

2021-2022 Güz Yarıyılı

Algoritma Analizi

Ödev – 1

Ders Yürütücüleri

Doç. Dr. M. Elif KARSLIGİL

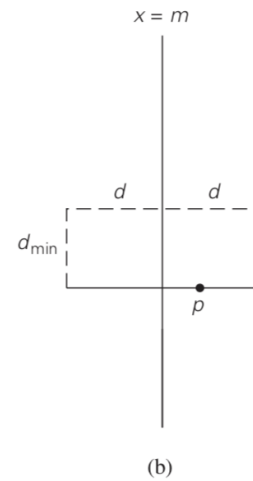
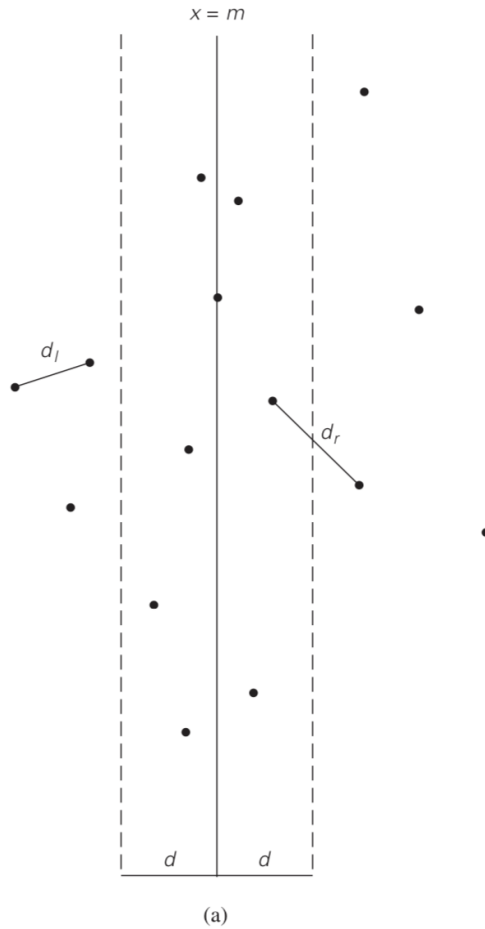
Doç. Dr. M. Amaç GÜVENSAN

Konu : Divide – and - Conquer Algorithms

Problem: Closest-Pair Problem

Bu ödevde kartezyen düzleminde verilen n adet nokta arasında birbirine en yakın 2 noktanın Divide-and-Conquer yaklaşımı ile bulunması istenmektedir. Problemden tüm noktaların birbirinden farklı koordinatlarda yer aldığı kabul edilmelidir.

1. Öncelikle “sample.txt” dosyasında sırasıyla verilen x ve y koordinatları okunmalıdır.
2. Noktalar x koordinatına göre artan sıralı ile sıralanmalıdır.
3. Aşağıdaki Şeklin a şıkında gösterildiği gibi noktalar x koordinatlarının medyan değeri kullanılarak ikiye ayrılmalıdır.



4. Her iki bölgede (P_1 , P_r) kalan noktalar arasındaki mesafeler hesaplanarak her iki bölgedeki en yakın ikililerin uzaklıkları (d_l, d_r) hesaplanır ve içlerinden küçük olanı seçilir ($d = \min \{d_l, d_r\}$)
5. d uzaklığı bulunmasına rağmen ikililer her zaman aynı bölgede olmak zorunda değildir. En yakın ikilinin noktaları sol ve sağ bölgede yer alabilir. Bu noktaların kontrol edilmesi için her iki bölgeyi ayıran $x=m$ doğrusuna dik çizilmiş $2d$ yakınlığında yer alan noktalara bakmak yeterli olacaktır. Daha yakın bir ikili bulunursa d_{\min} güncellenir.
6. $P(x,y)$ 'nin bu listedeki noktalardan biri olduğu farz edildiğinde $p'(x',y')$ noktası ancak yukarıdaki şekilde belirtilen b şıkında olduğu gibi dikdörtgen üzerinde yer alırsa d_{\min} 'den daha kısa mesafede bir nokta bulunabilir. (Bu dikdörtgen üzerinden en fazla 6 noktanın bulunabileceği detaylı bir analiz sonucunda ortaya konmuştur.) İlgili nokta adayları kontrol edilmelidir.

