

Deniz ŞİMŞEK g201210042-2C

Ödevde istenen satırdaki noktaların oluşturacağı toplam uzunluğu bulmamız ve bu uzunluklara göre bir AVL ağacı oluşturmamız, oluşturduğumuz ağacı postorder ile okumamız ve okunan satırdaki noktaların orijine yakınlık değerlerini yakından uzağa ekrana yazdırmaktı.

Ödevde ilk olarak nokta sınıfı oluşturdum. Bu sınıf noktanın x,y,z değerlerini, noktanın bulunduğu satırdaki nokta sayısını, noktanın orijine uzaklığını ve sonraki noktanın adresini tutuyor.

Sonra DogruKuyruğu sınıfı oluşturdum. Bu sınıfta nokta ekleyen, noktanın orijine uzaklığını hesaplayan, noktaların aralarındaki uzunlukları hesaplayan yine noktalara bulundukları satırları ve bulundukları satırlardaki nokta sayısını atayan fonksiyon bulunuyor. Aynı zamanda kuyruğun en yüksek değerlerini bulan ve en yüksek değerlerini çıkaran fonksiyonlar da bu sınıfta bulunuyor. Noktaların birbirleriyle olan uzaklıklarını topladıktan sonra bunu her satırın en sonundaki noktanın uzunluk değerine atayıp satırdaki diğer noktaların uzunluk değerlerine 0 atadım. Böylece karşılaştırma yaparken sadece toplam uzunlukları karşılaştırmış oldum.

Noktaların orijinlere uzaklıklarını sıralamak için farklı bir OncelikliNokta sınıfı ve OncelikliKuyruk sınıfı oluşturdum. DogruKuyruğu nesnesindeki noktaları orijine uzaklığına göre satır bilgileriyle beraber öncelikli kuyruğa atadım. OncelikliKuyruk sınıfının yazdır fonksiyonunda satır numaralarına göre noktaların orijine yakınlığını ekrana yazdırdım.

AVL ağacı içinse Dugum sınıfı ve AVLAgaci sınıflarını oluşturdum. Dugum sınıfında dugum constructor fonksiyonu ekle metodu çağırıldığında parametre olarak nokta sınıfından nesne alıyor. Noktanın uzunluk, orijine uzaklık, satır, nokta sayısı bilgilerini alıyor. Sol, sag adındaki işaretçilerine 0 değerini atıyor.

AVLAgaci sınıfında noktaEkle fonksiyonu parametre olarak nokta ve dugum işaretçileri alıyor. Eğer noktanın toplam uzunluğu 0 ise ekleme yapmıyor. Toplam uzunluk değeri sıfırdan farklıysa ekleme yapıyor. Ekleme yapılırken ağacın dengesi kontrol ediliyor. Dengesizlik var ise sagaDondur ya da solaDondur fonksiyonları çağırılıyor. Postorder fonksiyonu da dugum ve önceliklikuyruk sınıflarından işaretçileri parametreler olarak alıyor. Postorder fonksiyonun içerisinde ağaçtan okunan düğümün satır numarasını vererek öncelikli kuyruğun yazdır fonksiyonunu çağırıyorum. Böylece ağaçtan okunan satırdaki noktaların orijine uzaklığını ekrana bastırıyorum.

Main.cpp içerisinde while döngüsü ile dosyayı okudum. Dosyadan alınan her sayıyı bir diziye atadım ve okunan her satırdan sonra artan bir sayaç tuttum. Bu sayaç 3 olduğunda kuyruğa yeni nokta ekledim. Nokta ekledikten sonra satır, nokta sayısı, orijine uzaklık atamalarını yaptım. Satır bittikten sonra ise son düğüme satırdaki noktaların toplam uzunluğunu atadım ve ağaca bu uzunluğu ekledim. Dosya okuma bittikten sonra nokta sayısına göre olan bir döngüyle DogruKuyrugundaki orijine en yakın değer sıralamasına göre değerleri OncelikliKuyruğa atadım. Bu atama da bittikten sonra postorder ile ağacı ekrana yazdırdım.