



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

VERİ YAPILARI 2. ÖDEV RAPORU

NUMARA:.....B191210008

AD:.....Gamze

SOYAD:.....Ceylan

GRUP:.....1/B

Özet

Kisiler.txt dosyası okunacaktır, kişiler dosyadan okundukça AVL ağacına atılacaktır. AVL ağacı kişinin boyuna göre oluşturulacaktır. Her düğümde düğümün hareketini tutacak iki yığıt bulunacaktır. Yığıtlardan biri düğümün yüksekliğini, diğeri ise derinliğini tutacaktır.

1. GELİŞTİRİLEN YAZILIM

- İlk olarak Kisiler.txt dosyası okundu ve nesneler oluşturuldu. Kisiler okundukça AVL ağacında atıldı.
- Aynı zamanda kisi nesneleri heapte oluşturulduğu için çöp oluşmaması için kisiler dizi içerisinde tutuldu

```
Kisi *kisiler[sayac];
int i=0;
if(dosyaOku2.is_open()){
    while (getline(dosyaOku2, satir))
    {
        // dosyadan okunan satırlar isim, yas, boy olarak ayrılıyor
        stringstream ss(satir);
        getline(ss, isim, '#');
        getline(ss, yas, '#');
        getline(ss, boy, '#');

        // kisi nesneleri oluşturuluyor
        Kisi *kisi = new Kisi(isim, stoi(yas), stoi(boy));
        agac->Ekle(kisi); // kisiler agaca ekleniyor

        kisiler[i] = kisi; // kisi nesneleri dizi içine atılıyor

        i++;
    }
}
```

- Daha sonra kisiler bellekten temizlendi

```
// kisiler bellekten siliniyor
for (int i=0; i<sayac; i++){
    delete kisiler[i];
}
```

- Aynı zamanda kisi nesneleri ağaca eklendikten sonra yükseklik ve derinliklerinin tutulması için Stack sınıfı oluşturuldu. Kisi sınıfında nesneler için Stack türünde stack_yukseklik ve stack_derinlik değişkenleri tutuldu.

```
class Kisi{
public:
    Stack *stack_yukseklik;
    Stack *stack_derinlik;
    string isim;
    int yas;
    int boy;
```

- AVL sınıfında ise Kisiler eklendikçe düğümün yükseklik ve derinlikleri yığta ekleniyor

```
alt_dugum->kisi->stack_yukseklik->push(Yukseklk(alt_dugum));
alt_dugum->kisi->stack_derinlik->push(Yukseklk(kok) - alt_dugum->yukseklk);
```

- Ağac LevelOrder yazdırılırken Stack'ten her nesnenin yükseklik ve derinliği top() ile ekrana basıldıktan sonra pop() ile yığıt temizleniyor

```
void Stack::YazdirYukseklk(Stack *stack){
    cout << "Y(";
    for(int i=0; i<size; i++){
        if(i==size-1)
            cout << stack->top() << ")", ";";
        else
            cout << stack->top() << " ", ";";
        stack->pop();
    }
}

void Stack::YazdirDerinlik(Stack *stack){
    cout << "D(";
    for(int i=0; i<size; i++){
        if(i==size-1){
            cout << stack->top() << ")", ";";
        }
        else
            cout << stack->top() << " ", ";";
        stack->pop();
    }
}
```

2.ÇIKTILAR

Dosya İçeriği:

Dosya Düzen Biçim Görünüm Yardım

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 Berkay#32#163
Ayse#37#159 Fatma#41#191 Polat#63#187
Kerem#12#177 Samet#11#172 Erhan#44#169
Elif#32#164 Ceren#19#176 Salih#66#182
Halil#43#173 Seyma#33#166 Canberk#27#178
Mine#23#160 Kemal#35#174 Emre#22#192
Burak#25#170

Çıktı:

```
Elliana 38 171 Y(5,5,5,5,5,4,4,4,4,4,4,4,4,3,3,3,3,3,3,2,2,2,1,1,1,0,0),D(0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,1,-1,-1,2,0,0)
Berkay 32 163 Y(3,3,3,2,2,2,2,1,1,0,0),D(2,1,1,3,2,3,3,2,2,3,3)
Rihanna 45 180 Y(4,4,4,3,3,3,3,3,3,2,2,2,1,1,1,0,0),D(1,1,2,1,1,1,1,1,2,1,1,2,1,2,1,-1,-1)
Selim 19 153 Y(3,3,3,2,2,1,1,0,0),D(1,1,2,0,0,3,4,3,3)
Veli 27 165 Y(2,3,2,2,2,1,1,1,0,0),D(3,3,1,1,4,3,4,3,2,2)
Ahmet 39 175 Y(3,3,2,2,2,1,1,1,0,0),D(2,3,2,2,4,3,3,3,2,2)
Ali 15 190 Y(2,2,2,1,1,0,0),D(3,2,3,2,3,1,1)
Savannah 22 145 Y(2,2,1,1,0,0),D(1,2,2,2,1,1)
Ayse 37 159 Y(1,0,0),D(5,3,3)
Elif 32 164 Y(0,0),D(4,4)
Erhan 44 169 Y(1,1,0,0),D(4,5,4,4)
Halil 43 173 Y(1,1,1,0,0),D(3,3,5,4,4)
Kerem 12 177 Y(2,2,2,1,1,0,0),D(4,2,3,2,2,4,4)
Salih 66 182 Y(1,1,0,0),D(2,2,4,4)
Fatma 41 191 Y(1,0,0),D(5,3,3)
Kaan 23 143 Y(0,0),D(3,3)
Mine 23 160 Y(0,0),D(5,5)
Seyma 33 166 Y(0,0),D(4,4)
Burak 25 170 Y(0,0),D(5,5)
Samet 11 172 Y(1,0,0),D(5,4,4)
Kemal 35 174 Y(0,0),D(5,5)
Ceren 19 176 Y(0,0),D(4,4)
Mehmet 54 179 Y(1,1,0,0),D(5,4,2,2)
Can 51 181 Y(1,0,0),D(4,3,3)
Polat 63 187 Y(1,0,0),D(5,3,3)
Emre 22 192 Y(0,0),D(5,5)
Canberk 27 178 Y(0,0),D(4,4)
```

3. SONUÇ

- Verilen ödev ile birlikte ağaç veri yapısını oluşturmayı öğrendim ve mantığını kavradım.
- Fonksiyonları yazarken arka planda işlemlerin nasıl yapıldığını gördüm ve recursive fonksiyon mantığını anladım
- Ödevde tutulması istenilen yükseklik ve derinlikleri stack te tutarken zorlandım ve ödevin o kısmını tam olarak tamamlayamadım.