**YZM 2124**

Veri Yapıları

2022-2023 Bahar Dönemi

**ÖDEV # RAPORU**

Teslim Tarihi: 02.06.2023

Öğrenci Adı: Taha KOÇER

Öğrenci Numarası: 212802077

1. Giriş

Bu proje yüksek seviyeli programlama dillerinde kullanılan Ağaç Veri yapısının arka planında çalışan algoritmaları anlamak amacıyla hazırlanmıştır.

Bu programda girdi olarak sehir\_plaka\_kodlari.txt doyasında bulunan plaka kodlarından 2 tanesi alınır, çıktı olarak ise ağaç veri yapısının görselleştirilmiş hali, derinliği, ağacın denge durumu, girilen plaka kodlarına ait şehir isimleri, bu şehirlerin ağaçtaki düzey farkı, preorder, inorder ve postorder sıralanışı alınır.

Kullanılan dosyalar

sehir\_plaka\_kodlari.txt: Ödevin içerisinde verilmiş 18 adet şehir ve plaka kodu bulunduran text dosyasıdır.

Kullanılan fonksiyonlar

newNode() : Yeni bir düğüm oluşturmak için kullanılan Node sınıfına ait yardımcı public fonksiyon.

insertHelper() : Ağaç veri yapısına düğüm eklemek için yardımcı private fonksiyon. insert() fonksiyonun içinde çağırılır.

removeHelper(): Ağaç veri yapısından düğüm silmek için kullanılan yardımcı private fonksiyon. remove() fonksiyonun içinde çağırılır

depthHelper() : Ağaç veri yapısının derinliğini integer cinsinden döndüren yardımcı private fonksiyon. depth() fonksiyonun içinde çağırılır

inorderTraversalHelper() : İnorder tarayarak ağacın düğümlerine erişmek için kullanılan yardımcı private fonksiyon. inorderTraversal() fonksiyonun içinde çağırılır.

preorderTraversalHelper() : Preorder tarayarak ağacın düğümlerine erişmek için kullanılan yardımcı private fonksiyon. preorderTraversal() fonksiyonun içinde çağırılır

postorderTraversalHelper() : Postorder tarayarak ağacın düğümlerine erişmek için kullanılan yardımcı private fonksiyon. postorderTraversal() fonksiyonun içinde çağırılır

printBinaryTree() : Ağacın dallarını görselleştirmek için kullanılan private fonksiyon. build() fonksiyonunun içinde çağırılır.

isBalanced() : Ağacın dengeli bir yapısı olup olmadığını boolean cinsinden döndüren private fonksiyon.

destroyTree() : Ağacın bütün düğümlerini bellekten temizlemek için kullanılan fonksiyon.

dataFromFile() : sehir\_plaka\_kodlari.txt dosyasındaki verileri iki boyutlu bir diziye çekmek için kullanılan fonksiyon.

1. Geliştirme ve Çalıştırma Ortamları

Visual Studio Code editörü üzerinde geliştirilip, console ekranı üzerinde çalıştırılmıştır.

1. Veri Yapıları ve Değişkenler

Ağaç veri yapısı oluşturulmuştur. Bu veri yapısı oluşturulurken başka bir veri yapısı kullanılmamıştır.

1. Program Akışı

Düğüm (Node) sınıfı veri yapısını oluşturmaya yarayan, içerisinde verileri ve kendinden sonraki sağ ile sol düğümün bellek adresini tutan değişkenler içerir.

BinarySearchTree sınıfı ağaç veri yapısının temelini oluşturan fonksiyonları içerisinde barındırır. İlk oluşturulan ve ağaca atılan düğüm kök (root) düğümü oluşturur. Bu düğüm ağaç veri yapısının baş kısmıdır. Son eklenen düğümden daha küçük veriler soluna daha büyük veriler ise düğümün sağına bağlanır. Böylece hiyerarşik bir veri yapısı oluşmuş olur. Bu yapı verilere erişimi daha hızlı ve ideal kılar.

1. Sonuç

Bu ödev yüksek seviyeli programlama dillerinde kullanılan veri yapılarının arka planındaki algoritmaların nasıl çalıştığını anlamama yardımcı oldu. Aynı zamanda pointer ve bellek yönetimi gibi konuları pekiştirmiş oldum. C++ syntax’ı ve sınıf kullanımı konusunda da belirli bir seviye katettim.