Ödevi yaparken öncelikle öncelikli kuyruk ve avl ağacını anlamam gerekti. Kayhan hocanın videoları ve kendi yaptığım deneme yanılmalar ile yapıyı anladıktan sonra, ödevi yapmaya başladım. Ödevi yaparken İngilizce yazmaya karar verdim. Önceki ödevde ı harfi kullandığım için sorun çıkmıştı bu yüzden direk İngilizce yazmak istedim. Daha sonrasında ise normal kuyruk yapısını yaptım. Burası zor değildi zor olan kısım kuyruğa üç değer (x,y,z) atayabilmekti. Burada bir array kullandım ve bir pointer oluşturup bu arrayi göstermesini sağladım. Bu sayede sadece tek bir değer döndürerek bütün x,y,z değerlerine erişimim olacaktı. Ekleme kısmını buraya göre ayarladıktan sonra önceliği ayarlamaya geldi sıra. Lakin bundan önce önceliğin şartı olan orjine uzaklığın fonksiyonunu yazdım. Zaten kolay bir fonksiyon parametre olarak array pointerını alıyor ve oradan x, y, z değerlerine erişiyor daha sonra matematik işlemi ile uzaklığı döndürüyor. Öncelik fonksiyonu hocamızın da izniyle videodakinin hemen hemen aynısı. Üzerinde biraz değişiklik yaparak istediğimi verecek şekilde güncelledim. Diğer fonksiyonları daha sonrasında ihtiyaç doğrultusunda ekledim ama şimdi anlatmak daha iyi olacak gibi. Bir tane kuyruğun boş olup olmadığını kontrol eden fonksiyonum var. Bir tane öncelikli değerin orjine uzaklığını veren fonksiyonum var. Öncelik ayrı fonksiyonda olduğundan yapması kolaydı. Silme fonksiyonu için ise öncelikli olanı çağırıyor ve gerekli kontrollerden sonra siliyor. Kuyruğu silme ise boş olmadığı sürece sil şeklinde basit ve etkili bir kod ile sağlanıyor.

Sırada ağacımız var. Ağaçta ekleme, ağacı ortadan kaldırma ve postorder dolaşma kodları var. Tabikide ekleme fonksiyonu için döndürme fonksiyonlarımız ve yükseklik fonksiyonum var. Burada ekstrem bir durum yok normal avl ağacındaki fonksiyonlarla aynı diyebiliriz. Ekleme yaparken kuyruk değerini pointer olarak almadığımda verdiği hatayı anlarken biraz zorlandım ama en sıkıntılı kısmı toplam uzunluğu hesaplamak oldu çünkü ben hesabı orjine göre olan sırayla bakılacağını düşünerek yazdım ama verilen dökümandaki değerlerden farklı çıkınca hatamı fark etmekte epey zorlandım. Ekleme fonksiyonu hem düğüm hem kuyruk hem de uzunluğu alıyor. Veriler kuyruğa yerleştirildikten sonra uzunluğu hesaplamam imkansız olacağı için veri eklenirken düğümde toplam uzunluk tutuluyor ve döndürmeler ona göre oluyor. Ekleme fonksiyonu recursive olduğu için başta biraz zorlandım çünkü aklımda canlandıramıyordum. Sonrasında kağıt kalem alıp çizince daha iyi anladım ve her aşamayı o şekilde test ettim. Bana göre ödevin en zor kısmına gelirsek eğer dosyayı okurken epeyce zorlandım. En başta ifstream hata verdi ve getline fonksiyonu düzgün çalıştığı halde hatalı gösterdiği için visial studio code onun sebebini bulmakla da zor zamanlar geçirdim. Üstelik daha sonra önce verileri tek tek almayı yazdım o düzgün çalışmayınca üçerli almaya başladım verileri. O yüzden en baştan yazmak zorunda kaldım. Daha sonrasında da stringstream düzgün çalışmadığı için hata aldım ve onunla uğraşmak zorunda kaldım. O sorunu da çözdükten sonra ufak bir yer hatası yüzünden uğraşmak zorunda kaldım. Kısacası bayağı uğraştırıcı bir yerdi. Ağacı silme kısmına gelirsek recursive void bir postorder yapısından farkı yok. Avl ağacının düğümünün silinme kısmına da kuyruğun silme fonksiyonunu çağırması için ayarladım. Bu sayede ağaç silinmek istediğinde kuyruk yapısı da silinerek geriye heapte hiç açık bırakmamış oluyorum. Lakin bunu yaptığımda root kısmını 0 yapmadığımdan hata aldım.

Daha da kötüsü hatayı çözmek için ne yapmam gerektiğini bulmam uzun sürdü. Çözüm olarak ise silmeyi çağırdıktan sonra root=0 yazmam oldu.