

Selection Sort

Seçme Sıralaması

Selection Sort algoritması $O(n^2)$ grubuna giren bir sıralama yöntemidir. Dolayısıyla büyük sayıda verileri sıralamak için elverişsizdir. Bunun yanında, *bubble sort* algoritmasındaki takas işlemlerinin çoğunu yapmadığı için, bubble sort algoritmasının iyileştirilmiş biçimi sayılır. Çünkü takas işlemlerinin sayısını $O(n^2)$ den $O(n)$ ye düşürür. Dolayısıyla daha etkilidir. Ancak, yaptığı mukayese işlemleri gene $O(n^2)$ düzeyindedir.

Algoritmanın çalışma ilkesi basittir.

1. Başlarken dizinin ilk öğesini en küçük öğe kabul eder. Tabii, bu kabul geçicidir. Sonra kalan terimler arasında daha küçükler varsa, onların en küçüğü olan terimle takas eder. O terimi en sola koyar; bu terim sıralama sonunda ilk terim olacaktır.
2. Diziden seçilen bu en küçük terimi çıkarınca, geri kalan alt dizine aynı yöntemi uygular. Altdiziden seçtiği en küçük öğeyi, ilk seçtiğinin sağına koyar. Dizinin sol ucunda iki terimli alt dizi küçükten büyüğe doğru sıralı bir altdizi oluşturur.
3. Bu işleme kalan altdiziler bitene kadar devam edilir. Her adım başlarken, sol yanda sıralı bir altdizi, sağ yanda sırasız bir alt dizi vardır. Her adımda sıralı dizi bir terim artar, sırasız dizi bir terim azalır. Sağ yandaki sırasız altdizi bitince, sıralama işlemi biter.

Örnek:

int [] $a = \{3,17,86,-9,7,-11,38\}$ arrayi verilsin. Görselliği sağlamak için, bu arrayi

3 17 86 -9 7 -11 38

dizisi biçiminde yazalım. Bu dizinin öğelerini *selection sort* algoritması ile sıralayacağız. Dizinin (array) indislerinin 0 dan başladığını anımsayınız. $a[0] = 3, \dots, a[6] = 38$ dir.

Selection sort algoritması, verilen diziyi sıralı ve sırasız olmak üzere iki alt diziyeye ayırır. Sırasız alt dizinin en küçük öğesini bulup seçer ve onu sıralı diziyeye en büyük öğe olarak katar.

Başlangıçta bütün dizi sırasızdır. Dizinin ilk öğesini seçip, tek öğeli (sıralı) bir alt dizi oluşturabiliriz. Geçici olarak, verilen dizinin ilk öğesini en küçük öğe (*minimal*) imiş gibi kabul edelim. Sonra mukayese ile daha küçük terim olup olmadığını araştıracağız. Ondan daha küçükler varsa, onların en küçüğünü $a[0]$ ile takas edeceğiz. Böylece verilen dizinin en küçük terimi en sola yerleşir.

1.Aşama

Başlangıçta $a[0] = 3$ olduğu için $minimal = 3$ olur. Bu eşitliği sağlayan indise *minIndex* diyelim. İlk değeri $minIndex = 0$ dir.

$a[0]$	$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	$a[6]$
3	17	86	-9	7	-11	38

Sonra *minimal* 3 sayısını, sırayla dizinin öteki terimleriyle karşılaştırarak, 3 den daha küçük öğe olup olmadığını, varsa onların en küçüğünün hangisi olduğunu arayalım. 17 ve 86 terimleri koşulu

sağlamaz; onları atlıyoruz. Üçüncü $a[3]$ terimi ile mukayese yapınca $-9 < 3$ olduğunu görüyoruz. Bu durumda daha küçük olan -9 teriminin indisini *minIndex* yapıyoruz.

$$\text{minIndex} = 3$$

Bu andan sonra minimal öğemiz 3 değil -9 olmuştur. Ondan sonraki terimleri -9 ile karşılaştıracamız. 7 koşulu sağlamaz, onu atlıyoruz. Beşinci $a[5]$ teriminde $-11 < -9$ olduğu için, $\text{minIndex} = 5$ olur. Bu aşamada minimal öğe indisi 5 olan -11 öğesidir. Kalan 38 terimini -11 ile mukayese ediyor ve koşulu sağlamadığını görüyoruz. O halde dizinin en küçük öğesi indisi $\text{minIndex} = 5$ olan -11 öğesidir. Dolayısıyla, 3 ile -11 öğelerinin yerlerini değiştiriyoruz (takas işlemi).

