Dizideki en büyük sayıyı bulma probleminin

Kaba Kodu:

1-Öncelikle, dizinin ilk elemanını en büyük olarak varsayalım ve bunu bir değişkende saklayalım. Bu, dizinin herhangi bir elemanıyla karşılaştıracağımız başlangıç noktası olacak.
2-Dizinin diğer elemanlarını tek tek kontrol ederiz. Bu işlemi bir döngü içinde gerçekleştiririz.
3-Döngü içinde, her elemanı sırayla alırız ve en büyük olarak varsaydığımız değerle karşılaştırırız.
4-Eğer eleman, en büyük olarak varsaydığımız değerden daha büyükse, bu elemanı yeni en büyük değer olarak belirleriz.
5-Döngü, dizinin son elemanına kadar devam eder.
6-Döngü tamamlandığında, en büyük değer bulunmuş olur.
7-Son olarak, bulunan en büyük değeri döndürürüz.

Gerçek kodu:

```
using System;

class Program

{

   static void Main(string[] args)

   {

      // Örnek bir dizi oluşturalım

      int[] dizi = { 5, 8, 3, 12, 6, 9 ,16,20,25,10,29};


      // En büyük sayıyı bulmak için bir değişken oluşturalım

      int enBuyuk = dizi[0]; // İlk elemanı en büyük olarak kabul edelim

      // Dizi boyunca döngü başlatalım

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

      {

         if (dizi[i] > enBuyuk)

         {

            enBuyuk = dizi[i];

         }

      }

      // En büyük sayıyı ekrana yazdır

      Console.WriteLine("Dizideki en büyük sayı: " + enBuyuk);

      Console.ReadLine();
```

```
    }
}
```

Bu algoritma, dizideki en büyük sayıyı bulmak için dizideki her bir elemanı tek tek kontrol eder ve döngü içinde en büyük olanı güncelleyerek bulur.Bu işlem, dizi boyunca tekrarlanarak en büyük sayı bulana kadar devam eder. Bu algoritmanın karmaşıklığı $O(n)$dir, çünkü döngü dizinin tüm elemanlarını tek tek kontrol eder, burada n dizinin uzunluğunu temsil eder

İteratif **Kodu:**

```
class Program
{
// Verilen bir dizi içindeki en büyük sayıyı bulan bir metod oluşturdum
   static int EnBuyukSayi(int[] dizi)
   {
      int enBuyuk = dizi[0]; // İlk elemanı en büyük olarak kabul edelim
// Döngü oluşturup  dizinin tüm elemanlarını kontrol ediyoruz.
      for (int i = 1; i < dizi.Length; i++)
      {
// Eğer dizinin mevcut elemanı, şu ana kadar bulunan en büyük sayıdan daha büyükse,
 // en büyük sayıyı güncelliyoruz
         if (dizi[i] > enBuyuk)
         {
            enBuyuk = dizi[i];
         }
      }
      return enBuyuk;
   }
    static void Main(string[] args)
   {
// Örnek bir dizi oluşturuyoruz.
      int[] dizi = { 5, 8, 3, 12, 6, 9 };
  // EnBuyukSayi metodunu çağırarak en büyük sayıyı buluyoruz.
      int enBuyuk = EnBuyukSayi(dizi);
    // Bulunan en büyük sayıyı ekrana yazdırıyoruz.
      Console.WriteLine("Dizideki en büyük sayı (iteratif): " + enBuyuk);
      Console.ReadLine();
   }
```

}

Algoritma Analizi:

1. İteratif Yaklaşımın:
Bu iteratif algoritma, bir dizi boyunca bir döngü kullanarak her bir elemanı kontrol ediyoruz.
Döngü her elemanı tek tek kontrol ediyor, bu nedenle zaman karmaşıklığı $O(n)$ olur, buradaki n dizi uzunluğunu temsil eder.

Recursive (Özyinelemeli) Kodu:

```
using System;

class Program
{
    // Dizideki en büyük sayıyı özyinelemeli (recursive) olarak bulan bir metod tanımladık.
    static int EnBuyukSayiyiBul(int[] dizi, int n)
    {
        // Dizinin boyutu 1 ise, dizinin tek elemanı en büyük sayıdır, dolayısıyla bu elemanı döndürürüz.
        if (n == 1)
            return dizi[0];

        // "Math.Max" fonksiyonu, iki sayı arasından büyük olanı döndüren matematiksel bir fonksiyondur
        // Burada, dizinin son elemanı ile (dizi[n - 1]) bir önceki en büyük sayıyı karşılaştırıyoruz
        // en büyük olanı belirleriz ve recursive olarak bu işlemi devam ettiririz.
        return Math.Max(dizi[n - 1], EnBuyukSayiyiBul(dizi, n - 1));
    }

    static void Main(string[] args)
    {
        // Örnek bir dizi oluşturulur.
        int[] dizi = { 5, 8, 3, 12, 6, 9 };

        // EnBuyukSayiyiBul metodu çağırılarak en büyük sayı bulunur.
        // Metoda, dizi ve dizinin uzunluğu (n) parametre olarak verilir.
        int enBuyuk = EnBuyukSayiyiBul(dizi, dizi.Length);

        // Bulunan en büyük sayı konsola yazdırılır.
        Console.WriteLine("Dizideki en büyük sayı (recursive): " + enBuyuk);

        // Kullanıcının konsolu kapatmadan önce bir tuşa basmasını beklemek için kullanılır.
        Console.ReadLine();
    }
}
```

Algoritma Analizi:
Recursive (Özyinelemeli) Yaklaşım:
Bu recursive algoritma, her adımda dizinin son elemanıyla bir önceki en büyük sayıyı karşılaştırarak ilerler.
Algoritmanın her adımında, dizinin boyutu bir azaltılarak bir alt soruna dönüşür.
Bu nedenle, bu algoritmanın zaman karmaşıklığı da O(n) olur.

Ancak, recursive çağrılar için ekstra bellek tüketimi olur. Bu durum, recursive algoritmanın döngüye kıyasla bazı durumlarda daha fazla bellek tüketebileceği anlamına gelir.

** yani en büyük sayıyı bulma probleminde iteratif algoritma ile recursive algoritmanın, algoritma analizi yani karmaşıklığı aynı O(n) olmasınına rağmen recursive algoritmada bellek tüketimi iteratif algoritmaya göre daha fazladır…