



Bağlı Listeler

Günlük yaşamda listeler bilgisayar programlarında da yaygın olarak kullanılan veri modellerindendir.

Programlama açısından liste, aralarında doğrusal ilişki olan veriler topluluğu olarak görülebilir.

Yığın ve kuyrukların genişletilmesi yani üzerlerindeki sınırlamaların kaldırılması ile liste yapısına ulaşılır.

Bağlı listeler ise verilerin birbirlerine bir işaretçi ile bağlandıkları doğrusal veri yapılarıdır.

Bağlı listeler bellek bölgelerinde tutuluyorsa, bu bellek bölgesinin adresinin değerini öğrenebilecek bir yapı olması gerekir. Yani yapının kendisi gibi bir yapıyı gösteren işaretçi üyesine sahip olması gerekir.

Bağlı Listeler

Dizi (sıralı bellek) kullanımının en büyük dezavantajları, kullanılmasına bağlı olmaksızın sabit miktarda belleğin ayrılması ve bu bellek miktarının sabit olmasından dolayı yetersiz kalması durumunda değiştirilememesidir.

Bağlı listeler bellekten gerektiği kadar yer kullandığından etkin ve sınırları olmayan bir yapı sağlamaktadır. Bir sonraki düğümün adres bilgisini de saklaması daha fazla bellek kullanımı gibi gözüktüğü de, bir dizinin boyutu sabit olduğundan ve boyutun aşılmaması için yüksek tutulduğundan, bağlı liste diziyeye oranla daha etkin ve daha az bellek kullanmaktadır.

Bağlı Listeler

Diziler(arrays), doğrusal listeleri oluşturan yapılardır. Bu yapıların özellikleri şöyle sıralanabilir:

- Doğrusal listelerde süreklilik vardır. Dizi veri yapısını ele alırsak bu veri yapısında elemanlar aynı türden olup bellekte art arda saklanırlar.
- Dizi elemanları arasında başka elemanlar bulunamaz. Diziyeye eleman eklemek gerektiğinde (dizinin sonu hariç) dizi elemanlarının yer değiştirmesi gerekir.
- Dizi program başında tanımlanır ve ayrılacak bellek alanı belirtilir. Program çalışırken eleman sayısı artırılmaz veya eksiltilemez.
- Dizinin boyutu baştan çok büyük tanımlandığında kullanılmayan alanlar oluşabilir.
- Diziyeye eleman ekleme veya çıkarmada o elemandan sonraki tüm elemanların yerleri değişir. Bu işlem zaman kaybına neden olur.
- Dizi sıralanmak istendiğinde de elemanlar yer değiştireceğinden karmaşıklık artabilir ve çalışma zamanı fazlalaşır.

Bu olumsuzluklar sıralı liste yapıları ile çözülebilir.

Bağlı Listeler

Doğrusal veri yapılarında dinamik bir yaklaşım yoktur. İstenildiğinde bellek alanı alınamaz ya da eldeki bellek alanları iade edilemez. Bağlantılı listeler dinamik veri yapıları olup yukarıdaki işlemlerin yapılmasına olanak verir. Bağlantılı listelerde düğüm ismi verilen bellek büyüklükleri kullanılır.

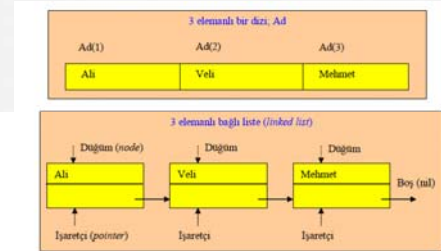


Tek yönlü bağlantılı listenin düğüm yapısı

Bağlantılı listeler çeşitli tiplerde kullanılmaktadır. Bunlar şu şekildedir:

- Tek yönlü sıralı doğrusal bağlantılı liste
- İki yönlü sıralı doğrusal bağlantılı liste
- Tek yönlü sıralı dairesel bağlantılı liste
- İki yönlü sıralı dairesel bağlantılı liste

Bağlı Listeler



Bağlı Listeler

Sıralı Liste yapısı

Bu veri yapısı iki ayrı alandan oluşur.

Adres	Veri alanı	bağ
1	H	0
2		
3	C	4
4	E	8
5		
6		
7	B	3
8	F	1

Liste başı →

Listede bulunan elemanlar fiziksel olarak sıralı olmak durumunda değildir. Yukarıdaki liste alfabetik karakterleri sıralı olarak saklayan bir listedir. Listede alfabetik olarak en küçük olan elemanın adresi bilinir. Liste başı işaretlendikten sonra büyüklük sırasına göre bir sonraki elemanın adresleri bağ alanlarına yazılarak liste sıralanır. En son elemanın bağ alanında sıfır bulunur, bu listenin bittiğini gösterir.

