



**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**VERİ YAPILARI 2.ÖDEV RAPORU**

**İKİLİ ARAMA AĞAÇLARINDAN**

**ASCII KARAKTER DÖNÜŞÜMÜ**

**G211210029 - Aybüke Berfin SÜER**

**SAKARYA**

**Ağustos, 2023**

Veri Yapıları Dersi

# Aybüke Berfin SÜER

## G211210029 – 1.öğretim A grubu

### Özet

Bu yazılımda dosyadan satır satır okunan veriler. Yığıtlara aktarılacaktır. Her satırda belirlenen bazı koşullar vardır bu koşullar sonucunda satırda birden fazla yığıt oluşabilmektedir. Bir satırdaki tüm yığıtlar oluşturulduktan sonra Bu yığıtların içleri boşlatılıp birer ikili arama ağaçlarına yerleştirilecektir. Oluşan ikili arama ağaçlarından en büyük yüksekliğe sahip olan ağaç seçilecektir. Eğer yükseklikler eşitse elemanları toplamı en büyük olan ağaç seçilecektir. Toplamları da eşitse daha önce oluşturulan ikil arama ağacı seçilip bunun elemanlarını ascı karakter kodlamasıyla ekran yazdıracağız. Aynı işlemi her satır için bulacağız.

© 2023 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Her hangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: Yığıtlar, İkili Arama Ağaçları, Bellek Yönetimi

### 1. GELİŞTİRİLEN YAZILIM

Bu proje için öncelikle Yığınlar ve İkili Arama Ağaçları için genel sınıf tasarımları yaptım. Kurucu, yıkıcı fonksiyonları olmak üzere kendi sınıflarına ait kullanacağım özel fonksiyonlar da burada yazdım. Yığıt içinde ekleme, çıkarma, son elemanı getirme, boyut getirme şeklinde bir yığıtta olamsı gereken genel fonksiyonlar bulunmaktadır. İkili Arama ağacında ise projemizde asıl kullandığımız fonksiyonlar yer almaktadır.

Geliştirdiğim yazılımda satırlarda oluşanları kaybetmemek için bir bağlı liste tasarladım. Bu bağlı liste ile hem yığıtları hem de oluşan ikili arama ağaçlarını tutuyorum. Projemde satır okunmaya başlıyor bağlı listenin ilk ve son elemanı bellek problemi yaratmaması sebebiyle sıfırlanıyor. Daha sonra bir yığıt oluşturuluyor. Dosyadan yeni okunan sayı, çift ve o an işlem yapılan yığıttan çıkmak üzere olan (peek fonksiyonu) sayıdan büyük ise yeni bir tane daha yığıt oluşturulup ona eklenecektir. Aksi halde var olan yığıta eklenmeye devam edecektir. Bu sebeple bir satırda birden fazla yığıt oluşturulabiliyor. Oluşan yığıtları bağlı listede tutuyoruz. Daha sonra bu satırda oluşacak yığıtları bağlı listeden çekip her bir yığıtın içini boşaltıp ayrı ayrı ikili arama ağaçlarına ekledim. Oluşan ikili arama ağaçlarını yine bağlı listede tuttum.

Daha sonra bir satırda oluşan tüm ikili arama ağaçlarından her birini bağlı listeden çekip en yüksek olanı bulamıyoruz. Bunları BinarySearchList dosyasında yazdığım fonksiyonlarla yapıyoruz. İstenilen karşılaştırmalar sonucu en yüksek olan ağacı seçip yine aynı dosyadaki postorder sıralama ve ascı karakteriyle yazdırma fonksiyonunu çağırıyoruz.

Son olarak yeni satıra geçmeden oluşturduğumuz ve içinde yığınlarla ağaçları tuttuğumuz bağlı listeyi temizleyip çöp oluşmasını engelleyip yeni bir satıra geçiyoruz.

Dosya okuma işlemi tüm satırların sonunda tamamlanıyor ve programı sonlandırıyoruz.

## 2. ÖRNEK ÇIKTILAR

```
g++ -I "./include" -c ./src/main.cpp -o ./lib/main.o
g++ -I "./include" -c ./src/Stack.cpp -o ./lib/Stack.o
g++ -I "./include" -c ./src/BinarySearchTree.cpp -o ./lib/BinarySearchTree.o
g++ -I "./include" -c ./src/ListNode.cpp -o ./lib/ListNode.o
g++ ./lib/main.o ./lib/Stack.o ./lib/ListNode.o ./lib/BinarySearchTree.o -o ./bin/program
./bin/program
♦ A @ Q K F g Y i v s y S
A ♦ ♦ ♥ K W p k Q E @
+ u i Q
♥ ♦ x v I
l s j
@ ♥ B O E + k Q
@ H Q S c F @ w q l
@ n V +
@ E s o Q N ♦ ♦
+ ♦ R P K e u y d ♦
@ ♦ G M S K f u a ♥
V T e K y o
♦ ♦ @ Q w e E B
```

```
A ♦ r w X
♦ g L i +
@ @ S p o I
V Q w i L @
Y Q z c N E @
+ @ Y T K w ♦
@ I e d k x u p Q ♦
J D X a Q m i r o @
N B ♥ q s z u g
@ I j m i Y V ♦
P W T I ♦ u s @
♥ E M h r i U A
♦ ♦ @ A K O F T i s o x w u W ♦
@ ♦ K T Y ♦ ♥ t e a
@ ♥ h M q m @
B ♦ ♦ v U O
P L S F X U m g v s o a @
@ I ♦ ♥ U @
W J z y q e @
PS C:\Users\aybuk\OneDrive\Masaüstü\VeriYapilariveAlgoritma>
```

## KAYNAKLAR

- 1) <https://forum.donanimhaber.com/>
- 2) <https://stackoverflow.com/>
- 3) <https://www.mmotutkunlari.com/>
- 4) [https://www.youtube.com/@MF\\_A/videos](https://www.youtube.com/@MF_A/videos)
- 5) <https://www.youtube.com/c/KayhanAyar>
- 6) <https://emrecelen.com.tr/>