

# BAĞLI LİSTE İLE KELİME SAYMA

Metehan MART, M. Gürkan CİNGÖZ

Bilgisayar Mühendisliği Bölümü  
Kocaeli Üniversitesi  
metehanmart@gmail.com, gurkancingoz@yandex.com

## 1. PROBLEM TANIMI

Txt'den alınan paragrafta geçen kelime sayısını bulup büyükten küçüğe bağlı liste kullanarak ve bizden kullanmamız istenilen başa ekle, araya ekle, sona ekle fonksiyonlarını kullanarak bağlı listeye ekleyip listelemek. Bunu yaparken izlenmesi gereken algoritma önce kelimeyi bulup sonra adedini bulup bağlı listeye eklemektir.

## 2. YAPILAN ARAŞTIRMALAR

Proje yapılırken birçok problemle karşılaşıldı. Dosyadan paragrafı projenin istediği gibi alma konusunda sıkıntı yaşadık bunu internette araştırıp benzer bir kodu kendi kodumuza uyarladık. Listelemeyi başardığımızda kelimelerin sonlarında değişik semboller çıkıyordu bu problemi maindeki memmove fonksiyonlarının parametrelerini değiştirerek ve kelime sonlarına '\0' ifadesini yerleştirerek çözdük. İlk düğüm ve son düğüm yazılırken sıkıntı çıkıyordu. İlk düğüm tamamen anlamsız semboller olarak son düğümün son karakteri eksik olarak çıkıyordu. Son düğümdeki sıkıntı txtdeki son karakterden kaynaklanıyordu. Onun için ayrı bir şart koyduğumuzda problemi çözdük. İlk düğümdeki sıkıntı ekleme fonksiyonlarından kaynaklandığını tespit ettikten ve ilk düğümün ikinci düğüm eklendikten sonra null olduğunu gördük ve bunu düzelttik. Kodumuzda büyük bir tutarsızlık problemi vardı. Aynı kodu başka bir projede çalıştırdığımızda ya da birbirimize yolladığımızda hata alıyorduk. Belli değerleri değiştirince birimizin bilgisayarında çalışırken aynı işlemi diğer bilgisayarda yapınca çalışmıyordu. Hatanın ilk başta derleyiciden kaynaklı olduğunu düşünüp 32 bitlik derleyici ile denediğimizde de anı sonuçla karşılaştık. Sonra hatanın SonaEkle fonksiyonunda olduğunu gördük yeni eklenen düğümün nexti null olmadığından sürekli hata alıyorduk bunu ekledikten sonra kod stabil hale geldi.

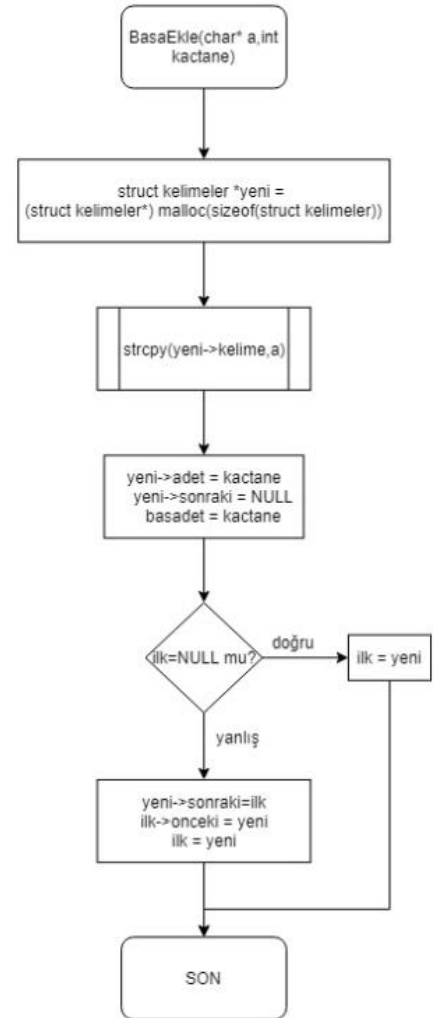
```
yeni->sonraki = NULL; //tutarlılık sağlandı
```

