ALGORİTMA ANALİZİ-ÖDEV 2

Kaba Kod

1. Başla
2. Onlu sayıyı temsil eden bir değişken oluştur ve değerini kullanıcıdan al
3. İkili sayıyı temsil eden bir değişken oluştur ve başlangıçta boş bir string olarak ayarla
4. Onlu sayıyı ikili sayıya dönüştürmek için aşağıdaki adımları uygula
   4.1 Onlu sayının 2'ye bölümünden kalanını al ve bu değeri ikili sayının başına ekle
   4.2 Onlu sayıyı 2'ye böl ve tam kısmını alarak onlu sayıyı güncelle
5. İkili sayıyı ekrana yazdır
6. Bitir

ALGORİTMANIN ANALİZİ

1. Özyinelemeli (Recursive) Algoritma

```
static string OndalikToIkili(int n)
  {
      if (n == 0)                                                    //n
          return "";                                                 //1
      else
          return OndalikToIkili(n / 2) + (n % 2).ToString();         // n
  }                                                                  //n+n+1 = 2n+1
```

- Bu yöntemde, her bir adımda ondalık sayı 2'ye bölünür ve kalan ikili sayının sonuna eklenir. Daha sonra, bu işlem rekürsif olarak ondalık sayı sıfır olana kadar devam eder.
- Rekürsif çağrılar, her bir adımda bir işlem yapar ve her adımda ondalık sayı yarıya indirilir.
- Rekürsif çağrıların sayısı, ondalık sayının büyüklüğüne bağlı olarak değişir. Her bir rekürsif çağrı, sabit zaman karmaşıklığına sahiptir.
- Bu yöntemin zaman karmaşıklığı, ondalık sayının bit sayısına ($\log_2(n)$) bağlı olarak O(log n) olur.

2. Yinelemeli (Iteratif) Algoritma

```
  Console.WriteLine("Ondalık sayıyı girin:");
      int ondalikSayi = int.Parse(Console.ReadLine());

      string ikiliSayi = "";                                  // 1
      while (ondalikSayi > 0)                                 // n
      {
          int kalan = ondalikSayi % 2;                        // 1
          ikiliSayi = kalan + ikiliSayi;                      // 1
          ondalikSayi /= 2;                                   // 1
      }                                                       //1+n+1+1+1 = n+4
```

- Her bir adımda, ondalık sayıyı 2'ye böler ve kalanı alır. Bu kalan, ikili sayının sonuna eklenir. Ardından, ondalık sayıyı bölmenin sonucu, bir sonraki adımda kullanılmak üzere güncellenir.

- Ondalık sayı sıfır olana kadar bu işlem devam eder.
- Bu yöntemde her adımda bir işlem yapıldığı için, ondalık sayının büyüklüğüne bağlı olarak döngünün çalışma zamanı değişir. Ondalık sayının bit sayısına ($\log_2(n)$) bağlı olarak zaman karmaşıklığı O(log n) olur.

ÖZET

- İteratif ve rekürsif çözümler, onlu bir sayının ikili dönüşümünü hesaplar.
- Her iki çözümün de zaman karmaşıklığı, onlu sayının ikili dönüşümündeki basamak sayısına bağlı olarak O(log n) düzeyindedir.
- İteratif çözüm, bir döngü kullanarak işlem yaparken, rekürsif çözüm bir fonksiyon aracılığıyla kendisini tekrar çağırır.
- Genel olarak, tercih edilen çözüm iteratiftir, çünkü daha az bellek tüketir.