B141210019- B141210087 RAPOR

ÖDEVDE BİZDEN İSTENEN

Bu ödevde bizden geçen dönemden gelen c++ Dosyalama konusunu da kullanarak bu dönem

öğrendiğimiz dinamik bellek yönetimini uygulamamız istenmiş. Aynı zamanda Huffman ağacıyla bağlı listeyi harmanlayarak dosya sıkıştırmayı ve huffman algoritmasının yararının algılanması istenmiş. Burada bize öğretilmek istenen huffman algoritması sayesinde yüksek boyutlu dosyaları huffman algoritmasıyla birleştirererek daha düşük boyutlu dosyaları elde etmiş olduk. Bu ödevde bizden bunu öğrenmemiz istenmiş.

ÖĞRENDİKLERİMİZ

Bu ödeve Hufmann algoritamasını kullanmayı onu bağlı listeyle harmanlamayı ve bu algoritmanın bize sağladığı yararları öğrendik. Bu yarar örneğin çok büyük boyutlu bir txt dosyası düşünelim Bu txt dosyasını huffman ağacına taşıyıp bunun kodlarını ve harflerini alarak ve örneğin bit.bat dosyasına sıkıştırarak aynı dosyadan çok daha az yer kapladığını öğrendik.

ÖDEVDE YAPTIKLARIMIZ

Öncelikle Bağlı liste ile harmanlamamız gerektiğinden bahsetmiştik. Bağlı liste ile harmanlamak için bağlı liste kütüphanemizi oluşturduk ve kurucu fonksiyonunu ve operatorleri yazdık.Daha sonrahuffman ağacını oluşturmak için gerekli kodlar yazıldı. Burada üstüne gitmemiz gereken olay 3 tane dosya kullanmamız gerektiğiydi. Bu dosyalar "deneme.txt","tablo.txt","bit.bat" tı. Zor olan kısmı huffman sınıfının objesi direk olarak tabloyu ve bitleri üzerinde taşıyordu. Ama bizim yapmamız gereken ise huffman objesini yazdırmadan bitleri bit.bat dosyasına harf ve bit karşılıklarını da tablo.txt dosyasına yazdırmaktı. Bu sayede deneme.txt dosyası ile diğer iki dosyanın toplam alanını karşılaştırıp ağacı doğru oluşturup oluşturmadığımıza bakmamız gerekiyordu. Bu yüzden harfleri bulduğu yeri ve kodları oluşturduğu yeri debug kullanarak izleyerek bunları bulduk. O esnada kodları binary.bat dosyasına sıkıştırdık ve harf karşılaklarıyla birlikte kodlarıda tablo.txt dosyasına yazdırdık. En son yaptığımız işlemler sonucunda deneme.txt dosyası ile diğer iki dosyanın toplam boyutlarını karşılaştırdık. Sonuç olarak Huffman algoritmasını doğru şekilde uyguladığımız ortaya çıktı. Çünkü en son oluşan iki dosyanın boyutu deneme.txt dosyasının çok altındaydı.

EKSİK BIRAKTIĞIMIZ YERLER

-YOK

ZORLANDIĞIMIZ YERLER

Bu ödevde en çok zorlandığımız kısım, kütüphanelerin içiydi aslında. Ve mingw de derlerken zorlandık. Çünkü Sytem("pause") komutunu kullandık ve bu komut mingw de derlerken hataya sebep olduğunu farkettik. Bizde bu hatayı araştırmaya başladık. Araştırmalarımız sonucunda hatanın cstdlib kütüphanesini ekleyerek düzeltilebileceğini farkettik. Kütüphanelerde zorlanmamızın nedeniyse sürekli

char hatası veriyordu. Bu sorunuda metotlarımızı oluştururken sınıftan kalıtım alarak değilde direk sınıfın içinde metotlarımızı oluşturarak çözmüş bulunduk.