Arama Algoritmaları

Yunus Santur , ysantur@gmail.com

Arama Algoritmaları

- Lineer search (Doğrusal arama)
- Binary search (İkili arama)
- Jump search (Sıçramalı arama)
- Hash (Çırpı arama)

Arama algoritmaları

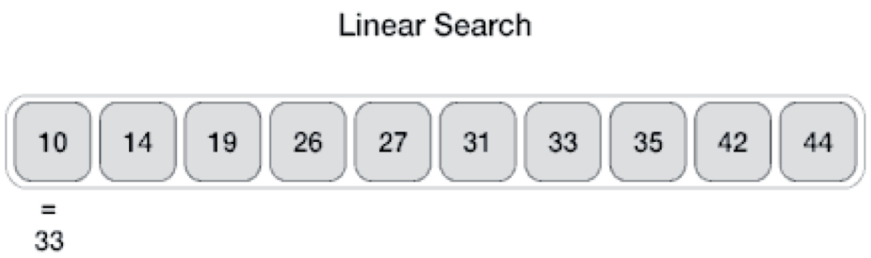
- Arama algoritmaları aranacak dizinin sıralı olup olmamasına göre değişmektedir.

- Dizi sırasız ise
  - Lineer search

- Dizi sıralı ise
  - Binary search
  - Jump search

Lineer search

- Tek bir döngüde arama yapılır, aranan bulunur ise döndürülür.



https://www.tutorialspoint.com/data_structures_algorithms/images/linear_search.gif

Lineer search

- Python kodu
  - Aranan bulunursa indisi ve elemanı, bulunamazsa False dönder.

```
def lineersearch(list, aranan):
    for i in range(0, len(list)):
        if list[i]==aranan:
            return (i,list[i])
    else:
        return False


list = [19,2,31,45,6,11,121,27]
print(lineersearch(list, 121))
```

Lineer search

- Kodlaması ve analizi basittir.
  - N eleman için
  - En iyi durum O(1)
  - En kötü durum O(n)
  - Ortalama durum O(n/2)
  - O(n)

Binary search

- **Sıralı dizi üzerinde çalışabilir!**
- Algoritma rekürsif çalışmaya uygundur.
- Ortanca elemanı bul
  - Aranan orta eleman ise döndür.
  - Aranan ortadan daha büyük ise dizinin sağ tarafı için yeniden arama yap.
  - Aranan ortadan daha küçük ise dizinin sol tarafı için yeniden arama yap.

Search for 47

0	4	7	10	14	23	45	47	53

https://ds055uzetaobb.cloudfront.net/image_optimizer/717403b1368376cb6f915e6b4beeb3a7ad54105e.gif

Binary search

- Python kodu:
  - Dizinin sıralı olduğu ve aranan elemanın dizide olduğu varsayılmıştır.
  - Rekürsif kodlanmıştır.

```
def simplebinarysearch(list,aranan):
        indis=len(list)//2
        if aranan==list[indis]:
            return (list[indis])
        elif aranan>list[indis]:
            list=list[indis:]
        else:
            list=list[:indis]

        return simplebinarysearch(list,aranan)
```

Ödev

- Verilen simplebinarysearch programında aranan elemanın dizide olduğu varsayılmıştır. Programı aşağıdaki gibi güncelleyiniz

  - Dizide olmayan eleman için False değeri döndürülmeli.
  - Rekürsif olarak kodlanmalı.
  - Aranacak elemanın kaç adımda bulunduğu döndürülmeli.

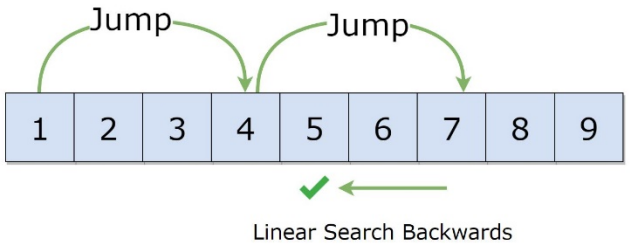
Binary search

- Rekürsif kodlamaya uygundur.
- N eleman için
  - En iyi durum O(1)
  - O(log_2^n)

1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9

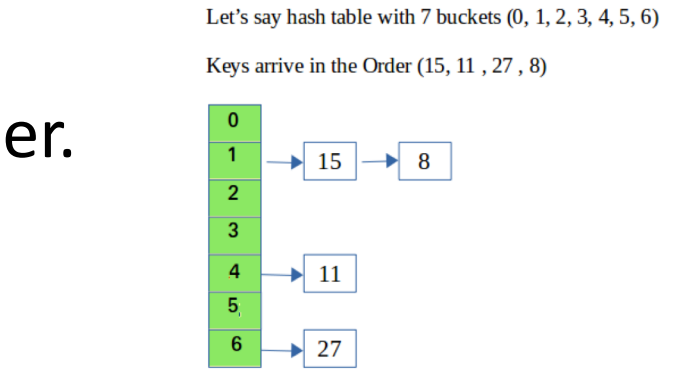
Jump Search

- Binary search gibi sıralı diziler üzerinde çalışabilir.
- Arama işlemi belirlenen adım sayısı kadar sıçrama yapılarak gerçekleştirilir.
- Aranan eleman daha küçük ise adım sayısı 1 eksiltilerek geri dönülür ve lineer search gerçekleştirilir.
- Adım sayısı genel olarak dizi uzunluğunun kare kökü seçilir.



Hashing/Hash Table

- Arama işleminin bir çırpıda yapılabilmesi işlemidir.
- Aranılan değerlerin kolay bulunabilmesini sağlayan bir veri yapısıdır O(1)
- Hash bir öğeyi hangi slot'a koyulacağını belirler.
- Dezavantajları
  - Collision



- Örnekte 7 elemanlı bir hash table kullanılıyor [15,11,27,8] dizisi hash fonksiyonundan geçirilerek (mod 7) tabloya bağlı liste olarak yerleştiriliyor.

Tartışma

- Metin içinde sözcük arama algoritması öneriniz.
- Araştırma konusu:
  - Arama motorları metin indeksleme
  - Elastic search vb