



KUYRUK YAPILARI

Kuyruk Yapıları

Kuyruk veri yapıları yığın veri yapılarına benzemektedir ve yığında yapılan işlemler kuyruk veri yapıları içinde geçerlidir. Eleman ekleme yığınlarda olduğu gibi verinin sonuna eklenmektedir. Fakat eleman silme işlemi yığınlardan farklıdır. **Yani ilk giren eleman ilk çıkarılmaktadır.** (First in first out)



27.02.2017

Dr.Eyyüp GÜLBANDILAR

2

Doğrusal kuyruk veri yapıları

Verilerin dizi yada listede şekildeki gibi ardışık olarak yerleştirilmesine doğrusal kuyruk denir. Dizilerdeki doğrusal liste de veriler yerleştirilirse **gereksiz bellek kullanımına neden** olur. Çünkü silme liste başından başlar ve boşaltılan yerler tekrar doldurulama ve bellek bloke olmuş olur.



27.02.2017

Dr.Eyyüp GÜLBANDILAR

3

Doğrusal Kuyruk Yapıları

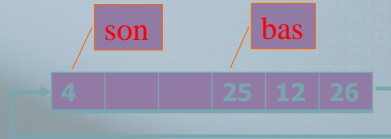
Kuyruğa eleman ekleme ve eleman çıkarma işlemi belirli kısıtlar dahilinde yapıldığı için doğrusal kuyrukta **ortalardaki bir elemana** ulaşmak için onun önünde bulunan tüm elemanların kuyruktan çıkarılması gerekir. Bu durum yığın yapısına benzediği için kuyruk üzerinde yapılacak işlemleri sınırlar.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

4

Dairesel kuyruk veri yapıları



Dairesel kuyruk fiziksel olarak yer bulunduğu sürece doğrusal kuyruğun eleman eklemadaki problemleri çözer. Eğer dairesel kuyruk fiziksel olarak dolu ise o zaman eleman eklemek mümkün olmaz.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

5

Dairesel Kuyruk Yapısı

Dairesel kuyruk yapısı ile bellek daha etkin kullanılır. Dairesel kuyruğa eleman ekleme ve dairesel kuyruktan eleman çıkarma da algoritma aracılığı ile yapılmaktadır. Dairesel kuyruk yapısında elemanlar bir dizide olduğu gibi art arda gelmektedirler ve kuyruğun en son elemanı kuyruğun birinci elemanını işaret etmektedir.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

6

Sabit Dizilerde Kuyruk Yapıları

Kuyruk tanımlanırken **tipi ve eleman kapasitesi** tanımlanmak zorundadır.

Tip Kuyruk[1:n]

olduğu gibidir.

Ayrıca;

KuyrukBaşı(bas) ← KuyrukSonu(son) ← 0

gibi indis **değişkenlerine** sıfır atanarak ilk değer ataması yapılır.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

7

Sabit Dizilerde Kuyruk Yapıları

Kuyruğun başlangıcı her zaman ilk eklendiğinde sıfır olmak zorunda değildir.



Kuyruk **bas** ve **son** indis değişkenleri dizide aynı **değeri** gösteriyorsa **doludur**.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

8

Sabit Dizilerde Kuyruk Yapıları

C dili için;

```
struct kuyruk {
    Ornekptr* dizi;
    int N, bas, son;
};
typedef struct kuyruk Kuyruk;
typedef Kuyruk* Kuyrukptr;
Kuyrukptr yeni_kuyruk(int N){
    Kuyrukptr kuyruk;
    kuyruk = malloc(sizeof(Kuyruk));
    kuyruk->dizi=malloc(N*SOPTR);
    kuyruk->bas=0;
    kuyruk->son=0;
    kuyruk->N=N;
    return kuyruk;
}
```

```
int kuyruk_dolu(Kuyrukptr k){
    if (k->bas==(k->son+1)%k->N)
        return 1;
    else
        return 0;
}
int kuyruk_bos(Kuyrukptr k){
    if (k->bas==k->son)
        return 1;
    else
        return 0;
}
```

Kuyruk dolu mu?

Kuyruk boş mu?

27.02.2017

Dr.Eyyüp GÜLBANDILAR

9

Sabit Dizilerde Kuyruk Yapıları

Java dili için;

```
Public Class Kuyruk {
    Ornekdizi[];
    int N, bas, son;
    public Kuyruk(int N){
        dizi=new Ornek[];
        this.N=N;
        bas=0;
        son=0;
    }
}
```

```
boolean kuyrukDolu(){
    if (bas==(son+1)%N)
        return true;
    else
        return false;
}
boolean kuyrukBos(){
    if (bas==son)
        return true;
    else
        return false;
}
```

Kuyruk dolu mu?