Not: $2 \leq n \leq 3$ iken problem brute – force yaklaşım ile çözümlenmelidir. $n > 3$ olduğu koşullarda noktaların listesi P_1 ve P_r olmak üzere ikiye bölünmeli ve 2.adımdan başlanarak algoritma rekürsif olarak implemente edilmelidir.

Ödev Teslimi ile ilgili önemli bilgiler:

Aşağıda verilen bütün bilgileri içeren tek bir doküman hazırlayarak **16.11.2021 saat 23:59'a** kadar online.yildiz.edu.tr üzerinden **HW1_OgrenciNumarasi.rar** dosyasını yükleyiniz.

- Ödev kontrolleriniz **17.11.2021 Çarşamba günü** dersin laboratuvar saatinde yapılacaktır. Detaylı bilgi için **Arş. Grv. Fuat ÖGME'nin** avesis.yildiz.edu.tr adresindeki sayfasını takip ediniz.

Teslim Edilecekler:

1. Yaptığınız çalışmayı yöntem ve uygulama bölümlerinden oluşan bir raporda anlatınız. **Yöntem** bölümünde problemi ve çözüm için önerdiğiniz yöntemi kısaca anlatıp rekürsif fonksiyonunuzun durma ve rekürsif olarak çağırılma koşullarını belirtip algoritmanın karmaşıklığını hesaplayınız.
2. **Uygulama** bölümünde sample.txt üzerindeki noktalar için elde ettiğiniz sonuçları açıklamalarınızla birlikte adım adım gösteriniz.
3. Algoritmanızın **C** dilinde programını hazırlayarak dokümana da ekleyiniz.
4. **Teslim Edilecekler**
 - a. HW#_OgrenciNumarasi.rar (Örn: HW1_15011001.rar)
 - i. OgrenciNumarasi.pdf (Örn: 15011001.pdf)
 - ii. OgrenciNumarasi.c (Örn: 15011001.c)
 - iii. sample.txt

Değerlendirme

Bilgisayar bilimleri anabilim dalı araştırma görevlilerinin [avesis](#) sayfalarında yer alan Ödev Teslim Kuralları başlığındaki kurallara uyulması gerekmektedir.

Algoritma Tasarımı ve Programın Çalışması: (%80)

1. Ödev, istenilen işlerin tamamını yerine getirmelidir.
2. Gereksiz kontrollerden ve işlemlerden arınmış bir tasarım yapılmalıdır.
3. Program hatasız çalışmalıdır.
4. Programın çalışması sırasında, konuyu bilmeyen kişilerin rahatlıkla anlayabilmesi için, giriş ve çıkışlarda mesajlarla bilgi verilmelidir.

Rapor Dokümantasyonu: (%20)

1. Raporun kapak sayfasında, dersin adı, öğrencinin ad, soyad ve numarası, ödev konusu bilgileri yer almalıdır.
2. Kaynak kodda değişken deklarasyonu yapılırken her değişken tek satırda tanımlanmalı, tanımın yanına değişkenin ne için kullanılacağı açıklama olarak yazılmalıdır.
3. Değişken isimleri anlamlı olmalıdır.
4. Her fonksiyonun yaptığı iş, parametreleri ve dönüş değeri açıklanmalıdır.
5. Gerekli yerlerde açıklama satırları ile kodda yapılan işlemler açıklanmalıdır.
6. Gereksiz kod tekrarı olmamalıdır.
7. Kaynak kodun formatı düzgün, okunabilir ve takip edilebilir olmalıdır.