Dr. Sıgım TUNCEL

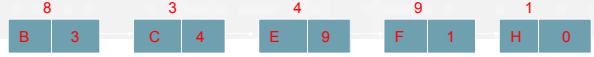
7

Bağlı Listeler

Bağlantılı listelerde bir ilk düğüm olması gerekir. Bu düğümün adresi bir değişkende tutularak listedeki diğer düğümlere bu adresten ilerleyerek ulaşılabilir. Diğer bir deyişle listedeki düğümler birbirlerinin adreslerini tutarlar. Son düğümün bağ alanına '0' yazılarak listenin bittiği belirtilir.

Tek yönlü sıralı doğrusal listeye düğüm ekleme

Sıralı liste yapısındaki örneği olarak bu listeye 'G' elemanını ekleyelim.



Liste başı

Boşlar listesinden boş bir düğüm alınarak düğümün adresi bir değişkene atanır.



X=5

Dr. Sıgım TUNCEL

8

Doğrusala Bağlı Liste

- Listedeki her düğümde bir sonraki düğümün adresinin tutulduğu veri yapısı (doğrusal) bağlı liste olarak adlandırılır.
- Listenin her bir elemanına düğüm (node) adı verilir. Düğümler, bilgi ve bağ (adres) sahalarından oluşmaktadır.
- Bağ sahalarında işaretçiler kullanılmaktadır.
- Listenin ilk elemanına dışarıdan bir işaretçi (list) ile erişilmektedir.
- Diğer düğümlere de bağlar yardımı ile ulaşılabilir.
- Son düğümün sonraki adres (next) sahası NULL değerini içerir.
- NULL bağı, liste sonunu belirtir.
- Elemanı olmayan liste boş liste olarak adlandırılır.
- Herhangi bir boyutta dinamik olarak genişletilip daraltılabilen yığın ve kuyrukların gerçekleştirimi bağlı listeler üzerinde yapılmaktadır.

Dr. Sıgım TUNCEL

9

Listeler Üzerindeki Bazı işlemler ve Tanımları

- EmptyList(List) : returns Boolean:**
Listenin boş olup olmadığını belirleyen fonksiyon.
- FullList(List) : returns Boolean**
Listenin dolu olup olmadığını belirleyen fonksiyon.
- LengthList(List) : returns integer**
Listedeki eleman sayısını bulan fonksiyon.
- InsertElement(List, NewElement)**
Listeye yeni bir eleman ekleyen fonksiyon.
- DeleteElement(List, Element)**
Listeden bir elemanı arayarak çıkartan fonksiyon.
- DestroyList(List)**
Listedeki tüm elemanları silerek boş liste bırakan fonksiyon.
- GetNextItem(List, Element)**
Etkin elemandan bir sonraki döndüren fonksiyon
- RetrieveElement(List, Element, Found)**
Elemanın listede olup olmadığını bulan ve döndüren fonksiyon

Dr. Sıgım TUNCEL

10

Bağlı (Bağlaçlı) Listeler

Kendi tipindeki bir yapıyı gösteren bir işaretçi üyesine sahip yapılara self-referential structures adı verilir.

```
struct node {
    char info;
    struct node *next; };

```

Yukarıdaki yapı yapısı, info adlı karakter tipli bilgi elemanının yanında, bir düğüm yapısında bir bellek bölgesine işaret eden next işaretçisine sahiptir.

Bu tür yapıların arka arkaya birbirine bağlanması mantığı listelerde, yığınlarda, kuyruklarda ve ağaçlarda oldukça yararlıdır.

Dr. Sıgım TUNCEL

11

Bağlı listelerde elemanların sıralı olacak şekilde uygulanmasına sıralı bağlı listeler denilmektedir.

Sıralı olmayan bağlı listeler ile aralarındaki fark; eleman ekleme işleminin, eklenecek elemanın listeye sıralı tutacak şekilde liste üzerinde dolaşarak araya eklenmesidir. Eleman çıkarma işlemi sıralı olmayan bağlı listeler ile aynıdır; silinmek istenen eleman bulunana kadar liste üzerinde dolaşılıp elemanın listeden çıkarılmasıyla sağlanır.

