



**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**VERİ YAPILARI ÖDEV RAPORU**

**YIĞITLAR VE İKİLİ ARAMA AĞAÇLARI**

**G211210031-Metehan Dünder**

**G201210073-Muhammed Sefa Özdemir**

**SAKARYA**

**Ağustos, 2023**

# YIĞITLAR VE İKİLİ ARAMA AĞAÇLARI

Muhammed Sefa Özdemir - G201210073-1/A

Metehan Dünder - G211210031-1/A

**Özet :**Ödevde bizden istenilen 0-256 aralığında sayılardan oluşan bir metin dosyasını okutup okunan verileri yığıtlara ve yığıtlardan da ikili arama ağaçlarına belli kriterlere göre aktardıktan sonra ağaçlardaki verilerin ASCII karşılıklarını “postorder” sırada ekrana yazdıracak bir konsol uygulaması hazırlamak.

## 1. GELİŞTİRİLEN YAZILIM

### Veri Yapıları:

**Yığın (Stack):** Bir yığın, elemanların son giren ilk çıkan (LIFO) mantığıyla eklenip çıkarıldığı doğrusal bir veri yapısıdır. Bu projede, yığına ekleme (push), yığından çıkarma (pop), yığının en üstündeki öğeyi alma (top) gibi temel işlevler bulunmaktadır. Yığının boyutunu ve toplamını almak için ek işlevler de eklenmiştir.

**İkili Ağaç (BinaryTree):** Bu ağaçta, her düğümün en fazla iki alt düğümü vardır - bir sol ve bir sağ düğüm. Ağaçla ilgili temel işlevler arasında bir değer eklemek (insert), postorder ile ağacı gezinti yapmak (postOrder), ağacın yüksekliğini ve toplamını almak bulunmaktadır.

**Dinamik Dizi (DynamicArray):** Boyutu dinamik olarak değişebilen bir dizi. Eleman eklerken, dizi kapasitesi dolduğunda kapasite otomatik olarak iki katına çıkarılır.

"main.cpp" dosyasında, "Sayılar.txt" adında bir dosya okunur. Dosyadan her satır okunduğunda, o satırdaki sayılar bir yığında depolanır. Yığın oluşturma kuralı şudur: Eğer yığında eleman yoksa veya okunan sayı, yığının en üstündeki sayıdan küçük veya eşitse (veya tekse) bu sayı yığına eklenir. Aksi takdirde, yeni bir yığın oluşturulur ve bu sayı yeni yığına eklenir.

Oluşturulan her yığın için, yığındaki sayılar kullanılarak bir ikili ağaç oluşturulur. Yığından bir sayı çıkarıldığında, bu sayı ikili ağaca eklenir.

Son olarak, tüm ağaçlar arasında en yüksek olanı (veya aynı yükseklikte olanlar arasında en büyük toplama sahip olanı) bulunur ve bu ağacın düğümlerindeki değerlerin ASCII tablosundaki karşılıklarını ‘postorder’ sırada ekrana yazdırır.

Her satır okunduktan sonra, oluşturulan yığınlar ve ağaçlar bellekten silinir ve bir sonraki satır okunur.

## 2. ÇIKTILAR

```
C:\Users\bir_k\Desktop\masaüstü\VeriYapilari2>make
g++ -I ./include/ -o ./lib/main.o -c ./src/main.cpp
g++ -I ./include/ -o ./lib/BinaryTree.o -c ./src/BinaryTree.cpp
g++ -I ./include/ -o ./lib/DynamicArray.o -c ./src/DynamicArray.cpp
g++ -I ./include/ -o ./lib/Node.o -c ./src/Node.cpp
g++ -I ./include/ -o ./lib/Stack.o -c ./src/Stack.cpp
g++ -o ./bin/Program ./lib/main.o ./lib/BinaryTree.o ./lib/DynamicArray.o ./lib/Node.o ./lib/Stack.o
./bin/Program
♦ S A U ▼
#
C:\Users\bir_k\Desktop\masaüstü\VeriYapilari2>
```

## 3. KAYNAKLAR

<https://www.geeksforgeeks.org/stack-data-structure/>

<https://www.geeksforgeeks.org/binary-search-tree-data-structure/>

[https://www.youtube.com/playlist?list=PLh8R31K8J\\_9CITt-87z7y2F-uzOrDgIof](https://www.youtube.com/playlist?list=PLh8R31K8J_9CITt-87z7y2F-uzOrDgIof)