

Linear Search

(Doğrusal Arama)

Bir dizi (array) içinde bir verinin olup olmadığını anlamak için kullanılan basit bir algoritmadır. *Doğrusal Arama Algoritması* aranan veriyi, arrayin ilk ögesinden başlayarak son ögesine doğru, her terimle tek tek karşılaştırır. Bu algoritma, dizi içinde aranana eşit olan bir terim bulursa onun indisini verir. Aranan veriye eşit olan terim bulamazsa -1 değerini verir. -1 değeri verdiğinde, aranan öge array içinde değildir. Array'in terimlerini, sırayla aranan öge ile karşılaştırdığı için, bu algoritmaya *dizisel arama* algoritması adı da verilir.

Doğrusal aramayı iki özel duruma ayırabiliriz:

1. Array'in sırasız olması durumu
2. Array'in sıralı olması durumu.

Şimdi bunları ayrı ayrı inceleyelim.

Array'in sırasız olması durumu

Sırasız bir array içinde bir ögenin olup olmadığını aramak için, array'in ilk teriminden başlanır, aranan değer her bir terimle mukayese edilerek, arrayin sonuna doğru gidilir. Aranan öge varsa, algoritma o terimin indisini verir. Arrayin sonuna gelindiği halde, aranan öge bulunamamışsa, algoritma -1 değerini verir.

Şimdi bu algoritmanın yalancı kodlarını (pseudo code) yazabiliriz.

a[sol...sağ] arrayi içinde bir *b* ögesinin olup olmadığı aranacaktır

1. *i = sol,...,sağ* için tekrarla:
 - 1.1. Eğer *b = a[i]* olursa, döngüyü durdur; *i* indisini ver.
2. Bulunamadıysa -1 değeri ver.

Bu yalancı kodların dediği işi yapan bir java metodu yazalım:

```
public static int linearSearch(Object[] a, int sol, int sağ, Object aranan)
{
    for (int i = sol; i < sağ; i++) {
        if (aranan.equals(a[i])) {
            return i; // aranan bulundu, return index.
        }
    }
    return -1; // aranan bulunamadı
}
```

Tabii, burada Object yerine javada her hangi bir veri tipi alınabilir. Şimdi yukarıdaki metodu bir java uygulama programı içine koyalım. Örneğimizde *Object[]* yerine *int[]* alınmıştır.

```
package DoğrusalArama;
```

```
import java.util.*;

public class LinearSearch01 {

    public int find(final int[] data, final int key) {
        for (int i = 0; i < data.length; ++i) {
            if (data[i] > key) {
                return -1;
            } else if (data[i] == key) {
                return i;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        final int arr[] = new int[10];
        System.out.println("10 tamsayı giriniz");
        Scanner input = new Scanner(System.in);
        for (int i = 0; i < arr.length; i++) {
            arr[i] = input.nextInt();
        }
        LinearSearch01 search = new LinearSearch01();
        System.out.print("Aranacak sayıyı giriniz: ");
        int num = input.nextInt();
        int n = search.find(arr, num);
        if ((n >= 0) && (n < arr.length)) {
            System.out.println("Aranan sayının indisi: " + n);
        } else {
            System.out.println("Bulunamadı");
        }
    }
}

/*
10 tamsayı giriniz
1 2 3 4 5 6 7 8 9 10
Aranacak sayıyı giriniz: 8
Aranan sayının indisi: 7
*/

/*
Çıktı:
10 tamsayı giriniz
1 2 3 4 5 6 7 8 9 10
Aranacak sayıyı giriniz: 56
Bulunamadı
*/
```

Yukarıdaki java programındaki `int[]` yerine `Object[]` olarak `Integer[]` kullanılırsa, program şöyle yazılabilir:

```
//package DoğrusalArama;

import java.util.*;

public class LinearSearch01 {
```

```

    public int linearSearch(Object[] a, int sol, int sağ, Object aranan)
    {
        for (int i = sol; i < sağ; i++) {
            if (aranan.equals(a[i])) {
                return i; // aranan bulundu, return index.
            }
        }
        return -1; // aranan bulunamadı
    }

    public static void main(String[] args) {
        final Integer[] arr = new Integer[10];
        System.out.println("10 tamsayı giriniz");
        Scanner input = new Scanner(System.in);
        for (int i = 0; i < arr.length; i++) {
            arr[i] = input.nextInt();
        }
        LinearSearch01 search = new LinearSearch01();
        System.out.print("Aranacak sayıyı giriniz: ");
        int num = input.nextInt();
        int n = search.linearSearch(arr, 0, 9, num);
        if ((n >= 0) && (n < arr.length)) {
            System.out.println("Aranan sayının indisi: " + n);
        } else {
            System.out.println("Bulunamadı");
        }
    }
}
/*
10 tamsayı giriniz
13 28 48 54 76 89 37 100 -2 64
Aranacak sayıyı giriniz: 89
Aranan sayının indisi: 5
*/

```

Array'in sıralı olması durumu

Arrayin öge sayısı çok değilse *Doğrusal Arama* algoritması tercih edilebilir. Ama öge sayısı çok olduğunda bunun yerine *Binary Search* algoritması tercih edilmelidir.