



SAKARYA
ÜNİVERSİTESİ

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ (İÖ)

VERİ YAPILARI DERSİ

2. Ödev Raporu

ÖĞRENCİ ADI.....: Serdar Arıcı

ÖĞRENCİ NUMARASI...: G191210020

DERS GRUBU.....: B grubu

Bu ödevde benden istenen Noktalar.txt adında bir dosyadan her bir satırı tek tek okumak ve satırda belirtilen koordinatları üçerli şekilde okuyarak nokta oluşturmak. Daha sonra DogruKuyruğu isimli bir sınıf oluşturarak oluşturduğumuz noktaları bu sınıfa öncelikle kuyruk şeklinde eklemek. Önceliği ise noktaların orjine uzaklığına göre hesaplamak. Sonrasında oluşan DogruKuyruğu nesnelerini AVL ağacına eklemek. Bu ağaca ekleme işlemini DogruKuyruğu nesnesinin oluşturacağı doğruların toplam uzunluğuna göre yapmak. Son olarak da ağacı postorder olarak dolaşmak ve her düğümdeki DogruKuyrugunu öncelikli biçimde almak ve kuyruktaki noktaların orjine uzaklık değerlerini ekrana çıkarmak.

Bu ödevi yaparken öncelikli kuyruk konusunda kuyruğa ekleme yapmayı ve kuyruktaki verileri belirttiğimiz önceliğe göre almayı öğrendim. Ayrıca AVL ağacı konusunda eksiklerimi tamamlamaya çalıştım. Ağaca ekeme işlemlerini ve ekledikten sonra ağaçta kaydırma yaparak ağacı düzenleme işlemlerini öğrendim. Ağacın yüksekliğini hesaplamayı ve ağaçtaki verileri yazdırmak için yapılan ağacı dolaşma işlemlerini öğrendim.

Bu ödevde öncelikle DogruKuyruğu sınıfında kullanmak için bir düğüm sınıfı oluşturdum. Düğüme öncelik değerini ve x,y,z koordinatlarını tanımladım. Sonra DogruKuyruğu sınıfını oluşturdum. Bu sınıfın kurucu ve yıkıcı fonksiyonlarını tanımladım. Ekleme, silme, orjine uzaklık ve en yakın düğümü getirme fonksiyonlarını burada tanımladım. Daha sonra AVL ağacında kullanmak için DugumAVL adında bir sınıf daha oluşturdum. Burada ağacın düğümlerinde kullanılacak olan sağ, sol, veri ve yükseklik değerlerini tanımladım. Sonrasında AVL sınıfını oluşturdum. Burada ağaca ekleme, ekledikten sonra sağa ve sola kaydırma, ağacın yüksekliğini bulma, ağacı postorder olarak dolaşma ve ağaçtaki verileri ekrana yazdırma fonksiyonlarını tanımladım. Ödevde bir de DosyaOku sınıfı oluşturdum. Bu sınıfta Noktalar.txt dosyasındaki veriler okunuyor. Her satır okunduğunda bir DogruKuyruğu nesnesi oluşuyor. Satırdaki boşluklara göre her üç koordinatta bir nokta oluşturuluyor. Dosyada fazladan boşluk verilerse onlar da kontrol edilerek düzeltiliyor. Satırdaki koordinat sayısı bittiğinde ise DogruKuyruğu AVL ağacına ekleniyor ve diğer satıra geçiliyor. Son olarak AVL ağacının yazdır fonksiyonu çalıştırılıyor ve dosya okuma işlemi tamamlanıyor. Ödevde bir de Test sınıfı oluşturdum. Burada dosyaOku fonksiyonu çalıştırılıyor.

Ödevde eksik bıraktığım bir yer olduğunu düşünmüyorum.

AVL ağacına ekleme yaptıktan sonra ağacı düzeltmek için yapılan kaydırma işlemlerinde zorlandım.