

## 2. Ödev [Tüm Şubeler İçin]

### Teslim Tarihi

6 Ağustos 2023 (Pazar günü saat 23:50'ye kadar.)

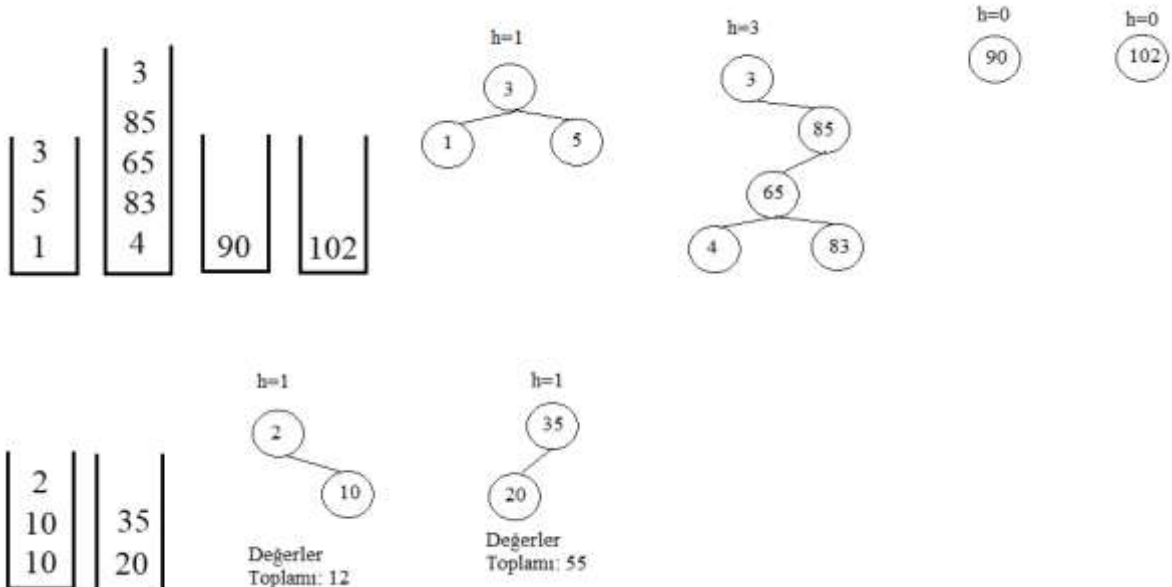
### Ödev İçeriği

MinGW C++ ortamında program yazılacaktır. Program çalıştığı gibi, 0-256 arası sayılarından oluşan ve sayıların arasını birer boşluğun ayırdığı Sayilar.txt dosyası okunacak. Her satır okunurken Yığita sayılar eklenecektir. Dosyadan yeni okunan sayı, **çift ve o an işlem yapılan Yığıttan çıkmak üzere olan sayıdan büyük ise** yeni bir Yığıta eklenecektir. Aksi halde var olan Yığıta eklenmeye devam edecektir. Bu şekilde dosyadaki bir satır okunduktan sonra her Yığıt boşaltılıp ayrı bir ikili arama ağacına eklenecektir. Ağaçta aynı değer varsa o değer ikili arama ağacına eklenmeyecektir. Bir satırda bulunan yığıt sayısı kadar ikili arama ağacı oluşturulmalıdır. Her satır için, oluşan ikili arama ağaçlarından **en büyük yüksekliğe sahip ikili arama ağacı seçilecektir**. Eğer yükseklikler eşit olursa düğüm değerleri toplamı büyük olan ağaç seçilecektir. Toplam değerleri de eşit olan ağaçlardan önce oluşturulan ağaç seçilecektir. Seçilen maksimum yüksekliğe sahip ikili arama ağacı **postorder okunup** sayısal değerlerin ASCII karakter karşılıkları ekrana yazılıp **ekranda 10 milisaniye beklenip** (sleep fonksiyonu), bir sonraki satır için yukarıdaki aynı işlemler uygulanıp tekrar karakterler ekrana yazılacaktır. Dosya okuma bittiği zaman program da sonlanacaktır.

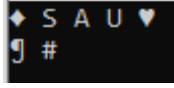
Örneğin dosya içeriği aşağıdaki gibi olsun

1 5 3 4 83 65 85 3 90 102
10 10 2 20 35

Oluşacak Yığıtlar ve **yığıtlardan eleman silinerek oluşturulacak ağaçlar aşağıdaki gibi olacaktır**. İlk satırda yüksekliği en fazla olan 3 yüksekliğine sahip olan ağaç olduğu için o ağaç seçilip ağaç postorder okunup sayıların ASCII karşılıkları karakter olarak aralarında bir boşluk ile ekrana yazılacaktır. İkinci satırdaki ağaçlar ise yükseklikleri eşit olduğu için düğümlerin değerinin toplamına bakılmakta ve toplamı 55 olan ağaç seçilmektedir.