-11	17	86	-9	7	3	38	<i>minIndex = 5</i>
------------	----	----	----	---	---	----	---------------------

2.Aşama

Bu aşamada dizi *sıralı* $\{-11\}$ ve *sırasız* $\{17, 86, -9, 7, 3, 38\}$ olmak üzere iki altdiziye ayrılmıştır. Şimdi sırasız altdiziye yukarıdaki seçme algoritmasını uygulayarak, en küçük öğesini seçebiliriz. Bunun -9 olacağını görüyoruz. Alt dizinin ilk öğesi olan 17 terimi ile en küçük öğesi olan -9 terimlerinin yerlerini değiştiriyoruz (takas). Sonunda, -9 terimini sıralı alt diziye ekliyoruz:

-11	-9	86	17	7	3	38	<i>minIndex = 3</i>
------------	-----------	----	----	---	---	----	---------------------

3.Aşama

Bu aşamada dizi *sıralı* $\{-11, -9\}$ ve *sırasız* $\{86, 17, 7, 3, 38\}$ olmak üzere iki altdiziye ayrılmıştır. Şimdi sırasız altdiziye seçme algoritmasını uygulayarak, en küçük öğesini seçebiliriz. Bunun 3 olacağını görüyoruz. Alt dizinin ilk öğesi olan 86 terimi ile en küçük öğesi olan 3 terimlerinin yerlerini değiştiriyoruz (takas). Sonunda, 3 terimini sıralı alt diziye ekliyoruz:

-11	-9	3	17	7	86	38	<i>minIndex = 0</i>
------------	-----------	----------	----	---	----	----	---------------------

4.Aşama

Bu aşamada dizi *sıralı* $\{-11, -9, 3\}$ ve *sırasız* $\{17, 7, 86, 38\}$ olmak üzere iki altdiziye ayrılmıştır. Şimdi sırasız altdiziye seçme algoritmasını uygulayarak, en küçük öğesini seçebiliriz. Bunun 7 olacağını görüyoruz. Alt dizinin ilk öğesi olan 17 terimi ile en küçük öğesi olan 7 terimlerinin yerlerini değiştiriyoruz (takas). Sonunda, 7 terimini sıralı alt diziye ekliyoruz:

-11	-9	3	7	17	86	38	<i>minIndex = 4</i>
------------	-----------	----------	----------	----	----	----	---------------------

5.Aşama

Bu aşamada dizi *sıralı* $\{-11, -9, 3, 7\}$ ve *sırasız* $\{17, 86, 38\}$ olmak üzere iki altdiziye ayrılmıştır. Şimdi sırasız altdiziye seçme algoritmasını uygulayarak, en küçük öğesini seçebiliriz. Bunun 17 olacağını görüyoruz. Alt dizinin ilk öğesi zaten 17 terimidir. Dolayısıyla bir takas işlemi gerekmiyor. Sonunda, 17 terimini sıralı alt diziye ekliyoruz:

-11	-9	3	7	17	86	38	<i>minIndex = 2</i>
------------	-----------	----------	----------	-----------	----	----	---------------------

6.Aşama

Bu aşamada dizi *sıralı* $\{-11, -9, 3, 7, 17\}$ ve *sırasız* $\{86, 38\}$ olmak üzere iki alt diziye ayrılmıştır. Şimdi sırasız alt diziye seçme algoritmasını uygulayarak, en küçük öğesini seçebiliriz. Bunun 38 olacağını görüyoruz. Alt dizinin ilk öğesi olan 86 terimi ile en küçük öğesi olan 38 terimlerinin yerlerini değiştiriyoruz (takas). Sonunda, 38 terimini sıralı alt diziye ekliyoruz:

-11	-9	3	7	17	38	86	<i>minIndex = 6</i>
-----	----	---	---	----	----	----	---------------------

7.Aşama

Bu aşamada dizi *sıralı* $\{-11, -9, 3, 7, 17, 38\}$ ve *sırasız* $\{86\}$ olmak üzere iki alt diziye ayrılmıştır. Sırasız dizi tek öğeli olduğu için, en küçük öğesi kendisidir. 86 terimini sıralı alt diziye ekliyoruz:

-11	-9	3	7	17	38	86	<i>minIndex = 2</i>
-----	----	---	---	----	----	----	---------------------

Böylece verilen dizi sıralanmış olur.

Şimdi selection sort algoritmasını yapan bir java metodu yazalım:

```
void selectionSort(int [] dizi,int n)
{
    int yedek;
    int minIndex;
    for(int i=0; i<n-1; i++)
    {
        minIndex=i;
        for(int j=i; j<n; j++)
        {
            if (dizi[j] < dizi[minIndex]) minIndex=j;
        }
        temp=dizi[i];
        dizi[i]=dizi[minIndex];
        dizi[minIndex]=yedek;
    }
}
```

Yukarıdaki metodu bir java uygulamasında çalıştıralım:

package sıralama;

```
public class SelectionSort {
    int[] a = {3,17,86,-9,7,-11,38} ;

    void selectionSort(int [] dizi)
    {
        int yedek;
        int minIndex;
        for(int i=0; i< dizi.length; i++)
        {
            minIndex=i;
            for(int j=i; j<dizi.length; j++)
            {
                if (dizi[j] < dizi[minIndex]) minIndex=j;
            }
            yedek=dizi[i];
            dizi[i]=dizi[minIndex];
            dizi[minIndex]=yedek;
        }
    }

    void diziYaz(int[] arr){
        for(int i=0; i < arr.length;i++){
            System.out.print(arr[i] + " ");
        }
    }

    public static void main(String[] args) {
        SelectionSort ss = new SelectionSort();
        System.out.println("\nSıralamadan önce:");
        ss.diziYaz(ss.a);
        ss.selectionSort(ss.a);
        System.out.println("\n\nSıralamadan sonra:");
        ss.diziYaz(ss.a);

    }
}
```