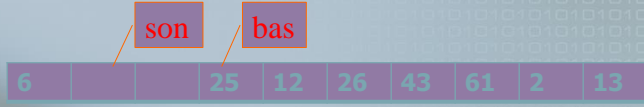
Kuyruk boş mu?

27.02.2017

Dr.Eyyüp GÜLBANDILAR

10

Sabit dizilerde kuyruğa eleman ekleme



```

void kuyruğa_ekle(Kuyrukptr k, Ornekptr yeni){
    if (!kuyruk_dolu(k)){
        k->dizi[k->son]=yeni;
        k->son=(k->son+1)%k->N;
    }
}
  
```

C dili için;

Maliyet O(1)

```

void kuyruğaEkle(Ornek yeni){
    if (!kuyrukDolu()){
        dizi[son]=yeni;
        son=(son+1)%N;
    }
}
  
```

Java dili için;

27.02.2017 Dr.Eyyüp GÜLBANDILAR 11

Sabit dizilerde kuyruğa eleman ekleme

- Öncelikle kuyruğun **dolu olup** olmadığı kontrol edilmelidir.
- Kuyruk boş ise «**yeni**» eklenir
- **Son** indis değışkeni bir artırılır.
 - Eğer eklenen dizinin N elemanı ise **son** indisi dizinin ilk elemanına arttırılır.

27.02.2017 Dr.Eyyüp GÜLBANDILAR 12

Sabit dizilerde kuyruktan eleman silme



C dili için;

```
Ornekptr kuyruk_sil(Kuyrukptr k){
    Ornekptr sonuc;
    if (!kuyruk_bos(k)){
        sonuc=k->dizi[k->bas];
        k->bas=(k->bas+1)%k->N;
        return sonuc;
    }
    return NULL;
}
```

Java dili için;

```
Ornek kuyrukSil(){
    Ornek sonuc;
    if (!kuyrukBos()){
        sonuc=dizi[bas];
        bas=(bas+1)%k->N;
        return sonuc;
    }
    return NULL;
}
```

Maliyet $O(1)$

27.02.2017 Dr.Eyyüp GÜLBANDILAR 13

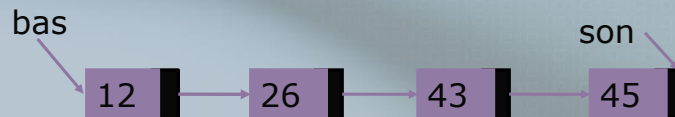
Sabit dizilerde kuyruktan eleman silme

- Öncelikle kuyruğun **boş olup** olmadığı kontrol edilmelidir.
- Kuyruk dolu ise «**sonuc**» silinir.
- **bas** indis değişkeni bir artırılır.
 - Eğer silinen dizinin N elemanı ise **bas** indisi dizinin ilk elemanına arttırılır.

27.02.2017 Dr.Eyyüp GÜLBANDILAR 14

Bağlantı liste ile kuyruk yapıları

Bağlantı listede eğer eleman yok ise **bas** ve **son** indis değişkenleri **NULL** karşılık gelir.



27.02.2017

Dr.Eyyüp GÜLBANDILAR

15

Bağlantı liste ile kuyruk yapıları

C dili için;

```

struct kuyruk{
    Elemanptr bas;
    Elemanptr son;
}
typedef struct kuyruk Kuyruk;
typedef Kuyruk* Kuyrukptr;
Kuyrukptr yeni_kuyruk(){
    Kuyrukptr kuyruk;
    kuyruk=malloc(sizeof(Kuyruk));
    kuyruk->bas=NULL;
    kuyruk->son=NULL;
}
int kuyruk_bos(Kuyrukptr k){
    if (k->bas==NULL)
        return 1;
    else
        return 0;
}

```

Maliyet O(1)

Dr.Eyyüp GÜLBANDILAR

27.02.2017

Java dili için;

```

public class kuyruk{
    Eleman bas;
    Eleman son;
    public Kuyruk(){
        bas=NULL;
        son=NULL;
    }
    boolean kuyrukBos(){
        if (bas==NULL)
            return true;
        else
            return false;
    }
}

