

CENG 213 Veri Yapıları 11: Sıralama Algoritmaları

Öğr.Gör. Şevket Umut ÇAKIR

Pamukkale Üniversitesi

Hafta 11

1 Seçmeli Sıralama(Selection Sort)

2 Kabarcık Sıralama

3 Eklemeli Sıralama

- Eldeki n adet değeri ya da elemanı mantıksal bir sıraya göre sıralamayı sağlayan algoritmalara *Sıralama Algoritmaları* adı verilir
- Karşılaştırmalı ve karşılaştırmasız türleri bulunmaktadır.
- En yaygın örneği n elemanlı bir dizinin sıralanmasıdır.

Seçmeli Sıralama(Selection Sort)

- Tüm diziyi arayarak en küçük eleman bulunur ve ilk elemanla yer değiştirilir.
- Kalan dizide(ilk eleman hariç) en küçük eleman bulunur ve dizideki ikinci elemanla yer değiştirilir.
- Bu şekilde dizideki bütün elemanlar uygun sıraya gelene kadar işleme devam edilir.

Seçmeli Sıralama Algoritması

Input: Sıralanabilir $A[0 \dots n - 1]$ dizisi

Output: Artan sırada sıralanmış $A[0 \dots n - 1]$ dizisi

```
for  $i \leftarrow 0$  to  $n - 2$  do  
     $min \leftarrow i$   
    for  $j \leftarrow i + 1$  to  $n - 1$  do  
        if  $A[j] < A[min]$  then  
             $min \leftarrow j$   
        end  
    end  
    swap  $A[i]$  and  $A[min]$   
end
```

Seçmeli Sıralama Kodu

```
1 void sort(int arr[])
2 {
3     int n = arr.length;
4     // One by one move boundary of unsorted subarray
5     for (int i = 0; i < n-1; i++)
6     {
7         // Find the minimum element in unsorted array
8         int min_idx = i;
9         for (int j = i+1; j < n; j++)
10             if (arr[j] < arr[min_idx])
11                 min_idx = j;
12         // Swap the found minimum element with the first
13         // element
14         int temp = arr[min_idx];
15         arr[min_idx] = arr[i];
16         arr[i] = temp;
17     }
18 }
```

Seçmeli Sıralama Örneği

| 89 45 68 90 29 34 17

Seçmeli Sıralama Örneği

	89	45	68	90	29	34	17
17	45	68	90	29	34	89	

Seçmeli Sıralama Örneği

	89	45	68	90	29	34	17
17		45	68	90	29	34	89
17	29		68	90	45	34	89

Seçmeli Sıralama Örneği

	89	45	68	90	29	34	17
17		45	68	90	29	34	89
17	29		68	90	45	34	89
17	29	34		90	45	68	89

Seçmeli Sıralama Örneği

	89	45	68	90	29	34	17
17		45	68	90	29	34	89
17	29		68	90	45	34	89
17	29	34		90	45	68	89
17	29	34	45		90	68	89

Seçmeli Sıralama Örneği

89	45	68	90	29	34	17
17	45	68	90	29	34	89
17	29	68	90	45	34	89
17	29	34	90	45	68	89
17	29	34	45	90	68	89
17	29	34	45	68	90	89

Seçmeli Sıralama Örneği

89	45	68	90	29	34	17
17	45	68	90	29	34	89
17	29	68	90	45	34	89
17	29	34	90	45	68	89
17	29	34	45	90	68	89
17	29	34	45	68	90	89
17	29	34	45	68	89	90

- En iyi veya en kötü durumdan söz edilebilir mi?
- En kötü durumda kaç karşılaştırma yapılır?

- En iyi veya en kötü durumdan söz edilebilir mi?
- En kötü durumda kaç karşılaştırma yapılır?
- Algoritma zaman karmaşıklığı: $\Theta(n^2)$

$$C(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \frac{(n-1)n}{2}$$

Kabarcık Sıralama(Bubble Sort)

- Baştan sona doğru elemanlar ikili olarak(komşusuyla birlikte) birer kabarcık olarak düşünülür.
- Kabarcık içindeki elemanlar karşılaştırılır ve sıralanır.
- İlk turun sonunda en büyük eleman en sona yerleşmiş olur.
- Bu işlem $n - 1$ defa yapıldığında dizi sıralanmış olur.

