



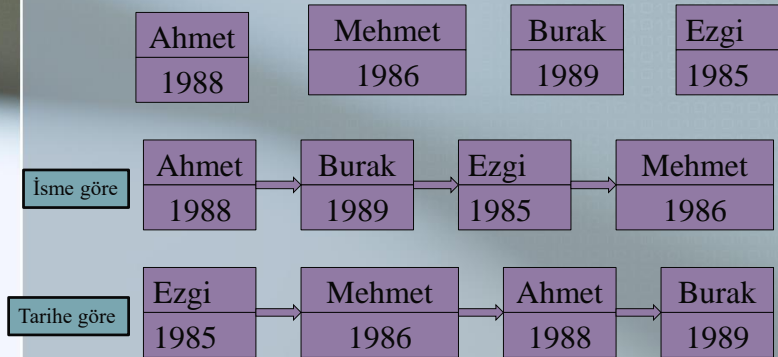
SAIRALAMA ALGORİTMALARI

Sıralama Nedir?

Sayısal ortamdaki bilgilerin veya verilerin belirli bir anahtar sözcüğe göre sıralı erişilmesini sağlama işlemidir. Bilgilerin yada verilerin sıralı olması bilgiye erişimi kolaylaştırır, sadeleştirir ve işlemin daha hızlı yapılmasını sağlar.

Sıralama Nasıl Yapılır?

Sıralama belirli bir anahtar sözcüğe göre yapılır.



27.02.2017

Dr.Eyyüp GÜLBANDILAR

Araya Sokma (Insertion) Algoritması

Sıralanacak dizinin ilk elemanı yerinde bırakılır. Sıralanacak bundan sonraki elemanları sıraya alarak **uygun** yere sokar.

- Sıralı olan dizilere yeni eleman eklemek için uygundur.
- Araya sokulacak elemanlar için kaydırma gerekir ve bu da zaman kaybına neden olur.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Araya Sokma (Insertion) Algoritması

Sıralanacak dizi; [8,2,6,7,11,15,5]

[8,2] [6,7,11,15,5] [2,8] [6,7,11,15,5]

[2,8,6] [7,11,15,5] [2,6,8] [7,11,15,5]

[2,6,8,7] [11,15,5] [2,6,7,8] [11,15,5]

[2,6,7,8,11] [15,5] [2,6,7,8,11] [15,5]

[2,6,7,8,11,15] [5] [2,6,7,8,11,15] [5]

[2,6,7,8,11,15,5] [] [2,5,6,7,8,11,15]

Sıralanacak kümenin eleman sayısından bir az satırda işlemi tamamlanır.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Araya Sokma (Insertion) Algoritması

*/*Araya sokma sıralamasıyla dizinin sıralaması*

arayaSokma(int D[], int N)

{

int i, k, ekle;

for (i=1; i<N; i++) {

ekle=D[i];

for (k=i-1; k>=&&ekle<=D[k]; k--)

*D[k+1]=D[k]; /*geriye kaydırma*/*

*D[k+1]=ekle; /*uygun yer boşaltıldı; eklendi*/*

}

}

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Seğmeli Sıralama (Selection Sort)

Dizinin başından yada sonundan başlanır. Önce ilk eleman alınır ve dizi içindeki en küçük eleman aranır (küçükten büyüğe doğru sıralama için). İlk elemanla en küçük eleman ile yer değiştirilir. Daha sonra ikinci eleman alınır ve ikinci küçük eleman aranır sonra ikinci eleman ile yer değiştirilir. İşlemler son elemana kadar tekrarlanır.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Seğmeli Sıralama (Selection Sort)

Sıralanacak dizi; [8,2,6,7,11,15,5]

[] [8,2,6,7,11,15,5]	[2] [8,6,7,11,15,5]
[2] [8,6,7,11,15,5]	[2,5] [6,7,11,15,8]
[2,5] [6,7,11,15,8]	[2,5,6] [7,11,15,8]
[2,5,6] [7,11,15,8]	[2,5,6,7] [11,15,8]
[2,5,6,7] [11,15,8]	[2,5,6,7,8] [15,11]
[2,5,6,7,8] [15,11]	[2,5,6,7,8,11] [15]
[2,5,6,7,8,11] [15]	[2,5,6,7,8,11,15]

Sıralanacak en küçük eleman sıradaki elemanla yer değiştirir ve diziye alınır.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Seçmeli Sıralama (Selection Sort)

```

/*Seçmeli sıralamasıyla dizinin sıralaması
secmelisiralama(int D[], int N)
{
    int i, index, j, enkucuk;
    for (i=0; i<(N-1); i++) { /*dizinin ilk elemanından başla*/
        enkucuk=D[N-1]; /*son eleman küçük kabul edilir*/
        index=N-1;
        for (j=i; j<(N-1); j++) /*daha küçük var mı*/
            if (D[j]<enkucuk) {
                enkucuk=D[j];
                index=j;
            }
        D[index]=D[i]; /*daha küçüğü varsa değiştirilir*/
        D[i]=enkucuk;
    }
}