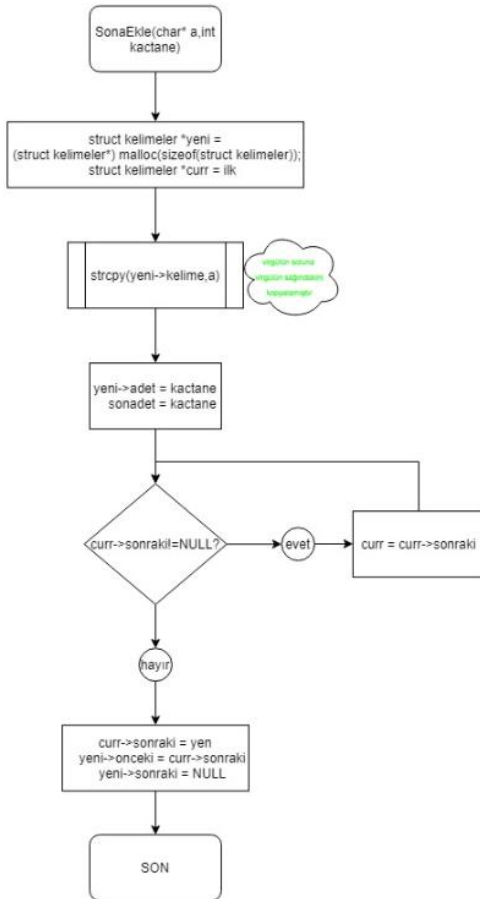
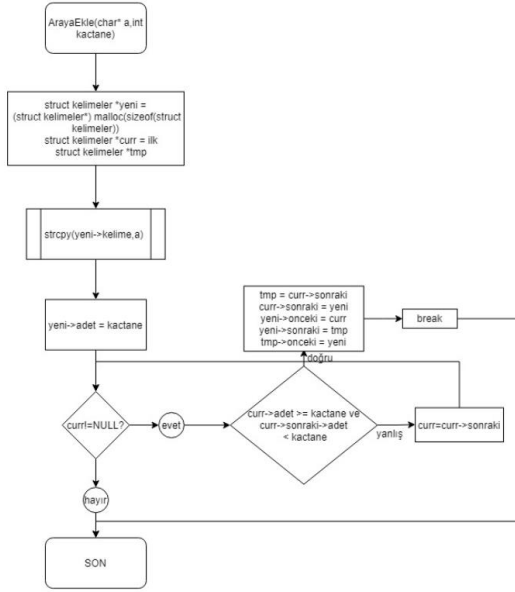
Bunu ekledikten sonra ilk ve son düğüm tekrar bozulmasına rağmen artık kod tutarlı davranıyordu. Farklı projede de farklı bilgisayarda da aynı hatayı veriyordu. Kalan iki hatayı çözmek zor olmadı. Son düğümdeki hatanın sebebi bizim onun için tasarladığımız else ife girmemesiydi. Satır sayısını doğru hesaplamaya başladığımızda son düğüm

düzeldi. İlk düğümdeki sıkıntı düğümleri listeleyen Listele fonksiyonu çalışmadan önce ilk düğümün freelenmesiydi. Ufak bir değişikle sorunu çözdük. Kodumuz artık istenilen gibi çalışıyordu.

## 3. TASARIM

### 3.1. AKIŞ ŞEMASI





[Diyagramın tamamı](#)

### 3.2. YAZILIM MİMARİSİ

Struct yapımız şekildeki gibidir:

```

struct kelimeler
{
    char kelime[255];
    int adet;
    struct kelimeler *sonraki;
    struct kelimeler *onceki;
};
  
```

Kelime değişkeni tutulacak kelime için adet kelime adedi içindir. Sonraki ve önceki bağlı listemizin çift yönlü olduğunu gösteriyor. Çift yönlü olması bize basa ekleme yaparken kolaylık sağlıyor. Aşağıda görüldüğü gibi sadece ilk düğümü başlattık.

```

struct kelimeler *ilk;
  