Kabarcık Sıralama Algoritması

Input: Sıralanabilir $A[0 \dots n - 1]$ dizisi

Output: Artan sırada sıralanmış $A[0 \dots n - 1]$ dizisi

```
for  $i \leftarrow 0$  to  $n - 2$  do
    for  $j \leftarrow 0$  to  $n - 2 - i$  do
        if  $A[j + 1] < A[j]$  then
            swap  $A[j]$  and  $A[j + 1]$ 
        end
    end
end
```

Kabarcık Sıralama Kodu

```
1 void bubbleSort(int arr[])
2 {
3     int n = arr.length;
4     for (int i = 0; i < n-1; i++)
5         for (int j = 0; j < n-i-1; j++)
6             if (arr[j] > arr[j+1])
7                 {
8                     // swap arr[j+1] and arr[i]
9                     int temp = arr[j];
10                    arr[j] = arr[j+1];
11                    arr[j+1] = temp;
12                }
13 }
```

Kabarcık Sıralama Örneği

Örnek Dizi İçin 2 Geçiş

89 \rightleftarrows 45 68 90 29 34 17

Kabarcık Sıralama Örneği

Örnek Dizi İçin 2 Geçiş

89	\rightleftarrows	45		68		90		29		34		17
45		89	\rightleftarrows	68		90		29		34		17

Kabarcık Sıralama Örneği

Örnek Dizi İçin 2 Geçiş

89	\Rightarrow	45		68		90		29		34		17
45		89	\Rightarrow	68		90		29		34		17
45		68		89	\Rightarrow	90	\Rightarrow	29		34		17

Kabarcık Sıralama Örneği

Örnek Dizi İçin 2 Geçiş

89	\Rightarrow	45		68		90		29		34		17
45		89	\Rightarrow	68		90		29		34		17
45		68		89	\Rightarrow	90	\Rightarrow	29		34		17
45		68		89		29		90	\Rightarrow	34		17

Kabarcık Sıralama Örneği

Örnek Dizi İçin 2 Geçiş

89	\Rightarrow	45		68		90		29		34		17
45		89	\Rightarrow	68		90		29		34		17
45		68		89	\Rightarrow	90	\Rightarrow	29		34		17
45		68		89		29		90	\Rightarrow	34		17
45		68		89		29		34		90	\Rightarrow	17

Kabarcık Sıralama Örneği

Örnek Dizi İçin 2 Geçiş

89	\Rightarrow	45		68		90		29		34		17
45		89	\Rightarrow	68		90		29		34		17
45		68		89	\Rightarrow	90	\Rightarrow	29		34		17
45		68		89		29		90	\Rightarrow	34		17
45		68		89		29		34		90	\Rightarrow	17
45		68		89		29		34		17		90

Kabarcık Sıralama Örneği

Örnek Dizi İçin 2 Geçiş

89	\rightleftarrows	45		68		90		29		34		17
45		89	\rightleftarrows	68		90		29		34		17
45		68		89	\rightleftarrows	90	\rightleftarrows	29		34		17
45		68		89		29		90	\rightleftarrows	34		17
45		68		89		29		34		90	\rightleftarrows	17
45		68		89		29		34		17		90
45	\rightleftarrows	68	\rightleftarrows	89	\rightleftarrows	29		34		17		90

Kabarcık Sıralama Örneği

Örnek Dizi İçin 2 Geçiş

89	\rightleftarrows	45		68		90		29		34		17
45		89	\rightleftarrows	68		90		29		34		17
45		68		89	\rightleftarrows	90	\rightleftarrows	29		34		17
45		68		89		29		90	\rightleftarrows	34		17
45		68		89		29		34		90	\rightleftarrows	17
45		68		89		29		34		17		90
45	\rightleftarrows	68	\rightleftarrows	89	\rightleftarrows	29		34		17		90
45		68		29		89	\rightleftarrows	34		17		90

Kabarcık Sıralama Örneği

Örnek Dizi İçin 2 Geçiş

89	\rightleftarrows	45		68		90		29		34		17
45		89	\rightleftarrows	68		90		29		34		17
45		68		89	\rightleftarrows	90	\rightleftarrows	29		34		17
45		68		89		29		90	\rightleftarrows	34		17
45		68		89		29		34		90	\rightleftarrows	17
45		68		89		29		34		17		90
45	\rightleftarrows	68	\rightleftarrows	89	\rightleftarrows	29		34		17		90
45		68		29		89	\rightleftarrows	34		17		90
45		68		29		34		89	\rightleftarrows	17		90

Kabarcık Sıralama Örneği

Örnek Dizi İçin 2 Geçiş

89	\rightleftarrows	45		68		90		29		34		17
45		89	\rightleftarrows	68		90		29		34		17
45		68		89	\rightleftarrows	90	\rightleftarrows	29		34		17
45		68		89		29		90	\rightleftarrows	34		17
45		68		89		29		34		90	\rightleftarrows	17
45		68		89		29		34		17		90
45	\rightleftarrows	68	\rightleftarrows	89	\rightleftarrows	29		34		17		90
45		68		29		89	\rightleftarrows	34		17		90
45		68		29		34		89	\rightleftarrows	17		90
45		68		29		34		17		89		90

- Geliştirme uygulanabilir mi?

