

ALGORİTMA ANALİZİ ÖDEV2

Dizideki en büyük elemanı bulma kaba kodu:

1-başla

2-bir dizi oluşturun

3-dizinin ilk elemanını en büyük olarak varsayıp bunu bir değişkene ata

4-sonraki elemanları bir döngü içinde gez

5-eğer en büyük sayıdan daha büyük sayı varsa o sayıyı değişkene ata yeni en büyük sayı o olsun

6-dizi elemanları bitene kadar devam et

7-bitir

İTERATİF KOD:

using System;

class Program

{

static void Main(string[] args)

{

int[] dizi = { 10, 5, 20, 15, 25 };

int enBuyuk = EnBuyukElemaniBul(dizi);

Console.WriteLine("Dizinin en büyük elemanı: " + enBuyuk);

}

static int EnBuyukElemaniBul(int[] dizi)

{

```

int enBuyuk = dizi[0];

for (int i = 1; i < dizi.Length; i++)
{

    if (dizi[i] > enBuyuk)
    {

        enBuyuk = dizi[i];
    }
}

return enBuyuk;
}
}

```

Dizinin elemanlarını kontrol etmek için bir döngü kullanılır. Bu döngü, dizi boyutu kadar çalışır. Her bir döngü adımında, mevcut elemanın değeri diğer elemanlarla karşılaştırılır. Yani toplam döngü adım sayısına bağlıdır. Dizi boyutuna n dersek döngü sayısı n olur. Sabit sayıda işlem olacağı için zaman karmaşıklığı $O(n)$ bulunur.

REKÜRSİF KOD:

```

using System;

class Program
{
    static void Main(string[] args)

```

```
{

    int[] dizi = { 10, 5, 20, 15, 25 };

    int enBuyuk = EnBuyukElemaniBul(dizi, dizi.Length);

    Console.WriteLine("Dizinin en büyük elemanı: " + enBuyuk);
}

static int EnBuyukElemaniBul(int[] dizi, int n)
{

    if (n == 1)
        return dizi[0];

    int enBuyukKalan = EnBuyukElemaniBul(dizi, n - 1);

    return Math.Max(enBuyukKalan, dizi[n - 1]);
}
}
```

Her bir rekürsif çağrıda, fonksiyon bir kez daha kendini çağırarak işlem yapar. Her bir rekürsif çağrıda dizinin boyutu bir azalır ve sonunda 1'e ulaşır. Toplam rekürsif çağrı sayısı, dizinin boyutu ile doğru orantılıdır. Yani toplam döngü adım sayısına bağlıdır. Dizi boyutuna n dersek döngü sayısı n olur. Sabit sayıda işlem olacağı için zaman karmaşıklığı $O(n)$ bulunur.