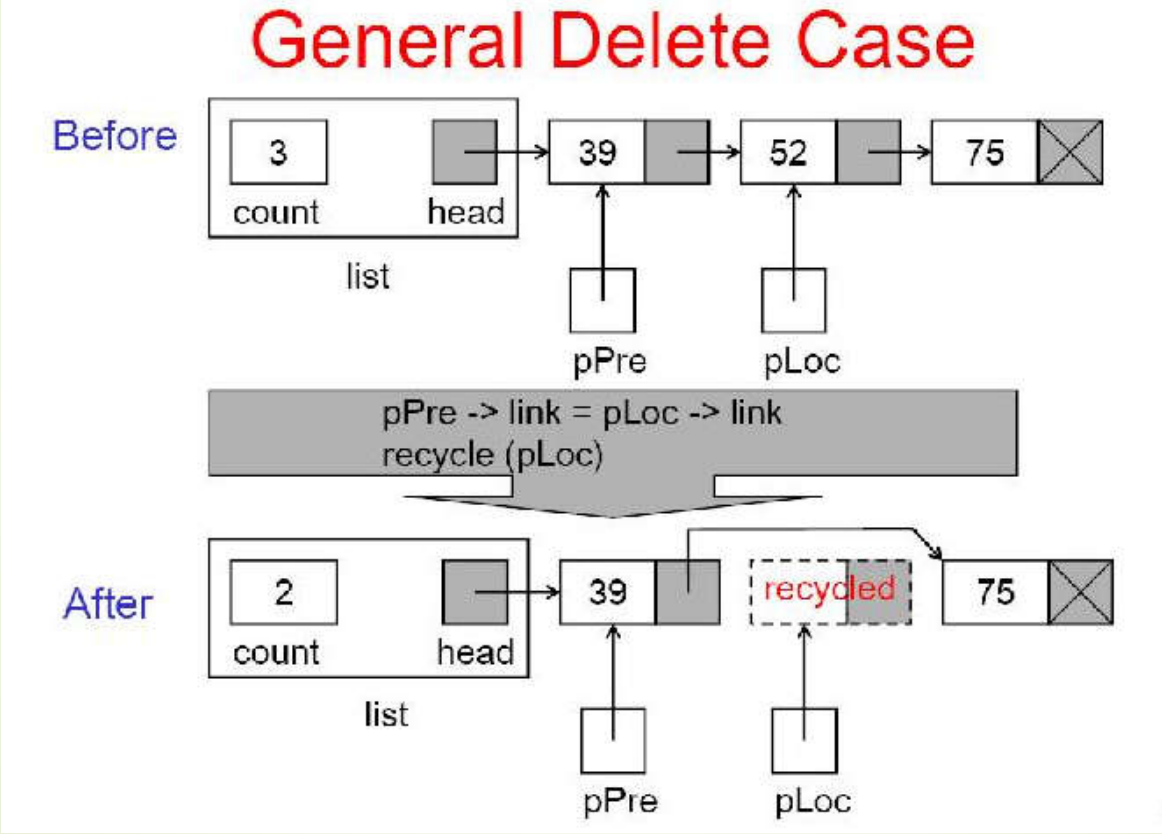
22 => 33 => 44 => 55 => 66 => 77
Yukarıdaki bağlı listeden 55 sayısını silmek istediğimizde aşağıdaki durum oluşur.
22 => 33 => 44 => 66 => 77

Bunu yazmak çok basit gibi öyle değil mi? dikkatlice bakarsanız silmek istediğimiz sayının bir öncesindeki sayı olan 44'ün, bir sonrasındaki sayı olan 66'yı işaret ettiğini görürsünüz. Yani mesele yalnızca 55 sayısını silmek değildir, 44 sayısının 55'i değil, 66'yı next değeri olarak göstermesini sağlamaktır.

O halde bizim traverse yaparken bulmamız gereken sayı 55 değil, 44'tür. yani statementımız temp->next != 55 olmalıdır. Böylece temp değeri 44 haline gelecektir.
44 sayısını temp'e atadıktan sonra işimiz kolay. temp->next silinecek olan sayi, temp->next->next ise bizim yeni "sonraki sayımızdır.

BAĞLI LİSTELER (LINKED LISTS) (KOD- Öne Ekle, Sona Ekle, Araya Ekle,Silme, Yazdırma)

Bağlı Liste (Linked List) – Aradan Eleman Silme



BAĞLI LİSTELER (LINKED LISTS) (KOD- Öne Ekle, Sona Ekle, Araya Ekle,Silme, Yazdırma)

```
public void silme(string silinecekVeri)


    {
        Dugum silinecek=new Dugum();
        silinecek.veri = silinecekVeri;
        Dugum dg = head;

        Dugum onceki = null;
        while (dg != null)
        {
            if (dg.veri == silinecek.veri)
            {
                break;
            }
            onceki = dg;
            dg = dg.sonraki;
        }
        if (dg != null)
            {
                if (onceki == null)
                {
                    if (onceki == tail)
                    {
                        head = null;
                        tail = null;
                    }
                    else
                    {
                        head = head.sonraki;
                    }
                }
                else
                {
                    onceki.sonraki = dg.sonraki;
                    if (onceki.sonraki == null)
                    {
                        tail = onceki;
                    }
                }
              }
            else
            {
               Console.WriteLine("silme olmadı");

            }
```

BAĞLI LİSTELER (LINKED LISTS) (KOD- Öne Ekle, Sona Ekle, Araya Ekle,Silme, Yazdırma)

```
class Program
    {
        static void Main(string[] args)
        {
            BagliListe bListe1 = new BagliListe();
            bListe1.oneEkle("veri");
            bListe1.oneEkle("yapilari");
            bListe1.oneEkle("dersi");
            bListe1.dugumYazdir();

            BagliListe bListe2 = new BagliListe();
            bListe2.sonaEkle("veri");
            bListe2.sonaEkle("yapilari");
            bListe2.sonaEkle("dersi");
            bListe2.sonaEkle("bilisim");
            bListe2.arayaEkle("deneme", 1);
            bListe2.dugumYazdir();
            bListe2.silme("dersi");
            bListe2.dugumYazdir();
Console.ReadLine();
        }
    }
```