```

16

Bağlantı listeli kuyruğa eleman ekleme



```

graph LR
    bas --> 12
    12 --> 26
    26 --> 43
    43 --> 45
    45 --> 06
    son --> 45
    son2[son] --> 06
  
```

C dili için;

```

void kuyruğa_ekle(Kuyrukptr k, Elemanptr yeni){
    if (!kuyruk_bos(k))
        k->son->ileri=yeni;
    else
        k->bas=yeni;
        k->son=yeni;
}
  
```

Java dili için;

```

void kuyruğaEkle(Eleman yeni){
    if (!kuyrukBos())
        son.ileri=yeni;
    else
        bas=yeni;
        son=yeni;
}
  
```

Maliyet $O(1)$

27.02.2017 Dr.Eyyüp GÜLBANDILAR 17

Bağlantı listeli kuyruğa eleman ekleme

- Öncelikle kuyruğun **boş** olup olmadığına bakılır. (Listede doluluk ihtimali yoktur)
- Eğer boş ise **bas** ve **son** indisler yeni ekleneni gösterir.
- Aksi durumda **son** indisi yeni ekleneni gösterir.

27.02.2017 Dr.Eyyüp GÜLBANDILAR 18

Bağlantı listeli kuyruktan eleman silme



C dili için;

```

Elemantptr kuyruk_sil(Kuyrukptr k){
    Elemantptr sonuc;
    sonuc=k->bas;
    if (!kuyruk_bos(k)){
        k->bas=k->bas->ileri;
        if (k->bas=NULL)
            k->son=NULL
    }
    return sonuc;
}

```

Java dili için;

```

Eleman kuyrukSil(){
    Eleman sonuc;
    sonuc=bas;
    if (!kuyrukBos()){
        bas=bas.ileri;
        if (bas=NULL)
            son=NULL
    }
    return sonuc;
}

```

Maliyet $O(1)$

27.02.2017 Dr.Eyyüp GÜLBANDILAR 19

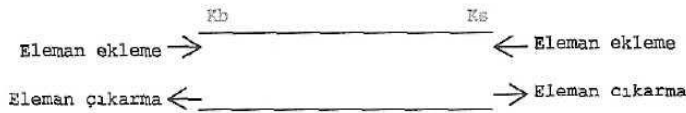
Bağlantı listeli kuyrukta eleman silme

- Öncelikle kuyruğun **boş** olup olmadığına bakılır.
- Eğer boş ise **bas** ve **son** indisler **NULL** gösterir.
- Eğer bir eleman varsa **bas** **NULL** olur ve **son** da **NULL** göstermesi sağlanır.
- Aksi durumda ise **bas** indisi bir artar.

27.02.2017 Dr.Eyyüp GÜLBANDILAR 20

Özel Kuyruk Yapıları

Her İki Tarafından Ekleme ve Çıkarma Yapılan Kuyruk : Bu kuyruk yapısında kuyruğun Hem 'Kb' ve hemde 'Ks' indisleri ile istenilen taraflarında hem eleman eklenir ve hem de eleman çıkarılır. Ortasında bulunan bir elemana ulaşma problemine çözüm bulunmalıdır.



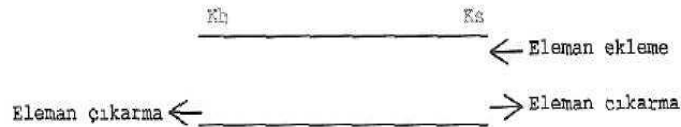
27.02.2017

Dr.Eyyüp GÜLBANDILAR

21

Özel Kuyruk Yapıları

■ **Girişi Sınırlandırılmış Çift Yönlü Kuyruk:** Bu kuyruk yapısında kuyruğun bir tarafından hem eleman ekleme ve hem de eleman çıkarma yapılırken kuyruğun diğer tarafından sadece eleman çıkarma yapılır. Yani kuyruğun bir tarafından eleman ekleme yapılmaz. Bu kuyruk yapısı bir önceki yapıda oluşan problemi çözer.



27.02.2017

Dr.Eyyüp GÜLBANDILAR

22

Özel Kuyruk Yapıları

■ **Çıkışı Sınırlanmış Çift Yönlü Kuyruk:** Çıkışı sınırlanmış çift yönlü kuyruk yapısında kuyruğun bir tarafından hem eleman ekleme ve hem de eleman çıkarma işlemine izin verilirken kuyruğun diğer tarafında sadece kuyruğa eleman ekleme işlemine izin verilir.