Bu örneğin ekran çıktısı aşağıda verilmiştir.



Çok büyük dosyaları okuyabilecek şekilde program yazılmalıdır. Programda çöp oluşması puan kırılmasına neden olur. 1000 satırlık bir örnek dosya <https://raw.githubusercontent.com/mfadak/DataStructures/main/Veri.txt> bu adreste paylaşılmıştır. Bu dosya için program ekran çıktısı aşağıdaki videoda verilmiştir.

<https://youtu.be/-mgMgyqbSyE>

Yazdığınız kodun performansı önemlidir. Yukarıdaki 1000 satırlık örnek dosya için 25-30 saniye sürmesi normaldir. Bilgisayar donanımı da her ne kadar performansı etkilese de yukarıdaki örnek dosya eğer 1 dakika veya daha fazla sürüyorsa algoritma ve tasarlanan veri yapıları gözden geçirilmelidir.

**Düşük performanstan yüksek puan kırılacaktır.**

**Hiçbir şekilde Şablon (Generic) veri yapısı kullanılamaz. Her sınıfın başlık ve kaynak dosyası ayrı olmalı ve başlık dosyasında metot gövdesi bulunmamalıdır. Hazır veri yapısı kullanılamaz.**

**Önemli Not: Raporunuz detaylı olmalı ve kendi cümleleriniz olmalıdır. Kopya ödevler sıfır olarak değerlendirilecektir. SABİS şifreniz sizin sorumluluğunuz altındadır eğer arkadaşınız sizden habersiz ödevinizi alırsa bundan sizde sorumlu tutulur ve sıfır alırsınız.**

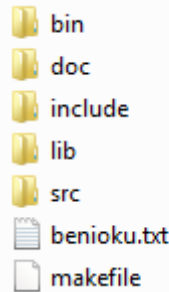
## Teslim Formatı

Klasör Hiyerarşisi

- src Klasörü (Kaynak dosyalarınızın bulunduğu klasör)
- **lib Klasörü** (.o dosyalarının bulunduğu klasör)
- include Klasörü (Başlık dosyalarınızın bulunduğu klasör)
- doc Klasörü (Raporunuzun bulunduğu klasör.)
- bin Klasörü (Çalıştırılabilir dosyalarınızın (.exe) bulunduğu klasör.)
- benioku.txt (Program için ayrıca belirtmek istedikleriniz. Önerileriniz. Boş kalabilir.)
- **makefile**

**makefile olmayan ödev derlenmemiş olarak kabul edilmektedir. (-%50 puan)**

**Rapor Yoksa veya yetersiz ise (-15 Puan)**



**Rapor pdf formatında olmalıdır. Raporu ayrıca çıktı olarak getirmenize gerek yoktur. Raporunuzda kısaca sizden istenilen, öğrendikleriniz, ödevde yaptıklarınız, eksik bıraktığınız yerler, zorlandığınız kısımlar anlatılabilir. Ödev raporunda yazı boyutu 12 puntodan büyük olamaz ve en az 1 sayfa en çok 2 sayfa olabilir.**

Klasörlerinizi bir ana klasör içine koyarak .rar veya zip'leyip **SABİS** üzerinden gönderiniz. Ana klasörünüzün adı sadece sizin öğrenci (Eğer grup ise aynı zamanda \_ ile ayırarak arkadaşınızın numarasını yazmalısınız.) numaranız (örn: B111210090) olmalıdır. Yukarıda belirtilen teslim tarihinden sonra gönderilen ödev **kesinlikle kabul edilmeyecektir.**

**Önemli: SABİS üzerinden bir ödev için dosya gönderme hakkınız 3 seferdir. Hoca ekranında en son yüklediğiniz dosya görünür.**

Yazmış olduğunuz bütün kaynak kodların en başında aşağıdaki bilgiler bulunmalıdır. Bilgileri kendinize göre güncelleyiniz.

```
/**
 * @file           Dosya adı
 * @description    Programınızın açıklaması ne yaptığına dair.
 * @course         Dersi aldığınız eğitim türü ve grup
 * @assignment     Kaçıncı ödev olduğu
 * @date           Kodu oluşturduğunuz Tarih
 * @author         Gruptakilerin yazar adları ve mail adresleri
 */
```

**Önemli: Yazacağınız kodlar MinGW derleyicisinde kesinlikle derlenmelidir. Derlenmeyen kod itibar görmez ve çok düşük puan üzerinden değerlendirilir. Sadece C++ kodları kullanılmalıdır.**

**Ödev 2 Kişilik Gruplar Halinde ya da Bireysel Yapılmalıdır. 2 Kişilik Gruplar Sadece Aynı Şubeden Olursa Kabul Edilmektedir. Aksi Halde Sıfır Verilecektir.**

**KOPYA ÖDEV SIFIR OLARAK DEĞERLENDİRİLMEKTEDİR\***

**\* Şubeler farklı dahi olsa kopya ödev gönderenler sıfır alacaklardır.**