



T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ

FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ

BÖLÜMÜ

VERİ YAPILARI ÖDEV RAPORU

**Verilen Sayılarla İkili Arama Ağaçları (BST) Oluşturma
ve Bu Ağaçlar Üzerinde Temel İşlemler Yapma**

**G211210065-Melih Tufan Atlı
G211210587-Ekrem Selçuk Çelik**

SAKARYA

Ağustos, 2023

Veri Yapıları Dersi

**Verilen Sayılarla İkili Arama Ağaçları (BST) Oluşturma
ve Bu Ağaçlar Üzerinde Temel İşlemler Yapma**

Melih Tufan Atlı -G211210065 / Ekrem Selçuk Çelik-G211210587 1A- Grubu

Özet

Bu proje, İkili Arama Ağacı (BST) veri yapısını anlamamızı sağlıyor. BST.h ve BST.cpp dosyaları, BST'nin temel işlevlerini içeriyor, özetleyecek olursak değer ekleme, yükseklik ve toplam değer hesaplama ve postorder sırasına göre ağacın yazdırılması. Node.h ve Node.cpp dosyaları, ağaçta kullanılan düğümleri tanımlıyor. Main.cpp dosyası, "numbers.txt" adlı bir dosyadan sayıları okuyor, her satır için en iyi BST'yi bulup (en yüksek yükseklik veya toplam değer kriterine göre) ve sonuçları postorder sırasına göre yazdırıyor. Proje, sayı dizileri üzerinde BST oluşturarak ve onları ekrana yazdırarak belirli kriterlere göre en uygun ağacı bulmaya odaklı bir proje.

© 2023 Sakarya Üniversitesi.

Bu rapor bizim özgün çalışmamızdır. Faydalanmış olduğumuz kaynakları içerisinde belirttik. Herhangi bir kopya işleminde sorumluluk bize aittir.

BST, Node, Veri Yapıları, Ağaç

1. GELİŞTİRİLEN YAZILIM

Ödev, İkili Arama Ağacı (BST) üzerinde temel operasyonları gerçekleştirme yeteneğini geliştirmeyi amaçlıyor.

Ödev genel olarak en çok bizi bellek temizliğinde zorladı. Çünkü runtime error'u ile karşılaşp durduk. Ama 2-3 gün üstünde durunca onu da başarılı bir şekilde hallettiğimizi düşünüyorum. Aşağıda adım adım tam olarak ne yapıyor açıkladık:

BST ve Node Sınıflarının Oluşturduk. Node.h ve Node.cpp dosyaları, ağaç düğümlerini tanımlıyor. BST'nin temel işlevleri BST.h ve BST.cpp dosyalarında tanımlı olarak bulunuyor.

Rekürsif Fonksiyonlar: BST içindeki özel fonksiyonlar (yükseklik, toplam ve yazdırma) rekürsif olarak gerçekleştirildi. Rekürsif yapmamız, BST işlemlerinin doğası gereği, her düğümde aynı işlemleri gerçekleştirdiğimiz için yaptık.

Yığın Tabanlı BST Oluşturma: Projenin ilginç bir yönü, bir yığından BST oluşturma kısmı.

Satır İşleme ve En İyi Ağacın Seçilmesi: Main.cpp dosyası, verilen kriterlere (yükseklik veya toplam değer) göre en uygun ağacı seçmek için her satırı işler. Bu, farklı BST'ler arasında karşılaştırma yapma ve en iyi sonucu seçmeyi amaçlıyor.

Dosya okuma/yazma işlemlerinde pratik yapma olanağı da bulmuş olduk.

Kopya Yapıcı ve Atama Operatörü: Bu işlevler, derin kopya yapma ve objeler arasında veri aktarmayı amaçlıyor.

Bu ödevin bize verilme amaçları arasında bulunabilecek nedenler:

BST ve onunla ilgili algoritmaları anlama

Rekürsif fonksiyonların nasıl kullanılacağını pekiştirmek

Dosya işlemleri konusunda uzmanlaşmak

Farklı veri yapıları ve algoritmalar arasında bağlantı kurma yeteneği kazandırmak.

Son olarak BST'nin temel konseptlerini öğrenmenin yanı sıra, bu bilgiyi gerçek dünyadaki senaryolarla nasıl uygulayabileceğimizi görmüş olduk.

2. ÇIKTILAR

Ödev dosyasında verilen 2 satırlık kod ve 1000 satırlık kod örnekleri sıkıntısız bir şekilde çalışıyor. Bir kaç dosyada daha deneme yapıldı bir sorun görülmedi.

3. SONUÇ

Gerçek hayatta bazı noktalarda kullanılabilecek deneyimler kazandırıyor bu proje örnek verecek olursak:

Veritabanı sistemlerinde endeksleme yaparken ve dosya sistemlerinde hızlı bir şekilde veri ararken bu tür yapılar kullanılır.

Performansı optimize etmek için gerçek hayatta ağaçlar kullanılır.

BST'nin özellikle derin kopya yapıcıları ve yıkıcıları sayesinde dinamik bellek kullanımı, veri yapısı tasarımında ve gerçekleştirilmede önemlidir.

Referanslar

- [1] Stackoverflow üzerinde ağaçlar ve bellek temizliği konusunda bir kaç blogtan yararlandık
- [2] ChatGPT üzerinde bazı hata mesajları üzerinden yorumlar aldık. Özellikle Runtime error için kullandık.
- [3] Github üzerinde daha önceden yapılmış ağaçlar konusunda bir kaç proje inceledim.
- [4] Geçen dönemden kalma ağaç projesinden edindiğimiz bilgileri ve örnek kodları da kullandık. Yardımcı oldu.