

Veri Yapıları ve Algoritmalar

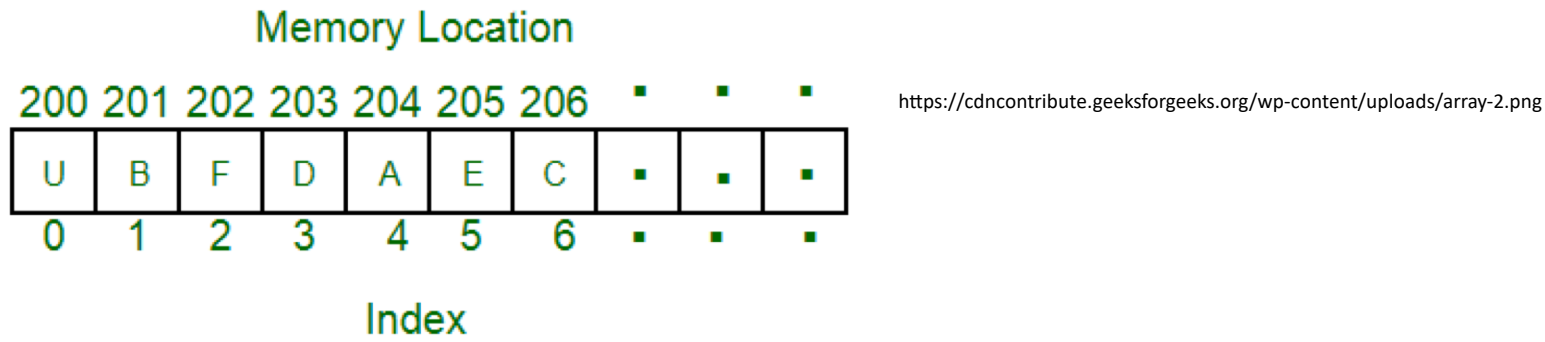
Listeler

Listeler

- Aralarında ilişki bulunan veri topluluğuna liste denir.
- Diziler listelere verilebilecek en basit veri türüdür.

Doğrusal Liste (Dizi)

- Bellekte peş peşe saklanırlar.
- Dizilerin boyutları program başında belirtilir ve bellekte yer ayrılır.
- Eleman sayısı değiştirilemez.
- Dizinin ilk elemanının bellek adresini bilirsek indislerle diğer elemanlara ulaşabiliriz.

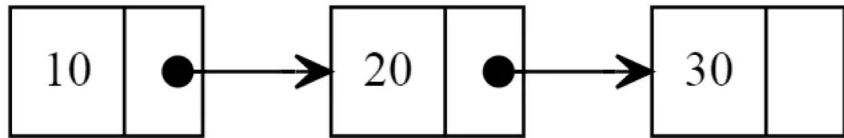


Doğrusal Liste (Dizi)

- Dezavantajlar
 - Yeni eleman eklemek istediğimizde dizinin boyutunu arttıramayız.
 - Dizinin boyutu gerektiğinden büyük tanımlanırsa fazladan bellek alanı işgal etmiş oluruz.
 - Sıralanması gerektiğinde ek çalışma yükü gerekir.

Bağlı Listeler

- Diziler gibi bellekte sıralı olarak yer almaları gerekmez.
- Dizilerin dezavantajları bağlı listelerle giderilebilir.
- Bağlı listede ki her eleman(**node**) hem veriyi, hem de kendisinden sonraki elemanın bellekteki adresini gösterir.