Dr. Sıgım TUNCEL

12

Yığın, Kuyruk ve Bağlı Listeler

Yığınlarda ve kuyrukların gerçekleştiriminde sıralı bellek kullanımının (dizi) en büyük **dezavantajı**, hiç kullanılmasa veya bir kısmı kullanılsa bile sabit miktardaki belleğin bu yapılara ayrılmış olarak tutulmasıdır.

Ayrıca sabit bellek miktarı aşıldığında da taşma oluşması ve eleman ekleme işleminin yapılamamasıdır. Bağlı listeler üzerinde gerçekleştirildiklerinde ise bu problemler ortadan kalkmaktadır. Bellekten sadece gerektiği kadar yer ayrılmakta ve bellek boyutu bitene kadar bu yapılara ekleme işlemi yapılabilmektedir.

Dr. Sıgım TUNCEL

13

Bağlı Liste Veri Yapısı ve Avantajı

- ❖ Başka veri yapısının gerçekleştiriminde kullanılabildikleri gibi kendileri de veri yapılarıdır.
- ❖ Elemanların eklenme ve çıkarılmasında bir sınırlama yoktur. Başa , sona ve araya eleman eklenebilir / çıkarılabilir.
- ❖ Dolayısıyla herhangi bir elemanına erişilebilir. n. elemanına erişmek için n tane işlem yapmak başka bir deyişle kendinden önceki (n-1) eleman üzerinden geçmek gerekmektedir. Elemanların bellekteki yerleri dizilerdeki gibi sıralı olmadığından elemanlar ve sıraları ile yerleştikleri bellek bölgeleri arasında bir ilişki yoktur.
- ❖ Bağlı listelerin diziler üzerine avantajı, bir grup eleman arasına eleman eklemeye ve bir grup eleman arasından eleman çıkarmada ortaya çıkar.

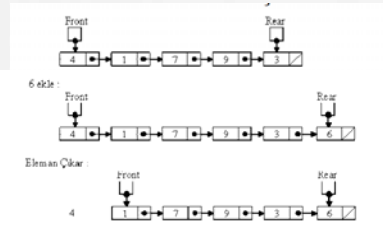
Dizilerde bir eleman silerken arada boşluk kalmasını engellemek için ilerisindeki (sağındaki) tüm elemanları bir geriye (sola) kaydırmak gerekir. Eleman eklemeye de yer açmak için konulacağı yerdeki ve ilerisindeki elemanları bir ileriye (sağa) kaydırmak gerekecektir. Kaç tane elemanın yer değiştireceği (biri kaydırılacağı) dizi boyutuna bağlı olarak ve eklenecek elemanın yerine bağlı olarak değişecektir. Bağlı listelerde ise eleman ekleme ve çıkarma için yapılan iş liste boyutundan bağımsızdır.

Dr. Sıgım TUNCEL

14

Kuyrukların Bağlı Liste Gösterimi

Eleman Ekleme ve Çıkarma



Dr. Sıgım TUNCEL

15

Öncelik Kuyruklarının Bağlı Liste Gerçekleştirimi

- **Yöntem 1 : (Sıralı liste tutularak)** (Artan sıralı öncelik kuyruğunda) Eleman ekleme, eklenecek elemanın listeyi sıralı tutacak şekilde liste üzerinde dolaşarak araya eklenmesi şeklinde gerçekleştirilir. Eleman çıkarma da, listenin ilk elemanının (en küçük değer) çıkarılması ile gerçekleştirilir.
 - **Yöntem 2 : (Sıralı olmayan liste)** (Artan sıralı öncelik kuyruğunda) Eleman ekleme kuyruğun herhangi bir yerine yapılabilir. Eleman çıkarma ise eleman bulunana kadar tüm kuyruk boyunca dolaşılması ve elemanın listeden çıkarılması ile gerçekleştirilir.
- Öncelik kuyruklarında listelerin sıralanarak kullanımı sıralanmadan kullanımına göre daha etkindir.

Dr. Sıgım TUNCEL

16

Diğer Bazı Liste Yapıları

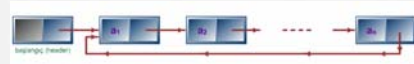
- Dairesel Bağlı Listeler - Circular Linked Lists
- Çift Bağlı Listeler - Doubly Linked Lists
- Dairesel Çift Bağlı - Listeler Circular Doubly Linked Lists

Dr. Sıgım TUNCEL

17

Dairesel Bağlı Listeler

Her bir düğüm kendinden sonraki düğümü son düğüm ise ilk elemanı gösterir. Böylece daireesel bir yapı oluşur.

