

# **Bilgisayar Mühendisliği**

## **Bilgisayar Mühendisliğine Giriş 101**

### **HW #09**

#### **1.Skip-list**

Skip-list kısacası hızlı arama yapmayı sağlayan veri yapılarıdır.Mantığı aslında aradığımız sayıya ulaşmadan önce yapacağımız karşılaştırma sayısı en aza indirmeye çalışmaktır.Diyelim ki uzun bir linked-list'e yani ilk elemanın adresini bildiğimizde sırayla tüm elemanlara küçükten büyüğe doğru erişebildiğimiz bir listeye sahibiz.Eğer ulaşmak istediğimiz eleman listenin sonunda ise sona gelene kadar her eleman için karşılaştırma yapmamız gerekir ki bu çok zaman alır.Bu durum aynı zamanda en kötü ihtimal (worst case) dediğimiz olası bir durumdur.İşte skip-list veri yapıları da bu listenin en kaba halinden farklı olarak bu listeye bir seviye daha ekleyerek karşılaştırma sayısını azaltmayı hedef alır.Örnek ile açıklayacak olursak:

Diyelim ki listemiz şu şekilde olsun:

4,5,9,11,17,23,30,34,46,52,60,75,195,295,300

şeklinde küçükten büyüğe sıralanmış listede arama yaparken, "4" elemanını bulmak çok hızlı olurken,"195" elemanını bulmak için çok fazla eleman ile karşılaştırma yapmak gerekicek ki bu zaman alır.Ama bir ikinci seviye tanımlarsak skip-list veri yapısına geçeriz.O da şöyle olur:

Seviye 2 – 4, 17, 34, 75, 295  
Seviye 1 – 4,5,9,11,17,23,30,34,46,52,60,75,195,295,300

"300" elemanına ulaşmak istediğimde ; önce "4" ile karşılaştırma olur.  $4 < 195$  olduğundan "17" ile karşılaştırma olur. $17 < 195$  şeklinde Seviye 2'de hareket ederiz. $75 < 195$  ve en son  $195 < 295$  olduğundan Seviye-2'de 75'den Seviye-1'e inirim ve sonuç olarak karşılaştırma sayısını azaltıp arama işlemine hız kazandırmış olurum.

Bu noktada bildiğimiz linked-list yapısı bir eleman bir sonraki elemana point ederken skip-list yapısında Seviye-2'deki eleman Seviye -1'deki karşılığına point eder.Bu şekilde seviyeler arasında yönlendirmeler sağlanmış olur.Avantajları da açıklandığı gibi listeler içinde arama yaparken hız kazanmak, aramalar için process sayısını azaltmış olmak şeklinde sıralanabilir.

List.element(a):listenin a indexli elemanı.  
List.add(a):listenin sonuna a elemanını ekle  
List.size = listenin uzunluğunu verir.  
For c in interval[a,b)= verilen matematiksel aralığın içinde dönen c değişkeni.

Algoritma açıklaması:

Fonksiyon , liste ve aranan eleman argümanlarını alır.Result adındaki değişkene 0 başlangıç değerini atadıktan sonra listenin uzunluğu kadar i'yi 0 dan itibaren döngüye sokar ve listenin “i” indexindeki elemanın her key’e eşit olma durumunda result’u 1 artırır.Döngü bitince resultu return eder.

```
def countInList(myList list,int key):  
    int result = 0  
    for i in interval[0,list.size):  
        if(list.element(i)==key):  
            result= result + 1  
  
    return result                                     //for döngüsünden sonra result'u return et.
```

Algoritma açıklaması:

Fonksiyon liste argümanı alır ve yeni bir liste oluşturur.Yine liste uzunluğu kadar for döngüsünde yeni listeye her seferde kendisinde bulunan 1 elemanı ekler.Ve hemen bu işlemin ardından result değişkenine yeni listemizde her “i” elemanı için countInList fonskiyonunu atar.Yani liste güncellenir güncellenmez yeni listeye son eklenen eleman listede kaç defa geçiyor diye kontrol eder.Her yeni eklemekten sonra son eklenen eleman 2 defa geçiyorsa o ilk kez tekrarlayan elemandır.Yani result 2 olduğunda if’e girerek hemen return eder.

```
def firstRepeating(myList list):  
    Mylist newList=[ ]  
    result = 0  
    for i in interval[0,list.size):  
        newList.add(list.element(i))  
        result = countInList(newList,newList.element(i))  
  
    if(result==2):  
        return newList.element(i)    // if’e girerse return et.
```

Algoritma açıklaması :

Bir for döngüsü ama yeni liste için artan bir counter eski liste(reverse'ni istediğimiz liste) için azalan bir counter tanımlı.Yani yeni listenin ilk indexinden başlayıp artacak şekilde bunun karşılığında ise eski listenin son indexinden başlayıp azalacak şekilde elemanları assign eder.Bu sayede tersi alınmış olur.

```
def reverse(myList list,myList newList):
    myList newList=[ ]
    counter = list.size-1                                //1.eleman 0.indexte olduğundan.
    for i in interval[0,list.size):
        newList.add(list.element(counter))
        counter=counter -1

    return newList                                       //for döngüsünden sonra yeni listeyi return et.
```