

BLM210 PROGRAMLAMA LABORATUVARI II

PROJE 1

PROJE TESLİM TARİHİ: 18.03.2022

Projenin Amacı:

Bu projenin amacı, C dilinde kullanılan kodların zaman ve yer karmaşıklığının hesaplaması amaçlanmaktadır.

Zaman ve Hafıza Karmaşıklığı

Bilgisayar bilimleri ve benzeri bilimlerde istenilen soruya karşılık her zaman istenilen cevaplar en hızlı veya en kesin sonucu verecek Algoritma ve yöntemler olmayabilir. Bu durumun nedeni yazılan yöntem ve algoritmanın verimliliği ile ilgilidir. Algoritma ne kadar verimli çalışır ve istenilene ne kadar yakın olursa kodun performansı o kadar iyi olmaktadır.

Bu durumlar altında kullandığımız algoritmaların bize olan zaman ve hafıza maliyetlerini hesaplamak, bunlar hakkında bilgi sahibi olmak çok önemlidir. Bu iki terim aslında beraber algoritmanın verimliliğini belirtmektedir. İyi bir algoritmadan az yer kaplaması ve az zaman harcaması beklenir.

Problemi çözmek için algoritmanın harcadığı zamanın analizi **zaman karmaşıklığını**, gerekli belleğin analizi ise **yer(space) karmaşıklığının** hesabını gerektirir. Hesaplanan karmaşıklıkları analiz etmek ve bunları temsil etmek için, **Asimptotik Notasyon** kullanılmaktadır.

Big Oh Notasyonu-O(n): Bir algoritmanın çalışma zamanının veya yerinin üst **sınırını** temsil eder. Big O Notation'ın rolü, bir algoritmanın yürütülmesi için alabileceği en uzun süreyi veya yeri hesaplamaktır, yani bir algoritmanın en **kötü durumunu** hesaplamak için kullanılır. Aşağıda kullanılan Bazı Big O notasyon gösterimleri yer almaktadır.

sabitler $\Rightarrow O(1)$

Logaritmik $\Rightarrow O(\log n)$

Lineer $\Rightarrow O(n)$

Loglineer $\Rightarrow O(n \log n)$

Üstel $\Rightarrow O(n^a) \parallel O(a^n)$

Aşağıda örnekler yer almaktadır.

Örnek 1:

```
int toplam(int dizi[], int n)
{
    int x = 0
    for (int i = 0; i < n; i++)
    {
        x = x + dizi[i];
    }
    return x;
}
```

Zaman karmaşıklığı :O(n)

Yer karmaşıklığı: 4*n + 16

Örnek 2:

```
void printAllNumbersThenAllPairSums(int arr[], int size)
{
    for (int i = 0; i < size; i++)
    {
        printf("%d\n", arr[i]);
    }

    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            printf("%d\n", arr[i] + arr[j]);
        }
    }
}
```

Zaman karmaşıklığı: $O(n+n^2) = O(n^2)$

Sunum sırasında yukarıdaki örneklere benzer (for, while, switch case, iç içe döngü ve recursive fonksiyon) kod blokları verilecektir. Zaman ve yer karmaşıklığı hesaplanacak dosya içerisindeki kod, karmaşık yapılardan oluşacaktır. Tek döngüden oluşmayacaktır. Bu durum dikkate alınmalıdır.

Aşağıdaki linkten zaman ve yer karmaşıklığı ile ilgili bilgiler alabilirsiniz.

- <https://bilgisayarnot.blogspot.com/2020/05/algoritma-zaman-hafza-karmasiklik.html>
- <https://ibrahimkaya66.wordpress.com/2013/12/30/10-algoritma-analizi-algoritmalarinda-karmasiklik-ve-zaman-karmasikligi/comment-page-1/>
- <https://www.javatpoint.com/big-o-notation-in-c>

Projede Yapılması istenen isterler hakkında

- A) Dosya içerisinde kodu okunması ve Dosyanın içeriğinin kontrol edilmesi
- B) Dosyadan okunan kod Big O notasyonuna göre Zaman karmaşıklığının hesaplanması
- C) Dosyadan okunan kod Big O notasyonuna göre yer (Hafıza) karmaşıklığının hesaplanması
- D) Dosyadan okunan kod çalıştırıldığında geçen süresin hesaplanması

KISITLAR:

- Proje C dili kullanılarak geliştirilecektir.
- Geliştirildiğiniz projenin, tüm kod blokları için yer ve zaman karmaşıklığını doğru hesaplaması beklenmektedir. Sunum sırasında farklı kod blokları üzerinden kontrol edilecektir.
- Dosyadan okunan kod bloklarında birden fazla karmaşıklık türü bulunacaktır. Sizden toplam karmaşıklık hesaplaması beklenmektedir.
- Dosya içerisinde C programlama Kodların yer alacaktır.

Sunum: Proje sunumu E-Destek üzerine yükleyeceğiniz projenizdeki kodlar indirilerek alınacaktır. Bu nedenle E- Destek üzerine yükleyeceğiniz projenin doğruluğundan emin olunuz.

Açıklama:

Ödev Teslimi

- Ödevin raporu LaTeX veya Word kullanılarak yazılmalıdır.
- Proje en fazla iki kişilik gruplar halinde yapılacaktır. Her öğretim sadece kendi içerisinde grup olabilecektir.
- Rapor ieee formatında (önceki yıllarda verilen formatta) 4 sayfa, akış diyagramı veya yalancı kod içeren, özet, giriş, yöntem, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmalıdır.
- Dersin takibi projenin teslimi dahil edestek.kocaeli.edu.tr sistemi üzerinden yapılacaktır. edestek.kocaeli.edu.tr sitesinde belirtilen tarihten sonra getirilen projeler kabul edilmeyecektir.
- Proje ile ilgili sorular edestek.kocaeli.edu.tr sitesindeki forum üzerinden Arş.Gör. Emin Ölmez veya Arş.Gör. Dilara Görmez'e sorulabilir.
- Sunum tarihleri daha sonra duyurulacaktır.
- Sunum sırasında algoritma, geliştirdiğiniz kodun çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir.
- Kullandığınız herhangi bir satır kodu açıklamanız istenebilir.
- Göndermiş olduğunuz kodlar benzerlik testinden geçirilecektir. Benzerlik çıkması halinde aşağıdaki puanlama uygulanacaktır.
 - Gruplar arası Benzer çıkma (kod parçası alma) -projeden -(eksi)50 puan
 - Para karşılığı başka birine yaptıрма projeden -(eksi)100 puan
 - İnternette kodu (kod parçası) alma projeden -(eksi)50 puan