



SAKARYA ÜNİVERSİTESİ

Ad : Tarık

Soyad:Toplu

Numara : G231210010

Ders : Veri Yapıları

Ödev : 2

Benden İstenilen

Kısaca Binary Search Tree kullanarak verilen .txt'deki tüm satırları okumak ve ayrıca her satırı birbiriyle bağlamak için tek yönlü bağlı liste kullanmak. Bunlara ek olarak düğümlerin bilgilerini (sıradaki düğüm adresi, şu anki düğümün adresi, içindeki ağacın verisi) ekrana yazdırıp düğümler arasında dolaşabilmek ve ağacın görselleştirilmiş halini ekrana yazdırmak.

Öğrendiklerim

Bu ödevi yaparken özellikle BST veri yapısına daha çok hakim oldum önceki ödevdeki kodlarıma bu veri yapısını entegre ederek bazı fonksiyonları isteğime göre düzenleyip, ekleyip bazılarını kaldırdım. Ayrıca "cls" komutu ile terminalde istenilen şekilde yazdırmayı öğrendim. Bunu da dokümanlara eklenen videodan yararlanarak yaptım.

Yaptıklarım

1. Dosya Okuma: İlk olarak, ağaclar.txt dosyasındaki her bir satır okunarak, her satırdaki karakterler kullanılarak bir ikili arama ağacı oluşturdum. Dokümanda belirtilen gibi her karakterin ASCII değeri, ağaçta karşılaştırma yapmak için kullandım.
2. İkili Arama Ağaçları: Her satır için bir BST oluşturulmuş ve bu ağaçlar, bağlı listeye eklenmiştir.
3. Bağlı Liste Yapısı: Ağaçlar, tek yönlü bir bağlı liste içinde tutulmuştur. Liste düğümleri, her bir ağacın kök düğümünü ve bir sonraki düğümün adresini içermektedir. Kullanıcı, yalnızca ekranda 10 düğüm görebilmektedir.
4. Kullanıcı Etkileşimi: Kullanıcılar, 'a' ve 'd' tuşları ile liste üzerinde yukarı ve aşağı hareket edebilmekte, seçili düğümdeki ağacı görüntüleyebilmektedir. 'S' tuşuna basıldığında, seçili düğüm ve içindeki ağaç silinmektedir tüm düğümler silindiğinde programdan çıkılmaktadır. 'W' tuşu ile seçili ağacın ayna görüntüsü alınmaktadır.
5. Ağaçların Toplam Değeri: Her ağacın toplam değeri, sol düğümlerin değerinin iki katı ile sağ düğümlerin değerlerinin toplamı olarak hesaplanmaktadır.
6. Aynalama İşlemi: Kullanıcı, 'W' tuşuna basarak seçili ağacın ayna görüntüsünü alabilmektedir. Bu işlem, ağacın sol ve sağ alt ağaçlarının yer değiştirilmesi ile gerçekleştirilmiştir.
7. Bağlı Liste Yazdırma Fonksiyonu: Bu fonksiyon, bağlı listenin içeriğini ekrana yazdırmak için kullanılır. Fonksiyon, başlangıç indeksine göre 10 düğüm gösterecek şekilde tasarlanmıştır. Her düğüm için, düğümün adresi, bir sonraki düğümün adresi ve düğümde bulunan ağacın verisi (karakterler) yazdırılmaktadır. Bu sayede kullanıcı, bağlı listenin yapısını ve içeriğini kolayca görebilmektedir.
8. Ağaç Yazdırma: Bu fonksiyon, seçili düğümdeki ikili arama ağacını ekrana yazdırmak için kullanılır. Fonksiyon, ağacın yüksekliğini hesaplayarak, her seviyedeki düğümleri uygun boşluklarla hizalayarak ekrana yazdırmaktadır.

Zorlandığım Kısımlar

Ödevin en zorlayıcı kısmı, ağacı yazdırmak oldu. Ağacı yazdırma fonksiyonunu en az 5 6 kere silip baştan yazdım. İlk önce kâğıtta mantığını kavramaya çalışıp sonra kodlamaya çalıştım ama bir yerde hata yapıp durdum. En son algoritmayı daha iyi anlayınca fonksiyonu yazmayı başarabildim. Fakat ağacı levelorderla dolaşıp yazabilmeme rağmen her dal arasındaki bağıntıyı gösteren "." İşaretlerini tam olarak yazdıramadım. Ayrıca bağlı listeler arası dolaşım, silme ve aynalama işlemlerinde zorlandım silme işleminde hatalar çıktı, kod yanlış ağacı çizmeye başladı ama en son hepsini düzeltebildim.

Eksik Bıraktıklarım

Bu ödevde, sadece ağaç yazdırma kısmının "." Yazdırma kısmı eksik yapıldı. Bu hariç ağaç yazdırma

düzgün çalışmaktadır.