

İSİM: Emirhan

SOYİSİM: KAYA

NUMARA: G201210010

ÖDEV NO: 2

GRUP: 2A

### İstenilen

Noktalar.txt dosyasındaki satırlarda bulunan üçerli şekilde x, y, z koordinatlarına sahip noktaları orijine uzaklığını hesaplayıp ayırarak bir nokta sınıfına atmak sonrasında ise bu noktaları bir öncelikli doğru kuyruğu sınıfına atarak sıralamak ve atarken noktalar arasındaki uzaklıkları toplayarak noktaların kendi arasında oluşturduğu doğruların toplam uzunluk verisini tutma, sonrasında ise doğru kuyruklarının sahip olduğu toplam uzunluk verisine göre bir AVL ağacına yerleştirilmesi ve bu işlemlerin Noktalar.txt dosyasındaki her bir satır için tekrarlanması. Sonrasında ise öncelikli doğru kuyruklarının yerleştirildiği AVL ağacının postorder dolaşarak ağaç içerisindeki öncelikli doğru kuyruklarının içerisindeki noktaların orijine olan uzaklıklarını yazdırmak.

### Öğrendiklerim

Öncelikli kuyruk yapısını ve AVL ağacının yapısını öğrendim. AVL ağacında nasıl dengeleme yapılır öğrendim.

### Ödevde Yaptıklarım

Nokta adında bir sınıf oluşturdum. İçerisinde noktalar.txt dosyasında bulunan x, y, z koordinatları ve constructorda hesaplanan orijine uzaklık değeri mevcut.

NodeNokta adında bir düğüm sınıfı oluşturdum. İçerisinde nokta sınıfını taşımakta ve next pointeri bulundurmakta.

DogruKuyruğu adında bir öncelikli kuyruk sınıfı oluşturdum. İçerisinde noktalar arasındaki doğru parçalarının toplamı olan Toplamuzunluk değeri ve noktalar.txt dosyasındaki satırlar mevcut. İçerisinde NodeNokta sınıfının pointerini bulundurmakta ve kuyruk bu düğümlerle sıralanmakta.

Node adında bir düğüm sınıfı oluşturdum. Bu düğüm sınıfı ise AVL ağacında bulunacak ve right, left pointerları bulunacak içerisinde ise DogruKuyruğu sınıfını bulunduracak şekilde ayarlandı.

AVL adında bir AVL ağacı sınıfı oluşturdum. İçerisine eklenen Node sınıflarını dengeleyerek ekleyen bir sınıf şeklinde ayarladım.

Main.cpp içerisinde ise dosya okuma işlemlerini gerçekleştirdim. AVL ağacını oluşturdum ve doğru kuyruklarına okuduğum satırları gönderdim.

Ödevde okuduğum satırlardaki noktaları nokta sınıfında tuttum. Tuttuğum nokta sınıfını bir nodeNokta sınıfında pointer aracılığıyla tuttum. NodeNokta sınıfını kullanarak bu noktaları öncelikli doğru kuyruğu sınıfımda sıraladım. Doğru kuyruğunu ise Node adında düğüm sınıfında pointer aracılığıyla tuttum. Node sınıflarını ise içerisindeki doğru kuyruğunun içindeki noktaların arasındaki toplam uzaklıklara bakarak AVL ağacına ekledim. AVL ağacına ekledikten sonra dengeleme işlemlerini yaptırдым. En son postorder olarak ekrana yazdırma işleminde doğru kuyruklarını ekrana yazdırırken içerisindeki nokta sınıflarının orijine olan uzaklıklarını ekrana yazdırdım. Böylece ödevi başarı ile tamamlamış oldum.

### Eksik Bıraktıklarım

Ödevde herhangi bir eksik bıraktığım yer bulunmamaktadır.

### Zorlandığım Kısımlar

Aslında okuduğunuz vakitte komik bulacağınız bir kısımda saatlerce uğraştım. AVL ağacında dengeleme işlemlerini düzgün bir şekilde yaptım. Postorder doğru bir şekilde dolaşıyordu ancak ekran çıktısı örnek çıktıdakinden farklıydı. Ekran yazılan orijine uzaklık değerleri aynıydı ancak bu satırların sıraları farklıydı. Her şeyi doğru yaptığımı düşünüyordum, defalarca kez kodlarımı kontrol ettim, debug yapa yapa en sonunda sorunun toplam uzunluk değerini hesaplayan fonksiyonda olduğunu fark ettim ancak fonksiyonun tek işlevi 2 nokta arası uzaklık formülünü uygulamaktı ve formül doğru yazılmıştı. En sonunda ekrana postorder dolaşırken hem toplam uzunluk değerini hem de noktaların x, y, z değerlerini yazdırmaya karar verdim. Ekran çıktısında yazan nokta koordinatları bile doğrudu ancak yaptığım bu işlem ince bir şeyi fark etmemi sağladı. Noktalar fonksiyona öncelikli bir şekilde giriyormuş... Gelen doğru kuyruğunu peekMax() fonksiyonu yerine peek() fonksiyonu ile dolaşınca sorunun çözülmüş oldu.