- Geliştirme uygulanabilir mi?
- Eğer son turda hiç bir yer değiştirme olmadıysa dizi sıralanmış demektir. Algoritma sonlandırılabilir.

- En iyi veya en kötü durumdan söz edilebilir mi?
- En kötü durumda kaç karşılaştırma yapılır?

- En iyi veya en kötü durumdan söz edilebilir mi?
- En kötü durumda kaç karşılaştırma yapılır?
- Kabarcık sıralama zaman karmaşıklığı: $\Theta(n^2)$

$$C(n) = \sum_{i=0}^{n-2} \sum_{j=0}^{n-2-i} 1 = \frac{(n-1)n}{2} \in \Theta(n^2)$$

- İskambil kağıtlarını dizmeye benzerdir.
- Dizinin bir tarafı sıralanmış olarak kabul edilir.
- Sıralı olmayan bir kısımdan eleman seçilip sıralı olan kısımdaki uygun konumuna yerleştirilir.
- Bütün elemanlar sıralı kısma gelene kadar işlemler devam ettirilir.

Eklemeli Sıralama(Insertion Sort) Algoritması

Input: Sıralanabilir $A[0 \dots n - 1]$ dizisi

Output: Artan sırada sıralanmış $A[0 \dots n - 1]$ dizisi

```
for  $i \leftarrow 1$  to  $n - 1$  do  
     $v \leftarrow A[i]$   
     $j \leftarrow i - 1$   
    while  $j \geq 0$  ve  $A[j] > v$  do  
         $A[j + 1] \leftarrow A[j]$   
         $j \leftarrow j - 1$   
    end  
     $A[j + 1] \leftarrow v$   
end
```

Eklemeli Sıralama Kod

```
1 void sort(int arr[])
2 {
3     int n = arr.length;
4     for (int i=1; i<n; ++i)
5     {
6         int key = arr[i];
7         int j = i-1;
8         /* Move elements of arr[0..i-1], that are
9            greater than key, to one position ahead
10            of their current position */
11         while (j>=0 && arr[j] > key)
12         {
13             arr[j+1] = arr[j];
14             j = j-1;
15         }
16         arr[j+1] = key;
17     }
18 }
```

Eklemeli Sıralama Örneği

89 | 45 68 90 29 34 17

Eklemeli Sıralama Örneği

89		45	68	90	29	34	17
45	89		68	90	29	34	17

Eklemeli Sıralama Örneği

89		45	68	90	29	34	17
45	89		68	90	29	34	17
45	68	89		90	29	34	17

Eklemeli Sıralama Örneği

89		45	68	90	29	34	17
45	89		68	90	29	34	17
45	68	89		90	29	34	17
45	68	89	90		29	34	17

Eklemeli Sıralama Örneği

89		45	68	90	29	34	17
45	89		68	90	29	34	17
45	68	89		90	29	34	17
45	68	89	90		29	34	17
29	45	68	89	90		34	17

Eklemeli Sıralama Örneği

89		45	68	90	29	34	17
45	89		68	90	29	34	17
45	68	89		90	29	34	17
45	68	89	90		29	34	17
29	45	68	89	90		34	17
29	34	45	68	89	90		17

Eklemeli Sıralama Örneği

89		45	68	90	29	34	17
45	89		68	90	29	34	17
45	68	89		90	29	34	17
45	68	89	90		29	34	17
29	45	68	89	90		34	17
29	34	45	68	89	90		17
17	29	34	45	68	89	90	

Eklemeli Sıralama Karmaşıklığı

- En iyi veya en kötü durumdan söz edilebilir mi?
- En kötü durumda kaç karşılaştırma yapılır?

- En iyi veya en kötü durumdan söz edilebilir mi?
- En kötü durumda kaç karşılaştırma yapılır?
- Kabarcık sıralama zaman karmaşıklığı: $\Theta(n^2)$

$$C_{worst}(n) = \sum_{i=1}^{n-1} \sum_{j=0}^{i-1} 1 = \frac{(n-1)n}{2} \in \Theta(n^2)$$

Algoritma Karmaşıklıkları

Algoritma	Karmaşıklık
Seçmeli Sıralama	$\Theta(n^2)$
Kabarcık Sıralama	$\Theta(n^2)$
Eklemeli Sıralama	$\Theta(n^2)$

- Animasyonlu web sitesi
- YouTube Yöresel Danslar
- Görselleştirme



1

¹r/ProgrammerHumor