```

Son düğüm içinde değişken düşünüyorduk ancak bu projeyi gereksiz yere karmaşıktırdığından bu fikrimizden vazgeçtik. Ekleme fonksiyonlarımız şu şekildedir:

```

void BasaEkle(char* a, int kactane)
{
    struct kelimeler *yeni = (struct kelimeler*) malloc(sizeof(struct kelimeler));
    strcpy(yeni->kelime, a);
    //yeni->kelime[strlen(a)] = '\0';
    // yeni->sonraki = NULL;
    // yeni->onceki = NULL;
    yeni->adet = kactane;
    yeni->sonraki = NULL;
    basadet = kactane;
    if(ilk==NULL)
    {
        ilk = yeni;
        //printf("%s %d \n", ilk->kelime, ilk->adet);
    }
    else
    {
        //printf("ilkkin durumu: %s : %d\n", ilk->kelime, ilk->adet);
        yeni->sonraki=ilk;
        ilk->onceki = yeni;
        ilk = yeni;
    }
    //free(yeni);
}

void ArayaEkle(char* a, int kactane)
{
    struct kelimeler *yeni = (struct kelimeler*) malloc(sizeof(struct kelimeler));
    struct kelimeler *curr = ilk;
    struct kelimeler *tmp;
    strcpy(yeni->kelime, a);
    yeni->adet = kactane;
    while(curr!=NULL){
        if(curr->adet >= kactane && curr->sonraki->adet < kactane){
            tmp = curr->sonraki;
            curr->sonraki = yeni;
            yeni->onceki = curr;
            yeni->sonraki = tmp;
            tmp->onceki = yeni;
            //printf("ilkkin durumu: %s : %d\n", ilk->kelime, ilk->adet);
            break;
        }
        else{
            curr=curr->sonraki;
        }
    }
    //free(yeni);
    //free(curr);
}
  
```

```

void SonaEkle(char* a,int kactane)
{
    struct kelimeler *yeni = (struct kelimeler*) malloc(sizeof(struct kelimeler));
    struct kelimeler *curr = ilk;
    strcpy(yeni->kelime,a);
    yeni->adet = kactane;
    sonadet = kactane;
    while(curr->sonraki!=NULL){
        curr = curr->sonraki;
    }
    //printf("ilkkin durumu sonaekle calismadan once: %s : %d\n",ilk->kelime,ilk->adet);
    curr->sonraki = yeni;
    yeni->onceki = curr->sonraki;
    yeni->sonraki = NULL;//tutarlilik saglandi
    //printf("ilkkin durumu sonaekle calistiktan sonra: %s : %d\n",ilk->kelime,ilk->adet);
    //free(yeni);
    //free(curr);
}

```

#### 4. GENEL YAPI

Projemiz maindeki ikinci while'a girdikten sonra txt de verilen paragraf str değişkeninde tutulur. Bu str paragrafın bir satırıdır. For döngüsü bu satırda kelime arar. Burada kelimedenden kastedilen iki boşluk arasındaki karakterlerdir. Bu karakterleri bulduğunda bunu tmp adındaki geçici bir değişkene atar. Bu değişken Arama fonksiyonunda ilk önce büyük harfleri küçük harfe dönüştürülür eğer satirin sonundaysa '\n' karakteri silinir. Sonra bağlı listede zaten aynı kelime var mı bakılır eğer bu kelime varsa fonksiyon returnlenip tekrardan aynı kelimenin eklenmesinin önüne geçilir. Eğer bu kelime bağlı listede yoksa maindeki while ve for döngüsündeki yapılar benzer bir yapı kullanılarak paragraftaki adedi bulunur sonra ekleme fonksiyonuna adedi ile birlikte gider. Ekleme fonksiyonunu bir dağıtım merkezine benzetebiliriz.

```

void ekleme(char* a,int kactane)
{
    if(ilk == NULL)
    {
        //printf("basa ekleye giriyor nullken\n");
        BasaEkle(a,kactane);
        return;
    }
    else if(kactane > basadet){
        //printf("basa ekleye giriyor\n");
        BasaEkle(a,kactane);
    }
    else if(kactane <= sonadet || sonadet == -1){
        //printf("sona ekleye giriyor\n");
        SonaEkle(a,kactane);
    }
    else if(kactane <= basadet && kactane > sonadet){
        //printf("araya ekleye giriyor\n");
        ArayaEkle(a,kactane);
    }
}

