



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

VERİ YAPILARI DERSİ ÖDEV 2

B181210010 - Deniz Berfin Taştan

SAKARYA

Ağustos-2021

Veri Yapıları Dersi

Deniz Berfin Taştan

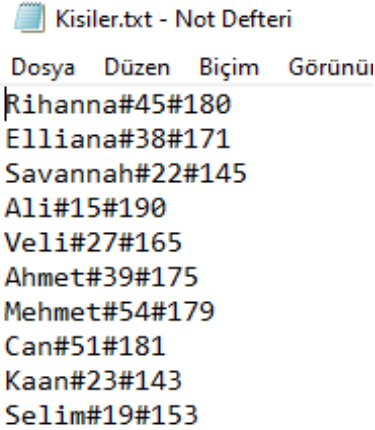
B181210010/1.Öğretim B grubu

1.GELİŞTİRİLEN YAZILIM

Geliştirdiğim yazılımda program önce “Kisiler.txt”dosyasını okuyarak “#” karakterine göre bölme işlemi yapıyor ve bunları isim,yas ve boy değişkenlerine atıyor. Boy ve yaş değişkenlerini string’ten integer’a çevirdikten sonra “AvlAgac.cpp” içerisinde tanımlanan **ekle()** fonksiyonu ile bu değişkenleri bir düğüme atıyor. Bu düğümler birbiri ardına eklenerek bir ağaç yapısı oluşturuluyor. **Ekle()** fonksiyonu her çalıştığında (her düğüm ataması yapıldığında) fonksiyon önceki düğümler ile yeni düğümü karşılaştırarak sağa veya sola atama yapıyor (Küçük ve eşit ise sola, büyük ise sağa). Yine aynı dosyanın içinde bulunan **yukseklik()** fonksiyonu düğümleri gezerek düğümler arası yüksekliği ölçüyor. **Dengele()** fonksiyonu ise **yukseklik()** fonksiyonunu kullanarak ağacın denge durumunu bir değişkene atıyor ve tekrar **ekle()** fonksiyonunda denge kıyaslaması yapıyor ve eğer şartlar sağlanmazsa yani ağaç dengede değilse düğümleri sağa veya sola kaydırarak (**sagaDon()** ve **solaDon()**) ağacı dengeli hale getiriyor. Yine “AvlAgac.cpp” içerisinde bulunan **siralala()** ve **yazdir()** fonksiyonları ise ağacı level order şekilde sıralayıp ekrana yazıyor.

“Dugum.hpp” içerisinde düğümlerin yüksekliğini tutmak için oluşturulan stack yapılarının doldurulması işlemi de “AvlAgac.cpp” içerisinde bulunan **stackDoldur()** fonksiyonu ile yapılıyor. **Ekle()** fonksiyonu her çalıştığında düğümün bulunduğu yükseklik “Stack.hpp” dosyası içerisinde tanımlanan **Push()** fonksiyonunu kullanarak stack’e ekleniyor. Eklenen düğüm yükseklikleri **Pop()** fonksiyonu kullanılarak stack’ten siliniyor fakat her silinen eleman ekrana yazılıyor. Bu sayede anlık düğüm yükseklikleri, ilgili düğümün yanına sırayla yazılmış oluyor.

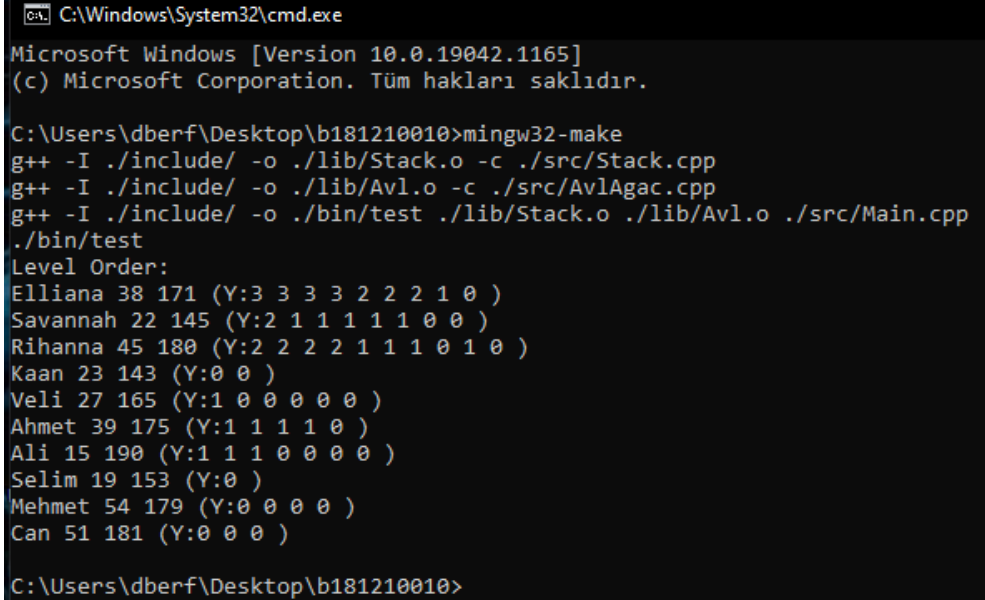
2.DENENEN TXT DOSYASI VE ÇIKTILAR



Kisiler.txt - Not Defteri

Dosya Düzen Biçim Görünü

Rihanna#45#180
Elliana#38#171
Savannah#22#145
Ali#15#190
Veli#27#165
Ahmet#39#175
Mehmet#54#179
Can#51#181
Kaan#23#143
Selim#19#153



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\dberf\Desktop\b181210010>mingw32-make
g++ -I ./include/ -o ./lib/Stack.o -c ./src/Stack.cpp
g++ -I ./include/ -o ./lib/Avl.o -c ./src/AvlAgac.cpp
g++ -I ./include/ -o ./bin/test ./lib/Stack.o ./lib/Avl.o ./src/Main.cpp
./bin/test
Level Order:
Elliana 38 171 (Y:3 3 3 3 2 2 2 1 0 )
Savannah 22 145 (Y:2 1 1 1 1 1 0 0 )
Rihanna 45 180 (Y:2 2 2 2 1 1 1 0 1 0 )
Kaan 23 143 (Y:0 0 )
Veli 27 165 (Y:1 0 0 0 0 0 )
Ahmet 39 175 (Y:1 1 1 1 0 )
Ali 15 190 (Y:1 1 1 0 0 0 0 )
Selim 19 153 (Y:0 )
Mehmet 54 179 (Y:0 0 0 0 )
Can 51 181 (Y:0 0 0 )

C:\Users\dberf\Desktop\b181210010>
```

Referanslar

1. <http://omercetin.com.tr/DERS/VY/Konu-10-Agaclar.pdf?i=1>
2. <https://www.journaldev.com/34929/level-order-traversal-in-a-binary-tree>
3. <https://www.geeksforgeeks.org/avl-tree-set-1-insertion/>
4. <https://www.codegrepper.com/code-examples/whatever/how+to+calculate+height+of+node+in+avl+tree+in+c%23>
5. Sabis Veri Yapıları Stack Dökümanları
6. C/C++ ile Veri Yapıları ve Çözümlü Uygulamalar/Muhammed Fatih Adak , Nejat Yumuşak/Seçkin Yayınevi/3.Baskı