Linked List

<https://i1.wp.com/www.rubyguides.com/wp-content/uploads/2017/07/linked-list-with-label.png?ssl=1>

Bağlı Listeler

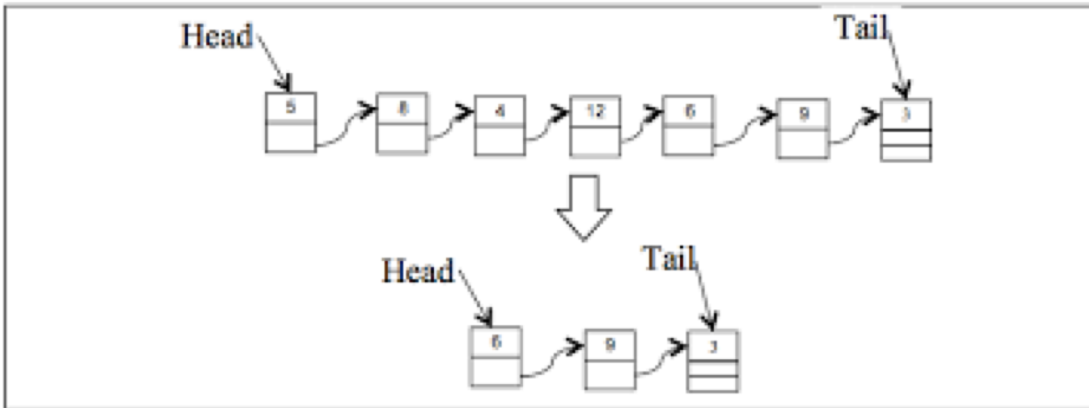
- Listedeki her bir eleman (node) iki kısımdan oluşur.
 - Data: Düğümde saklanan veri
 - Link: Bir sonraki düğümün adresi
 - Diğer? Ayrıca listenin ilk elemanını (**head**) ve son elemanını (**tail**) bilmemiz gerekiyor.



<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQfUkBufyFNv0LPQ4181E0Oq3mcaFX6Wa7CnTd3aNE-wY6ouTgMmA>

Bağlı Listeler

- Ayrıca listenin ilk elemanını (**head**) ve son elemanını (**tail**) bilmemiz gerekiyor.



<https://www.google.com.tr/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiS3qybvLbeAhVC-6QKHQGaCK8QjRx6BAgBEAU&url=https%3A%2F%2Fstackoverflow.com%2Fquestions%2F12760270%2Fwhen-a-single-linked-list-node-becomes-a-garbage-data&psig=AOvVaw2YkQDgR6cUvVZsZK9TZY26&ust=1541274131344634>

Bağlı Listelerde İşlemler

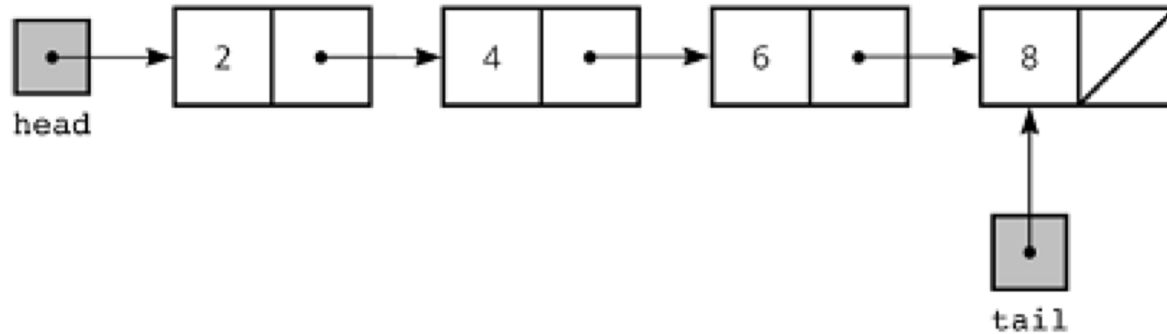
- Listeye eleman ekleme
 - Başa
 - Sona
 - Araya
- Listeden eleman silme
 - Baştan
 - Sondan
 - Aradan
 - Belirli kritere göre
- Arama
- Listeleme

Bağlı Liste Türleri

- Tek yönlü doğrusal bağlı liste
- İki yönlü doğrusal bağlı liste
- Tek yönlü dairesel bağlı liste
- ki yönlü dairesel bağlı liste