```

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Kabarcık Sıralama (Bubble Sort)

Sıralanacak eleman bir yönden diğer yöne ilerlerken komşu iki eleman yer değiştirilir. Komşu iki eleman yer değiştirilerek uygun elemanlar yer değiştirilir. İşleme bütün elemanlar sıralanana kadar devam edilir. Az elemana sahip dizilerde avantajlı bir sıralamadır.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Kabarcık Sıralama (Bubble Sort)

Sıralanacak dizi; [8,2,6,7,11,15,5]

k=0 8 ile 2 karşılaştırılır yer değiştirilir [2,8,6,7,11,15,5]

k=1 8 ile 6 karşılaştırılır yer değiştirilir [2,6,8,7,11,15,5]

k=2 8 ile 7 karşılaştırılır yer değiştirilir [2,6,7,8,11,15,5]

k=3 8 ile 11 karşılaştırılır aynı kalır [2,6,7,8,11,15,5]

k=4 11 ile 15 karşılaştırılır aynı kalır [2,6,7,8,11,15,5]

k=5 15 ile 5 karşılaştırılır yer değiştirilir [2,6,7,8,11,5,15]

İşlem tekrarlanır.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Kabarcık Sıralama (Bubble Sort)

```
/*Kabarcık sıralamasıyla dizinin sıralaması
kabarcik(int D[], int N)
{
    int gecici, k, hareket;
    for (hareket=0; hareket<(N-1); hareket++){
        for (k=0; k<(N-1-hareket) k++)
            if (D[k]>D[k+1]) { /*komşular karşılaştırılır*/
                gecici=D[k]; /*yer değiştirme*/
                D[k]=D[k+1];
                D[k+1]=gecici;
            }
    }
}
```

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Birleşmeli Sıralama (Merge Sort)

Algoritma böl ve yönet yaklaşımına dayanır. Sıralanacak elemanlar önce iki alt kümeye ayrılır, parçalama işlemi alt kümelerdeki eleman sayısı birer tane olana kadar devam eder. Geriye doğru gidilerek alt küme içindeki elemanlar sıralanır. Rekürsif yapıya uygun bir algoritmadır.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Birleşmeli Sıralama (Merge Sort)

Sıralanacak dizi; [8,2,6,7,11,15,5]

[2,8,6,7,11,15,5]

[2,8,6,7] [11,15,5]

[2,8] [6,7] [11,15] [5]

[2] [8] [6] [7] [11] [15] -

[2,8] [6,7] [11,15] -

[2,6,7,8] [5,11,15]

[2,5,6,7,8,11,15]

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Birleşmeli Sıralama (Merge Sort)

```

/*Bieleşmeli sıralamasyyla dizinin sıralaması
Birlesmeli(D[], sol, sag)
{
    if (sol < sag) {
        k = (sol + sag) / 2;
        birlesmeli(D, sol, k);
        birlesmeli(D, k + 1, sag);
        birlestir(D, sol, k, sag)
    }
    int i, j;
    int M[];
    for (i = sol; j = k + 1; i <= k && j <= sag; {
        if (D[i] < D[j]) {
            D[i] 'yi al M[] 'ye ekle
            i++
        }
        else {
            D[j] 'yi al M[] 'ye ekle
            j++
        }
    }
    M[] 'yi A[] 'ye kopyala;
}

```

Dr.Eyyüp GÜLBANDILAR

27.02.2017

Hızlı Sıralama (Quick Sort)

Böl ve yönet prensibine dayanır. Sıralanması istenen dizi belirli bir sınır (pivot) değerine göre ikiye ayrılır. Bu sınır değerinden küçükler bir tarafa büyükler diğer tarafa ayrılır. İşlemler tek eleman kalıncaya kadar devam edilir.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Hızlı Sıralama (Quick Sort)

Sıralanacak dizi; [8,2,6,7,11,15,5] sınır 7 olsun;

A1=[2,6,5] A2=[8,11,15] 7

İşlem tekrarlanır.

27.02.2017

Dr.Eyyüp GÜLBANDILAR

Hızlı Sıralama (Quick Sort)

```

/*hızlı sıralama dizinin sıralaması
hsirala(int dizi[], int sol, int sag)
{
    register int k, j;
    int ortadaki, gecici;
    /*diziyi ikiye parçalama*/
    k=sol;
    j=sag;
    ortadaki=dizi[(sol+sag)/2] /*sınır değeri*/
    do {
        while (dizi[k]<ortadaki && k<sag)
            k++;
        while (ortadaki<dizi[j] && j>sol)
            j--;
        if (k<=j) {
            gecici=dizi[k];
            dizi[k]=dizi[j];
            dizi[j]=gecici;
            k++; j--;
        }
    }while(k<=j); /*parçalama biter*/
    if (sol<j) hsirala(dizi, sol, j);
    if (k<sag) hsirala(dizi, k, sag)
}

```

27.02.2017

Dr.Eyyüp GÜLBANDILAR