

INF103 ALGORITMA VE İLERİ BİLGİSAYAR PROGRAMLAMA

Sıralama Algoritmaları

Sıralama algoritmaları verilen bir dizinin elemanlarını belirli bir sıraya koyan algoritmalarlardır. Matematikte ve bilgisayar bilimlerinde yaygın olarak kullanırlar. Bir çok farklı sıralama algoritması vardır. Algoritmalar performanslarına ve yöntemlerine göre farklılık gösterirler.

Bu TP’de göreceğimiz 3 sıralama algoritması:

1. Kabarcık sıralaması (Bubble Sort)
2. Eklemeli sıralama (Insertion Sort)
3. Seçmeli sıralama (Selection Sort)

Uygulama

Site üzerinden iskelet dosyasını indirin ve projenize ekleyin. Dosya toplamda 6 tane program dosyası içeriyor.

- `sorting.c` ve `sorting.h` Sıralama algoritmalarının tanımlarını içerir.
- `main.c` tanımlı fonksiyonların çağrıları ve testlerini içerir. Her fonksiyon için test durumu vardır. Yeni test durumları ve dizi yapıları ekleyip deneyebilirsiniz.
- `applications.c` sıralama algoritmaları kullanılarak çözülen problemlerle ilgili fonksiyon tanımlarını içerir.
- `item.c` ve `item.h` Tanımlanan `item` veri yapısı ve sıralama fonksiyonlarını içerir.

Aşağıda verilen problemleri sırasıyla ilgili dosyada fonksiyonlara kodlayınız. Programınızı test etmek için `main.c` kaynak dosyasını kullanabilirsiniz. `main.c` de tanımlı bazı test durumları fonksiyonların doğru çalışıp çalışmadığını anlamanız için eklenmiştir.

Sorular

1. Sıralama algoritmaları ile ilgili problemleri çözmeden önce verilen 3 tanesi kodlayarak başlayacağız. `sorting.c` de tanımlanacak 3 sıralama algoritması vardır.
 - Eklemeli Sıralama: Verilen dizide sırasıyla her eleman için uygun yeri bularak sıralama yapar.

- Seçmeli Sıralama: Verilen dizi içerisinde en küçük elemanı bulup, yer değiştirerek sıralama yapar. Dizi tamamen sıralanana kadar bu işlem devam eder.
- Kabarcık Sıralaması: Verilen dizi içerisinde yan yana iki eleman arasında karşılaştırma yaparak sıralar. Herhangi bir yer değişikliğine gerek olmayana kadar sıralama devam eder.

Yukarıda genel açıklamaları verilen sıralama algoritmalarını `sorting.c` içerisine kodlayınız. Fonksiyonlar verilen `int` tipindeki diziyi sıralar ve sıralı dizi olarak geri döndürür.

2. Yazdığınız fonksiyonları `main.c` içerisinde çağırıp, çalışma sürelerini karşılaştırınız. Bu işlemi `time.h` da tanımlı `clock()` fonksiyonunu kullanarak yapabilirsiniz. Hangi fonksiyon daha hızlı çalıştı? Nedenini açıklayınız. `sorting.c` içerisine yorum satırı olarak ekleyebilirsiniz.

Fonksiyonların sıralı ve sırasız dizilerde performansları ne olur? Hangisi sıralı dizi verildiğinde daha hızlıdır? Açıklayınız.

3. `applications.c` Sıralama algoritmaları farklı problem çözümlerinde kullanılabilir. Aşağıda sırasıyla verilen problemler için fonksiyonlar yazınız. Her problem için karmaşıklığı göz önüne alınız. Daha önce tanımladığınız sıralama algoritmalarını kullanabilirsiniz.

- (a) Parametre olarak verilen **sırasız** bir dizide, $x + y$ değerini maksimum yapan ikiliyi bulunuz. Fonksiyon karmaşıklığı nedir?
- (b) Parametre olarak verilen **sıralı** bir dizide, $x + y$ değerini maksimum yapan ikiliyi bulunuz. Fonksiyon karmaşıklığı nedir?
- (c) **Mod(mode)** değeri bir dizide en çok geçen sayıyı ifade eder. Örneğin 1, 2, 3, 4, 7, 2, 9 dizisinde mod değeri 2dir. Sırasız olarak verilen dizideki mod değerini bulan fonksiyonu kodlayınız.
- (d) **(BONUS)** Sırasız olarak verilen bir dizide en küçük k tane elemanı ne kadar sürede bulabilirsiniz? Algoritmanızı `minimum_k_elements()` fonksiyonu içine kodlayınız. Örnek olarak verilen 3, 7, 4, 9, 10 dizisi için en küçük 2 eleman 3, 4 geri dönmelidir.
- (e) **(BONUS)** Bir dizi aynı elemandan birden fazla içerebilir. Aynı eleman içeren bir dizi verildiğinde sadece farklı elemanlardan yeni bir dizi oluşturmak için bir fonksiyon yazınız. `remove_duplications` fonksiyonu sonuçta oluşan diziyi `dest` içerisinde geri döndürür.

1 Cevaplarınızı göndermeden önce dikkat etmeniz gerekenler

- Fonsiyon ve değişken isimlerinde Türkçe karakter kullanmayın.
- Projenizin adının **TP10_AD_SOYAD** olduğundan emin olun. Projeyi arşivleyip internet platformuna yükleyin.
- Oluşturduğunuz yeni dosya isimlerinde Türkçe karakter kullanmayın.

- Gönderdiğiniz program içerisinde derleyici hatası olmadığına emin olunuz.
- Yaptığınız işlemleri anlatmak için yorum yazabilirsiniz.