

SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
BSM207
VERİ YAPILARI

ÖDEV NO: 2

ÖĞRENCİ ADI: Yasemin MUZİROĞLU

ÖĞRENCİ NO: B191210054

DERS GRUBU: C

ÖĞRETİM TÜRÜ: 1. Öğretim

DR. ÖĞR. ÜYESİ Kayhan AYAR

ÖDEV RAPORU

Ödevin Konusu: Dosyadan Veri Okuma, Okunan Veriyi İşleme, Öncelikli Kuyruk Oluşturma, Kuyruğa Eleman Ekleme, AVL Ağacı oluşturma, AVL Ağacına veri ekleme ve postorder olarak dolaşım

Ödevde İstenilenler:

- ✓ Noktalar.txt dosyasından alınan koordinat bilgilerine göre, noktanın orijine uzaklığı belirlenecek ve orijine yakınlığına göre noktalar öncelikli kuyruğa eklenecektir.
- ✓ Noktalar.txt dosyasının her bir satırı okunduğunda bu satırdaki noktaları barındıracak bir DogruKuyruğu nesnesi oluşturulmalıdır.
- ✓ Bu nesneler oluşturulduktan sonra bir AVL ağacına yerleştirilmelidir.
- ✓ Arama ağacına yerleştirme DogruKuyruğu nesnesinin oluşturacağı doğruların toplam uzunluğuna göre yapılmalıdır.
- ✓ Program çalıştırıldığında AVL ağacı postorder olarak dolaşılmalı ve her düğümdeki DogruKuyruğu içeriği öncelik sırasına göre ekrana çıkartılmalıdır. Kuyruktaki noktaların koordinatları yerine orijine olan uzaklık değeri ekrana çıkartılmalıdır.

Ödevdeki Sınıflar ve Ödevin Çalışma Prensipleri:

Nokta.cpp : Dosyada verilen noktaları temsil eden sınıfın kaynak dosyasıdır. Noktaların x,y ve z koordinatlarını ve orijine uzaklığını tutar. Kurucu fonksiyonunun içerisinde dosyadan okuduğu x,y ve z koordinatlarını sınıfın üye değişkenlerine atar ve noktanın orijine uzaklığını hesaplar. Operatörü aşırı yükleyerek iki nokta arasındaki mesafeyi hesaplar.

Dugum.cpp : Kuyruğa eklenecek verilerin düğüm sınıfıdır. Parametre olarak aldığı veriyi düğümün verisine atar ve varsayılan olarak sonraki pointer'ına sıfır değerini atar.

DogruKuyruğu.cpp : Doğru kuyruğunun kurucu, yıkıcı ve öncelikli veri ekleyen fonksiyonu bulunur. Kurucu fonksiyon bir doğru kuyruğu nesnesi oluşturulduğunda varsayılan olarak ilk ve son göstericilerine 0 değerini atar. Yıkıcı fonksiyon bir döngü yardımıyla kuyruğun verilerini siler. Ekleme yapan fonksiyon ise iki parametre alır. Bu parametrelerden biri ekleme yapacağı kuyruktur, diğeri ise ekleyeceği nokta tipindeki veridir. Noktanın orijine olan uzaklığına bağlı olarak sıralı olarak veriyi kuyruğa ekler.

AVLDugum.cpp : AVL Ağacına eklenecek verilerin düğüm sınıfıdır. Kurucu fonksiyonu parametre olarak aldığı veriyi düğümün içerisine atar ve düğümün sağ ve sol göstericilerine varsayılan olarak 0 değerini atar.

AVLAgaci.cpp : AVL ağacının kaynak dosyasında AVL ağacının bozulması durumunda çağrılan sağa döndür ve sola döndür fonksiyonları , parametre olarak verilen düğümün yüksekliğini bulan yükseklik fonksiyonu, verinin kök düğümdeki veriden küçük veya büyük olmasına bağlı olarak AVL ağacına veri ekleyen ve geriye düğüm döndüren ekle fonksiyonu ve ödevde istenildiği gibi AVL ağacını postorder şekilde dolaşmayı sağlayan postorder fonksiyonu vardır.

main.cpp : Main fonksiyonunun içerisine girildiğinde önce AVL ağacının bir nesnesi oluşturulur ve Noktalar.txt dosyası, while döngüsü içerisinde satır satır okunmak üzere açılır. Ödevde istenilen neredeyse her şey bu while döngüsü içerisinde gerçekleştirilir. Döngü içerisinde Doğru Kuyruğu için de bir nesne oluşturulur. Yeni bir while döngüsü içerisinde dosya okunurken veriler sırayla koordinat değişkenine oradan da x,y ve z koordinatlarına atılır. X, y ve z koordinatlarına veriler alındıktan sonra kuyruğun ekle fonksiyonu bu koordinatları parametre olarak alacak şekilde çağrılır ve bu çağrıdan dönen değer kuyruğa atanır. Bu işlem satır sonuna kadar bu şekilde devam eder. Bir satırın sonuna gelindiğinde içerideki döngüden çıkılır ve bu satırdan oluşan noktalar arasındaki toplam mesafe hesaplanır. Hesaplanan bu mesafe AVL ağacına yerleştirilmek üzere AVL ağacının ekle fonksiyonuna parametre olarak verilir. Tüm bu işlemlerden sonra diğer satıra geçmek üzere döngünün bir sonraki adımı başlar. Dosya içerisindeki bütün satırlar bittiğinde dıştaki while döngüsü sona erer. Son durumda dosyadaki satır sayısı kadar veri, AVL ağacına eklenmiştir. Son olarak postorder dolaşımla AVL ağacındaki verileri ekrana çıkartacak olan postorder fonksiyonu çağrılır ve Noktalar.txt dosyası kapatılır.