

Ödevde istenilen, noktalar.txt dosyasından bir satır dizi koordinat okunması, bu noktaların 3'erli şekilde ayrılıp x, y ve z koordinatlarına sahip noktaların oluşturulması; bu noktaların bir doğru kuyruğu nesnesine atılması, her satır için bir doğru kuyruğu oluşturması ve bu doğru kuyruklarının noktalardan oluşan doğru parçalarının toplam uzunluklarına göre AVL ağacına eklenmesi, çıktı olarak da bu ağacı postorder şekilde gezerek her doğru kuyruğunda bulunan noktaların orijine olan uzaklıklarının en yakından en uzağa şekilde yazılmasıdır.

Ödevi yaparken öğrendiklerim, ağaç yapısının kullanılması ve uygulamaları, AVL ağaç yapısının hangi fonksiyonlarla kendini dengelediği, kuyruk yapısı, öncelikli kuyruk yapısının önceliğinin belirlenme ve uygulamalarıdır.

Ödevde bir Nokta sınıfı oluşturarak başladım. Her noktanın bir x, y ve z koordinatları olması gerektiğinden x, y ve z değerlerini tuttum. Bu değerleri kurucu fonksiyon içinde atadım ve her noktanın orijine uzaklığı özgün olacağından bir orijine olan uzaklık bulan fonksiyon ekledim

İkinci olarak DogruKuyruğu sınıfı oluşturdum. Bu sınıf kuyruk yapısının üstüne öncelik belirleyen fonksiyonlardan oluşuyor. Öncelik belirleme şekli ise o sınıfta bulunan nokta nesnelere ait orijine uzaklık fonksiyonu. Bu sınıf bir kuyruk yapısı olduğundan ilk ve son Nokta pointerları bulunuyor. Ayrıca burada bir satırdaki noktaların tamamı tutulduğundan dolayı toplam uzunluk fonksiyonu da bulunuyor. Bu fonksiyon bir döngü içerisinde noktaların tamamını kapsayacak şekilde sırası ile 2'li noktaları parametre olarak alan bir uzunluk döndürme fonksiyonu çağırıp bu uzunlukları topluyor. Çağırdığı fonksiyon ise iki nokta arasındaki uzunluk formülü ile uzunluğu döndürüyor. Yapıcı fonksiyonda tutulan pointerlar sıfırlanıyor. Yıkıcı fonksiyonda ise oluşturulan nokta nesneleri siliniyor.

AVLAgaci sınıfı ise ekleme, silme, toplam koordinat sayısını döndürme, yükseklik bulma ve postorder gezme fonksiyonları var. Ekleme fonksiyonu parametre olarak noktalar.txt den satır alıp bu satırı DogruKuyruğu sınıfına nokta oluşturacak şekilde dönüştürüyor. Toplam koordinat sayısı döndürme, satırı stringstream fonksiyonu ile dönerek toplam koordinat sayısını getiriyor. Toplam koordinat sayısı ise nokta nesnelerinin oluşturulmasında kullanılıyor. Postorder gezme fonksiyonunda çıktı olarak DogruKuyruğu sınıfına ait öncelik fonksiyonları kullanılıyor. Bu öncelik orijine göre olan uzaklık kullanılarak belirleniyor. Silme fonksiyonu ise postorder gezme fonksiyonu ile benzer yapıda. Yaprak durumundaki kuyruk nesnelere giderek bunları siliyor. Kuyruk nesnesinin yıkıcı fonksiyonu ise nokta nesnelerini bastan sona gezerek onları siliyor.

Ödevde en zorlandığım kısım AVL ağacı yapısına ekleme yaparken kuyruk nesnesi oluşturup buna okunan satırı vermek, okunan satırı ise uygun parçalara bölüp kuyruk üzerinden oluşturulmak istenen kadar nokta nesnesi oluşturmaktı. Bu kısmı, satırı ham olarak alıp AVL ağacının ekleme fonksiyonunda istenilen formata getirerek yaptım. Bu format ile kuyruk nesnesi oluşturdum. Kuyruk nesnesi de formattaki koordinatları kullanarak nokta nesnelerini oluşturdu.