

/**

* @file Odev2_Rapor.pdf

* @description Uygulama açılışta Noktalar.txt dosyasını okur.

* x,y,z koordinatları verilen noktaları DogruKuyruğu sınıfına kaydeder.

* Aralarındaki uzunluğuda ölçerek toplar ve aynı sınıftaki tuzunluk değişkenine kaydeder.

* Her bir satırdan oluşan yeni DogruKuyruğu nesnelerini AVL ağacına ekler.

* Ve son olarak yeni oluşan bu ağacı postorder metodu ile ekrana gösterir.

* @course BSM 207 - Veri Yapıları - 1. Öğretim A grubu

* @assignment 2

* @date 25.12.2021

* @author Ali YAZICI ali.yazici3@ogr.sakarya.edu.tr

*/

Öncelikle nokta koordinatlarını kaydedebileceğim ve DogruKuyruğu sınıfı içerisinde düğüm olarak kullanabileceğim nokta sınıfını oluşturdum. Sonrasında DogruKuyruğu bağlı liste sınıfını tanımladım. Ve ilk olarak noktalar.txt dosyasını bu sınıftan oluşturduğum nesnelere kaydederek denemelere ve sonuçlara baktım.

Herbir satır için oluşturduğum ve önceliğini merkeze yakınlığa göre oluşturtduğum nesnelerden toplam uzunluklarıda hesaplatıp göstermeye başladım. Fakat “veri yapıları 2.ödev. örnek veri seti.pdf” sonuca uymadığını görünce satırdaki sıralamayı kullanarak toplam uzunluk hesaplanması gerektiği yazısını gördüm. Ve fonksiyonumu buna göre değiştirdim.

AVL ağacı için ise öncelikle dugum sınıfını oluşturdum ve DogruKuyruğu nesnesini veri olarak kullandım, Daha sonra bu dugumlari oluşturan AVL ağaç sınıfını ve ve fonksiyonlarını oluşturdum. Ders notlarını ve “C++ ile Veri yapıları” kitabımdan esinlenerek oluşturdum, fakat yine de denemelerde oluşan yanlışları gördükçe nasıl düzeltebileceğime dair çokça araştırma ve denemeler yapmam gerekti.

Özellikle ekle fonksiyonu ve AVL kriterlerini sağladığında sol çocuk ile ve sağ çocuk ile değiştir fonksiyonları üzerine çalışmalar yaptım. Zaman zaman NULL değerler ve yanlış kullanımlar nedeni ile runtime crash problemleri yaşadım. Bu problemleri de teker teker çözmeyi başardım.

Uzun çabalar ve uğraşlar sonucunda dökümanda istenilen sonuçları elde ettiğimi gördüm.

Dosyaların açıklamaları:

avl.hpp , avl.cpp : AVL ağacı sınıfı dosyaları

dkgezici.hpp , dkgezici.cpp: DogruKuyrugu sınıfı için gezici sınıfı ve fonksiyonları

dogrukuyrugu.hpp, dogrukuyrugu.cpp : DogruKuyrugu sınıfı dosyaları

dugum.hpp , dugum.cpp : AVL ağacı için dugum sınıfları

nokta.hpp , nokta.cpp: DogruKuyrugu sınıfına kaydedilen nokta dugumu sınıfı

Definitions.hpp: Bazı çok kullandığım metinleri bu dosyada tanımladım.

Main.cpp: Noktalar.txt dosyasını okuyup, DogruKuyrugu sınıfına kaydeden ve oluşan dogrukuyrugu nesnelerini AVL ağacına kaydedip, postorder olarak yazdırır.

```
D:\work\KTBU\SOFT\AVL_Tree>mingw32-make
g++ -I ./include -o ./lib/nokta.o -c ./src/nokta.cpp
g++ -I ./include -o ./lib/dkgezici.o -c ./src/dkgezici.cpp
g++ -I ./include -o ./lib/dogrukuyrugu.o -c ./src/dogrukuyrugu.cpp
g++ -I ./include -o ./lib/dugum.o -c ./src/dugum.cpp
g++ -I ./include -o ./lib/avl.o -c ./src/avl.cpp
g++ -I ./include -o ./bin/main ./lib/dkgezici.o ./lib/avl.o ./lib/dugum.o ./lib/nokta.o ./lib/dogrukuyrugu.o ./src/main.cpp
./bin/main
116 118
66 98
64 95 97 104 123
120 147
50 66 99 125
35 90 98 108 112 127
99 100 115 125 130
95 121 121 122
75 97 119 120
90 94 96 108 109 110 117 132 135
79 83 89 115 124 147
23 55 94 108 137
38 63 88 97 102 116 124
86 90 95 104 105 109 120 124 149
59 61 73 77 88 90 96 110
61 69 81 84 87 92 92 97 98 105 112
67 74 79 84 91 93 104 110 119 142 142
45 49 66 82 94 97 102 112 116 116 134
69 77 87 91 115 116 118 121 136
46 69 78 91 96 125
```