

Ödevin Amacı Bu ödevin amacı, ikili arama ağaçları (BST) ve bağlı listelerin uygulamasını öğretmektir.

Öğrendiklerim

- **İkili Arama Ağacı Uygulaması:** Her karakterin bir düğüm olduğu bir diziden BST oluşturmayı öğrendim.
- **Ağaç Gezinme:** Ağaçta çeşitli sıralarda (in-order, pre-order, post-order) gezinerek arama, ekleme ve silme işlemlerini nasıl yapacağımı öğrendim.
- **Ağaç Değeri Hesaplama:** Düğümlerin değerlerine dayalı olarak tüm ağacın değerini nasıl hesaplayacağımı öğrendim.
- **Görsel Ağaç Gösterimi:** Özel bir buffer sınıfı kullanarak BST'yi konsolda görsel olarak nasıl göstereceğimi öğrendim.
- **Gezinme için Kullanıcı Arayüzü:** BST listesinde gezinmeyi sağlayan, önceki veya sonraki ağaca geçme, mevcut ağacı silme ve aynalama komutlarını içeren bir kullanıcı arayüzü oluşturmayı öğrendim.

Yaptıklarım

- **Sınıf Dosyaları:** Her sınıfın kendi .hpp dosyası var ve fonksiyonlar .cpp dosyalarında uygulanmıştır.
- **Veri Okuma:** Program [agaclar.txt](#) dosyasından verileri okur ve her satırdan BST'ler oluşturur, bunları bir bağlı listede saklar.
- **Ağaç Manipülasyonu:** Ağaçları aynalama ve değerlerini hesaplama işlevlerini uyguladım.
- **Otomatik İşlemler:** Verileri okuma, ağaçları oluşturma ve gösterme işlemlerini otomatikleştirdim.

Eksik Bıraktığım Yerler

- **Hata Yönetimi:** Program, kenar durumları ve geçersiz girdilerle başa çıkmak için daha sağlam bir hata yönetimine ihtiyaç duyuyor.
- **Performans Optimizasyonu:** Programın mümkün olduğunca verimli çalışmasını sağlamak için daha fazla optimize edilebilir.

Zorlandığım Kısımlar

- **Ağaç Çıktılarının Hizalanması:** Ağaç çıktılarının düzgün hizalanmasını sağlamak, özellikle birden fazla seviyeye sahip daha büyük ağaçlar için zordu.
- **BST İnşası:** Her giriş satırından doğru şekilde BST inşa etmek ve BST özelliklerini korumak zorlayıcıydı.