1.SORU

i)

SABİT DİZİ KUYRUĞA ELEMAN EKLEME

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

O(1)

SABİT DİZİ KUYRUKTAN ELEMAN SİLME

```
Ornek kuyrukSil(){
    Ornek sonuc;
    İf(!kuyrukBos()){
        sonuc = dizi[bas];
        bas = (bas+1) % N;
        return sonuc;
    }
    return null;
```

```
}
```

ii)

O(1)

BAĞLI LİSTE KUYRUĞA ELEMAN EKLEME

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

O(1)

BAĞLI LİSTE KUYRUKTAN ELEMAN SİLME

```
Eleman kuyrukSil(){
    Eleman sonuc;
    sonuc = bas;
```

```
        if(!kuyrukBos()){
            bas = bas.ileri;
            if(bas == null)
                son = null;
        }
        return sonuc;
    }
```

O(1)

iii)

Alan karmaşıklığı

Bağlı liste :O(n)

Sabit Dizi:O(1)

Dizilerde uygulama başlatılmadan önce boyutunun belirlenmesi gerekir.

Bağlı listelerde ise böyle bir kısıtlama yoktur.

Burdan kullanılıcak verinin bilindigi durumlarda bağlı listeleri kullanmanın daha mantıklı olacağını söyleyebilirim.

Bir diger yandan, Bağlantılı listenin bellek yükü genellikle her öğede fazladan bir işaretçinin depolanması nedeniyle toplam O (n) fazladan bellektir.Dinamik bir dizinin bellek kullanımı

oldukça iyidir - dizi tamamen dolu olduğunda, örneğin, sadece O (1) fazladan ek yük vardır.

Dinamik belleginde alan maliyeti açısından böyle bir avantajı vardır.

Verilen probleme göre kullanacağımız yöntem değişir.

(hocam ikiside O(n) demişsiniz fakat sabit dizide pointer yok ama)

2.SORU

KUYRUK CLASS

```
public class Kuyruk {
    Eleman bas;
    Eleman son;
    public Kuyruk(){
        bas=null;
        son=null;
    }
    boolean kuyrukBos(){
        if(bas == null)
            return true;
        else
            return false;
    }
    void kuyrugaEkle(Eleman yeni){
        if(!kuyrukBos())
            son.ileri=yeni;
        else
            bas=yeni;
        son=yeni;
    }
    Eleman kuyrukSil(){
        Eleman sonuc;
        sonuc=bas;
        if(!kuyrukBos()){
            bas=bas.ileri;
            if (bas==null)
                son=null;
        }
        return sonuc;
    }
    void  kuyrukYaz(){
        Eleman tmp;
        tmp=bas;
        while (true){
```

```
            System.out.println("Kuyrukta bekleyen sayı:"+tmp.icerik);
            tmp=tmp.ileri;

            if(tmp==null){
                break;
            }

        }
    }
    void kuyrukEnBuyukBul(){
        Eleman tmp1;
        Eleman tmpEnBuyuk;
        tmp1=bas;
        tmpEnBuyuk=bas;
        tmp1.icerik=bas.icerik;
        tmpEnBuyuk.icerik=bas.icerik;

        while(true){

            if(tmp1.icerik>tmpEnBuyuk.icerik){
                tmpEnBuyuk.icerik=tmp1.icerik;
            }

            tmp1=tmp1.ileri;

            if (tmp1==null)
                break;
        }
        System.out.println("Kuyruktaki en büyük sayı:"+tmpEnBuyuk.icerik);
    }
    void enBuyukYerDegistir(){
        Eleman tmp1;
        Eleman tmp2;
        Eleman tmp3;
        Eleman tmpEnBuyuk;
        tmp1=bas;
        tmp2=bas;
        tmp3=bas;
        tmp3.icerik=bas.icerik;
        tmpEnBuyuk=bas;
        tmp1.icerik=bas.icerik;
        tmpEnBuyuk.icerik=bas.icerik;

        while(true){

            if(tmp1.icerik>tmpEnBuyuk.icerik){
                tmpEnBuyuk.icerik=tmp1.icerik;
            }

            tmp1=tmp1.ileri;

            if (tmp1.ileri==null)
                break;
        }

        System.out.println("Kuyruktaki en büyük sayı:"+tmpEnBuyuk.icerik);//EN
BÜYÜGÜ BİR DAHA YAZDIRDIM GEREKSİZ ASLINDA
```

```
        while (true) {
            tmp2=tmp2.ileri;
            if(tmpEnBuyuk.icerik==tmp2.icerik){
                break;
            }
        }
        bas.icerik=tmp2.icerik;
        tmp2.icerik=tmp3.icerik;


    }
}
```

MAİN() FONKSİYONU

```
public static void main(String[] args) {
    Eleman e1=new Eleman(25);
    Eleman e2=new Eleman(22);
    Eleman e3=new Eleman(45);
    Eleman e4=new Eleman(27);
    Eleman e5=new Eleman(56);
    Eleman e6=new Eleman(33);

    Kuyruk kuyruk=new Kuyruk();

    kuyruk.kuyrugaEkle(e1);
    kuyruk.kuyrugaEkle(e2);
    kuyruk.kuyrugaEkle(e3);
    kuyruk.kuyrugaEkle(e4);
    kuyruk.kuyrugaEkle(e5);
    kuyruk.kuyrugaEkle(e6);

    kuyruk.kuyrukYaz();

    kuyruk.kuyrukEnBuyukBul();

    kuyruk.enBuyukYerDegistir();

    kuyruk.kuyrukYaz();

}
```

EKRAN ALINTISI

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C
Kuyrukta bekleyen sayı:25
Kuyrukta bekleyen sayı:22
Kuyrukta bekleyen sayı:45
Kuyrukta bekleyen sayı:27
Kuyrukta bekleyen sayı:56
Kuyrukta bekleyen sayı:33
Kuyruktaki en büyük sayı:56
Kuyruktaki en büyük sayı:56
Kuyrukta bekleyen sayı:56
Kuyrukta bekleyen sayı:22
Kuyrukta bekleyen sayı:45
Kuyrukta bekleyen sayı:27
Kuyrukta bekleyen sayı:56
Kuyrukta bekleyen sayı:33

Process finished with exit code 0
```

QUEUE ARAYÜZÜ İLE KUYRUK YAPISI

```
public static void main(String[] args) {
    Queue<Integer> kuyruk = new LinkedList<Integer>();

    kuyruk.offer(25);
    kuyruk.offer(22);
    kuyruk.offer(45);
    kuyruk.offer(27);
    kuyruk.offer(56);
    kuyruk.offer(33);

    Iterator it = kuyruk.iterator();


    while (it.hasNext()) {
        Integer iteratorValue = (Integer) it.next();
        System.out.println("Kuyrukta bekleyen sayı :" + iteratorValue);
    }


    System.out.println("*****************************************");

    Iterator it2=kuyruk.iterator();
```

```
    int enBuyuk=kuyruk.element();

    while(it2.hasNext()) {
        Integer iteratorValue = (Integer) it2.next();
        if (iteratorValue>enBuyuk){
            enBuyuk=iteratorValue;
        }
    }
    System.out.println("Kuyruktaki en büyük sayı :"+enBuyuk);



}
```

EKRAN ALINTISI

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Fi
Kuyrukta bekleyen sayı :25
Kuyrukta bekleyen sayı :22
Kuyrukta bekleyen sayı :45
Kuyrukta bekleyen sayı :27
Kuyrukta bekleyen sayı :56
Kuyrukta bekleyen sayı :33
****************************************
Kuyruktaki en büyük sayı :56

Process finished with exit code 0
```