

## SAKARYA ÜNİVERSİTESİ – BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

### VERİ YAPILARI DERSİ 2. ÖDEV RAPORU

Ödevde başlarken nasıl yapmam gerektiğini hakkında düşündüm. Ödev verildiğinde ikili ağaçları biliyordum ancak AVL ağacını daha işlememiştik. Araştırmalarım sonunca ikili ağaca ek bir özellik AVL ağacında dengeleme-döndürme işlemleri olmasıydı. Ağacın yapısını gözümün önünde canlandırdım . Bir ağaç nesnesi oluşturulacak ve düğümlerden oluşacak. Bu düğümler veri tutacak veriye ek olarak sol ve sağ düğümlerin adreslerini tutan iki adet pointer gerecekti. Bu ön plandan sonra ödevi yapmaya başladım.

Aslında ön planlamada metin işlemlerine yer bile vermemiştim ancak beklemediğim ödevde harcadığım zamanın büyük bir bölümünü metin işlemlerine harcadım. Metin işlemlerinde ilk önce kelimeAyir() diye bir metod yaptım bu metodun içinde Word diye bir değişken tanımladım. Yazdığım metod gelen string ifadeleri karakter karakter okuyor ve boşluk görene kadar Word e ekliyor boşluk gördüğü zaman Word ün bittiğini anlıyor ve yeni kelimeyi okumayı başlıyor. Ancak bu metod gerek boşluklarla gerek sıfırlarla hata verdiği için en son halinde bunu kullanmadım ancak kodun içinde bırakıldı. Diğer yöntemde dosya ifstream ile okundu , getline() metodu kullanıldı . Bu metod satır satır txt dosyamızı okumaya yarıyor. Ancak buradaki sorun ise okuma işlemini string şeklinde yapıyor. Bizim sayı işlemlerimiz için int veri tepine dönüşmesi gerekti. Bunun içinde “stoi” kullanıldı. (string to int’in kısaltılmış hali olduğunu gördüğümde şaşırmıştım). Ve en sonunda okunan ve integere çevrilen sayıları kuyruklara aktarıyoruz.

Uzun süren metin işlemlerinden sonra sayıları Radix sort ile sıralamamız gerekiyordu . Bunun içinde hocamızın Radix dosyasını kullandım. -Radix sort kuyruklar ile yapılan sıralama algoritmasıdır-

Ağac ve Kuyruk classları için de hocamızın githubdaki kodlarından yararlandım. Okunan metin int e çevrilerek kuyruklara ekleniyor ve Radix işlemlerinden sonra sıralı bir şekilde sıralanmış oluyor. Ve ağaçta oluşturulan düğüme ekleniyor. sumQueue ile kuyruktaki elemanların toplamalarını elde ediyoruz ve işlemleri buna göre yapıyoruz.

Enqueue ile kuyruğa eleman eklerken , kuyruktan çıkarma işlemleri dequeue ile yapılır.

Denge işlemleri ise yüksekliğe (height) göre yapılmakta , AVL ağacında yükseklik farkı 1 den fazla olamaz.Elemanlar yerleştikten ve dengelendikten sonra , ardından ödevde göre silinme işlemleri yapıldı ve preorder metodu çağrılarak istenilen şekilde ekrana yazıldı.

Dosyalar başlık ve kaynak olarak ayrıldı.

Ödevde en zorlandığım kısım dosya işlemleri çünkü bu konuya çok hakim değildim ve hayli hata aldım ama en sonunda çalıştırmayı başatabildim.

Eksik kısım olarak bazı sayılar test edildiğinde string okuma işlemleri hatalı olabiliyor ve char const ile başlıyan bir hata veriyor ve bazen saniyeler süren makefile okuma işlemi oluyor.

Dosyalama işlemleri istenildiği gibi yapılmış ve program mingw ile derlenmiştir.

Test ettiğim Sayilar.txt dosyasındaki sayılar ve okunduktan sonra programın çıktısı . En aşağıda ise Visualgo.net ten programımın doğruluğunu test etmek için oluşturulan AVL ağacı görünmekte. Eski Kök düğüm (98) bir aşağı gittiği için içindeki maximum 40 değeri silinmiş olduğu görünüyor.

Sayilar.txt - Not Defteri

Dosya	Düzenle	Görünüm
30 40 28		
10 20 36		
52 10 15		
32 23 14		
11 52 24		
2 2 1		
4 2 1		

Okunan metindeki Satir Sayisi : 7  
\*\*\*\*\*Preorder Yazim\*\*\*\*\*  
77: 52  
66: 20  
5: 1 2 2  
7: 1 2 4  
69: 14 23 32  
98: 28 30  
87: 11 24 52

Misafir Öğrenci

Yavuz Selim ŞAHİN / ~~Yavuz Selim ŞAHİN~~