

Bubble Sort

Kabarcık Sıralaması

Sıralama algoritmaları arasında yazılması en kolay olan, ama büyük dizilerde çok yavaş kalan bir sıralama yöntemidir. Zaman karmaşası (time complexity) $O(n^2)$ dir.

Bu algoritma şöyle çalışır. Verilen dizinin öğelerini, veriliş sırasıyla yatay bir doğru boyunca dizelim.

1.Geçiş: Verilen dizinin (soldaki) ilk öğesi ikinci öğe ile karşılaştırılır. Bu karşılaştırmada, büyük olan öğe solda kalırsa, sağ yana geçirilmesi için, iki öğenin yerleri değiştirilir (takas, swap). Sonra büyük olan öğe, üçüncü öğe ile mukayese edilir. Büyük olan sağa geçer ve dördüncü öğe ile mukayese edilir. Bu işlem verilen dizinin son öğesine gelinene kadar devam eder. Bu geçişin sonunda, verilen dizinin en büyük terimi, en sağa yerleşmiş olur.

2.Geçiş: İlk geçişte en sağa yerleşen en büyük terim çıkarıldığında geriye kalan altdizine ilk geçişin aynısı uygulanır. Bu işlem alt dizinin en büyük öğesini seçer ve sağ uca yerleştirir. Bütün diziye bakacak olursak, verilen dizinin en büyük iki terimi en sağ uca kendi aralarında sıralanmış olarak yerleşir. Sıralama sağdan sola doğru azalan sıradadır ki bu soldan sağa doğru artan sırada olmasına denktir.

3.Geçiş: İlk iki geçişte en sağa yerleşen iki büyük terim çıkarıldığında geriye kalan altdizine ilk geçişin aynısı uygulanır. Bu işlem alt dizinin en büyük öğesini seçer ve sağ uca yerleştirir. Bütün diziye bakacak olursak, verilen dizinin en büyük üç terimi en sağ uca, sağdan sola doğru azalan sırada yerleşmiş olur.

n.Geçiş: Dizinin n terimi varsa, yukarıdaki gibi her geçişten sonra geriye kalan altdizine ilk geçişte anlatılan yöntemi uygulayarak devam edelim. n geçişten sonra, dizinin terimleri sağdan sola doğru büyükten küçüğe doğru sıralanmış olacaktır. Bu ise, dizinin soldan sağa doğru artan sırada yerleşmiş olmasına denktir.

Örnek:

74 57 3 7 95 -6 17

dizisini *bubble sort* algoritması ile sıralayalım. Aşağıdaki tablolarda, dizinin sağında kırmızı ile yazılan alt diziler, sağdan sola doğru azalan sırada sıralanmışlardır.

74	57	3	7	95	-6	17
57	74	3	7	95	-6	17
57	3	74	7	95	-6	17
57	3	7	74	95	-6	17
57	3	7	74	95	-6	17
57	3	7	74	-6	95	17
57	3	7	74	-6	17	95
57	3	7	74	-6	17	95

Birinci geçişin sonu

57	3	7	74	-6	17	95	
3	57	7	74	-6	17	95	
3	7	57	74	-6	17	95	
3	7	57	-6	74	17	95	
3	7	57	-6	17	74	95	
3	7	57	-6	17	74	95	İkinci geçişin sonu
3	7	57	-6	17	74	95	
3	7	57	-6	17	74	95	
3	7	57	-6	17	74	95	
3	7	-6	57	17	74	95	
3	7	-6	17	57	74	95	
3	7	-6	17	57	74	95	Üçüncü geçişin sonu
3	7	-6	17	57	74	95	
3	7	-6	17	57	74	95	
3	-6	7	17	57	74	95	
3	7	-6	17	57	74	95	
3	7	-6	17	57	74	95	Dördüncü geçişin sonu
3	7	-6	17	57	74	95	
3	7	-6	17	57	74	95	
3	-6	7	17	57	74	95	
3	-6	7	17	57	74	95	Beşinci geçişin sonu
3	-6	7	17	57	74	95	
3	-6	7	17	57	74	95	
-6	3	7	17	57	74	95	Altıncı geçişin sonu
-6	3	7	17	57	74	95	
-6	3	7	17	57	74	95	
-6	3	7	17	57	74	95	Yedinci geçişin sonu

Bubble sort algoritmasını yapan bir java metodu yazalım.

```
public void bubbleSort(int[] arr) {
    boolean takas = true;
    int j = 0;
    int yedek;
    while (takas) {
        takas = false;
        j++;
        for (int i = 0; i < arr.length - j; i++)
        {
            if (arr[i] > arr[i + 1]) {
                yedek = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = yedek;
                takas = true;
            }
        }
    }
}
```

```

public class BubbleSort {
    int sayıDizisi[] = { 12, 93, 6, 27, 0, -42, 567, -2300, -1, 34 };

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

    public void bbSort(int[] arr) {
        boolean takas = true;
        int j = 0;
        int yedek;
        while (takas) {
            takas = false;
            j++;
            for (int i = 0; i < arr.length - j; i++) {
                if (arr[i] > arr[i + 1]) {
                    yedek = arr[i];
                    arr[i] = arr[i + 1];
                    arr[i + 1] = yedek;
                    takas = true;
                }
            }
        }
    }

    public static void main(String[] args) {
        BubbleSort bbs = new BubbleSort();
        System.out.println("\nSıralamadan önceki durum: ");
        bbs.diziYaz(bbs.sayıDizisi);
        bbs.bbSort(bbs.sayıDizisi);
        System.out.println("\nSıralamadan sonraki durum: ");
        bbs.bbSort(bbs.sayıDizisi);
        bbs.diziYaz(bbs.sayıDizisi);
    }
}

```

Yukarıdaki programı Java Collections ile daha kısa ve kolay biçimde yapabiliriz.

```
import java.util.Arrays;
public class BubleSort {
    int[] a = {3,17,86,-9,7,-11,38} ;

    public static void main(String[] args) {

        BubleSort bbs = new BubleSort();
        System.out.println("\nSıralamadan önce:");
        System.out.println(Arrays.toString(bbs.a));
        Arrays.sort(bbs.a);
        System.out.println("\n\nSıralamadan sonra:");
        System.out.println(Arrays.toString(bbs.a));
    }
}
```