1. Tek Yönlü Bağlı Listeler

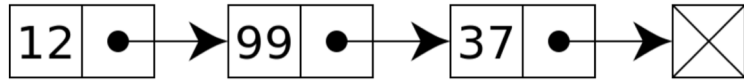
- Hareket yönü listenin başından sonuna doğrudur.



https://s3-us-west-2.amazonaws.com/ib-assessment-tests/problem_images/singly-ll.png

1. Tek Yönlü Bağlı Liste

- Örnek Liste



<https://medium.com/@kojinoshiba/data-structures-in-python-series-1-linked-lists-d9f848537b4d>

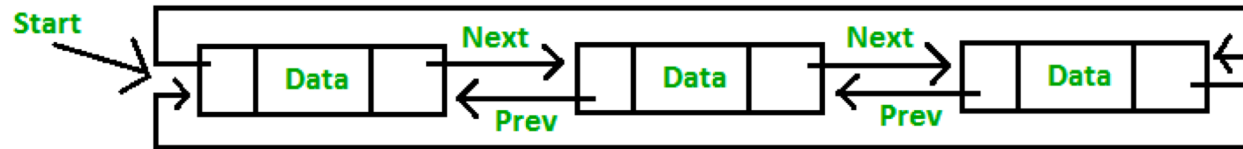
- Düğüm Python Kodu

```
class Node:
    def __init__(self, val):
        self.val = val
        self.next = None
```

```
node1 = Node(12)
node2 = Node(99)
node3 = Node(37)
node1.next = node2 # 12->99
node2.next = node3 # 99->37
```

2. İki Yönlü Bağlı Listeler

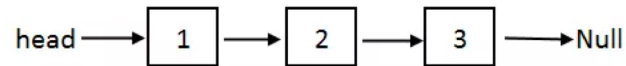
- Elemanlar arasında iki yönlü bağ vardır. Her düğüm hem kendinden önceki, hem kendinden sonraki düğümlerin adresini tutar.
- Böylece listede hem ileri doğru, hem geri doğru dolaşılabilir



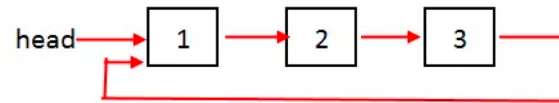
<http://cdncontribute.geeksforgeeks.org/wp-content/uploads/Circular-doubly-linked-list.png>

3. Tek Yönlü Dairesel Bağlı Listeler

- Listedeki elemanlar arasında yine tek yönlü bağ vardır. Doğrusal listeden farkı ise son elemanın, ilk elemanı göstermesidir.



Singly Linked List

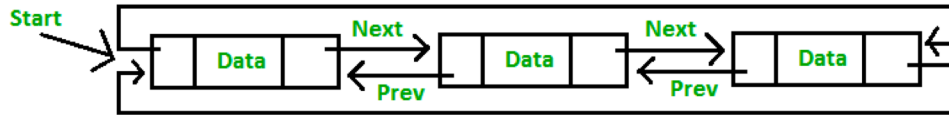


Circular Linked List

<https://i2.wp.com/algorithms.tutorialhorizon.com/files/2016/03/Circular-Linked-List.png>

4. Çift Yönlü Dairesel Bağlı Listeler

- Hem dairesel, hem çift yönlü doğrusal liste özelliklerine sahiptir. İlk eleman kendinden önce son elemanı, son eleman kendinden sonra ilk elemanı gösterir.



<http://cdncontribute.geeksforgeeks.org/wp-content/uploads/Circular-doubly-linked-list.png>

Ödev

- Aşağıdaki işlevlere sahip bir bağlı liste oluşturunuz
 - Başa eleman ekleme
 - Sona eleman ekleme
 - Araya eleman ekleme
 - Baştan eleman silme
 - Sondan eleman silme
 - Aradan eleman silme
 - Değere göre eleman arama
 - Tüm listeyi yazdırma