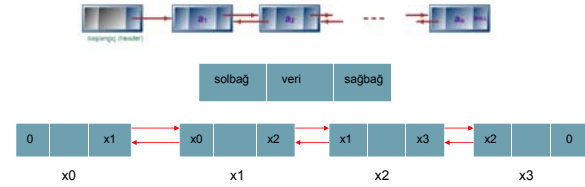


Dr. Sıgım TUNCEL

18

Çift Bağlı Listeler

Her düğümü iki bağ içerir. Bağlardan biri kendinden önceki, diğeri kendinden sonraki düğümü gösterir. Çift bağlı listelerde, tek bağlı listelerdeki geriye doğru listeleme ve dolaşımadaki zorluklar ortadan kalkar.



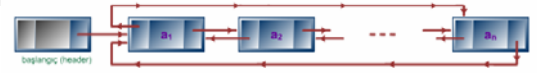
Dr. Savaş TUNÇEL

19

Dairesel Çift Bağlı Listeler

Hem dairesellik hem de çift bağlılık özelliklerine sahip listelerdir.

İlk düğümünden önceki düğüm son, son düğümünden sonraki düğüm de ilk düğümdür.



Dr. Savaş TUNÇEL

20

Tek yönlü sıralı bağlantılı listeye düğüm ekleme

'G' düğümünü eklemek istersek



x=5

Yeni düğümün listedeki yeri bulunur. Bunun için düğümün arasına gireceği düğümlerin adresi belirlenerek bunların adresleri i, j değişkenlerine atanır.

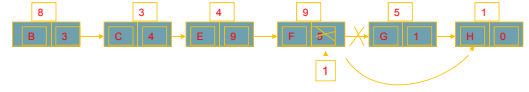


x adresli düğümün bağ alanına j düğümünün adresi yazılır. Bu şekilde yeni düğüm daha önce i düğümü tarafından işaretlenen j düğümünü gösterir. i, düğümünün bağ alanına da x düğümünün adresi yazılır ve yeni düğüm listeye eklenmiş olur.

Tek yönlü sıralı bağlantılı listeden düğüm çıkarma

'G' düğümünü çıkarmak istersek

- 'G' elemanından önce gelen 'F' elemanının bağ alanına çıkarılacak olan 'G' elemanının bağ alanındaki adresi yazmaktır.
- Bu durumda 'F' düğümü 'H' düğümünü göstereceği için 'G' düğümü listeden çıkarılmış olur.
- 'G' elemanından sonra gelen düğümler değişmediği için herhangi bir kaydırma işlemine gerek kalmaz.
- Bu işlem dizi kullanılarak yapılsa idi, 'G' den sonra gelen elemanların bir eleman sola çekilmesi gerekirdi.



Bağlı liste oluşturma

```
typedef struct Link_List {
    char isim;
    struct Link_List *next;
} Liste_Gir, *Listptr;
```

Yukardaki tip tanımlamasında iki yeni veri tipi göze çarpmaktadır. Tanımlama yapı(struct) tipinde Liste_Gir ve yine yapı tipi gibi bir işaretçi olan Listptr değişkenlerini içermektedir. Link_List yapı için kullanılan geçici bir isimdir ve bir sonraki alana geçiş sağlar. Bu yapı iki üyeden oluşmaktadır:

Liste_Gir.isim listede yer alacak isimler.

Liste_Gir.next bir sonraki kaydı gösteren pointer.

Bir bağlı liste oluşturmak için üç işaretçi gerekir. 1. işaretçi ilk kaydı, 2. işaretçi son kaydı ve 3. işaretçi o anda listeye girecek kaydı gösterir. Bu işaretçiler şu şekilde tanımlanabilir:

Listptr ilk, son, gir ;

Çalışma Sorusu

N adet öğrencinin numara ve harf ortalamalarını bağlı dizilerle gösteriniz.

- Herbir node öğrenci numarası, dersin kodu, dersin harf ortalaması bilgilerini bulunduracak.
- Herbir node bir sonraki öğrenciyi gösterecek.

Tamsayı elemanlardan oluşan bir bağlı liste sınıfına, elemanları küçükten büyüğe sıralı tutacak şekilde eleman ekleyen "ekle" metodunu yazınız.

Dr. Savaş TUNÇEL

24