```

Eklenecek düğümün başa mı araya mı yoksa sona mı ekleneceğine karar verir. Basa ekle için iki olasılıktan biri ilk düğümün null olması ikincisi ilk düğümün null olmamasıdır. Hangi olasılık doğru ise o çalışacaktır. ArayaEkle fonksiyon ekleme yapacağı yere kadar ilk düğümünden ilerler. Sonra eklemesini yapar. SonaEkle ArayaEkledekine benzer bir sistemle düğümün sonuna gelir ve

eklemesini yapar. Listele fonksiyonu ilk düğümünden başlar ve while döngüsü sağlanmayana kadar döner.

#### 5. DENEYSEL SONUÇLAR

okut.txt - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

\*\*\*\* En sıcak kasım rekoru daha önce 2016 ve 2019 yıllarında degismisti . Kuresel isinmanin etkileri her gecen yil daha net goruluyor . En sicak kasim rekoru daha önce 2016 ve 2019 yıllarında degismisti . 2019'un ardindan 2020 yilinin kasim ayinda rekor kirilmis oldu . En sicak kasim rekoru daha önce 2016 ve 2019 yıllarında degismisti . Detaylar birazdan ntv.com.tr'de ...

"C:\Users\Metehan\Documents\Mete Programlama\C\ProLab3\bin\

```

1. *: 5
2. .: 5
3. kasim: 4
4. daha: 4
5. en: 3
6. sicak: 3
7. rekoru: 3
8. once: 3
9. 2016: 3
10. ve: 3
11. 2019: 3
12. yillarinda: 3
13. degismisti: 3
14. kuresel: 1
15. isinmanin: 1
16. etkileri: 1
17. her: 1
18. gecen: 1
19. yil: 1
20. net: 1
21. goruluyor: 1
22. 2019'un: 1
23. ardindan: 1
24. 2020: 1
25. yilinin: 1
26. ayinda: 1
27. rekor: 1
28. kirilmis: 1
29. oldu: 1
30. detaylar: 1
31. birazdan: 1
32. ntv.com.tr'de: 1
33. ...: 1

```

Process returned 0 (0x0) execution time : 0.087 s  
Press any key to continue.

oku.txt - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

It secured supplies of the Pfizer - BioNTech vaccine following negotiations early on in the pandemic . It is contacting people with priority access to the vaccine through its health care system - by law all Israelis must register with a recognised health care provider .

```
"C:\Users\Metehan\Documents\Mete Programlama\C\ProLab3\bin
1. the: 3
2. it: 2
3. -: 2
4. vaccine: 2
5. .: 2
6. with: 2
7. health: 2
8. care: 2
9. secured: 1
10. supplies: 1
11. of: 1
12. pfizer: 1
13. biontech: 1
14. following: 1
15. negotiations: 1
16. early: 1
17. on: 1
18. in: 1
19. pandemic: 1
20. is: 1
21. contacting: 1
22. people: 1
23. priority: 1
24. access: 1
25. to: 1
26. through: 1
27. its: 1
28. system: 1
29. by: 1
30. law: 1
31. all: 1
32. israelis: 1
33. must: 1
34. register: 1
35. a: 1
36. recognised: 1
37. provider: 1

Process returned 0 (0x0)   execution time : 0.088 s
Press any key to continue.
```

## 6. KAYNAKÇA

<https://www.geeksforgeeks.org/linked-list-set-2-inserting-a-node/> (Erişim Tarihi: 12.12.2020)

<https://stackoverflow.com/questions/41902471/how-to-use-fgets-to-read-a-file-line-by-line> (Erişim Tarihi: 20.12.2020)

Onur Gök 25 Kasım 2020 Veri Yapıları ve Algoritmaları Dersi