

## ÖDEV TANIMI

- Veri.txt dosyasından sayılar okunarak AVL ağaçları oluşturulması
- Her AVL ağacının yaprak düğümleri postorder okuma ile ayrı bir yığıta yerleştirilmesi
- Yığıta yerleştirilmeyen düğümler için belirli bir formül uygulanarak elde edilen ASCII değerlerinin char veri tipine dönüştürülmesi
- Yığıtların tepe değerleri karşılaştırılarak önce en küçük değer çıkarılması daha sonra yığıtların yeni durumunun karşılaştırılarak en büyük değer çıkarılması işlemlerinin devam etmesi
- Yığıtların karşılaştırılması işlemi sırasında eğer bir yığın tamamen biterse o yığının ağacı ile birlikte silinmesi ve kalan yığıtları ilk durumuna geri getirerek işlemleri tek bir ağaç kalacak şekilde sürdürülmesi

Son kalan ağacın ASCII değerinin karakter karşılığı ve AVL numarasının tabloda yazdırılması

## ÖDEVDEN ÖĞRENDİKLERİM

- Farklı veri yapılarını birlikte kullanma pratiği yapması
- Dosya okuma işlemlerinde gelişme sağlaması
- Algoritma tasarımı pratiği yapması

## ÖDEVDE YAPTIKLARIM

- Her AVL ağacı sınıfında bir yığın işaretçisi tanımladım. Yığın sınıfını dinamik dizi olarak tasarladım.
- AVL ağaçlarını bir arada tutmak için AVLTrees adında AVLTree\* dinamik dizisi oluşturdum
- ReadAVLAndAdd fonksiyonun içinde önce FindAVLNumber fonksiyonu ile dosyadaki satır sayısını yani AVL ağacı sayısını buldum. Ardından bir döngüde getline fonksiyonu ile alınan her satırdaki sayıları ilgili ağaca ekledim.
- addleaves fonksiyonu ile yaprak düğümleri ilgili ağaca ekledim ve ağacın ascii karakter karşılığını buldum.
- comparetoStacks fonksiyonu ile ağaçların ascii karakter karşılıklarını printASCII fonksiyonu ile yazdım.
- Döngü içerisinde yığın karşılaştırılma işlemini yaptım .
- Yığın sınıfına eklediğim counter değişkeni yığıttan eleman çıkınca arttırdım bu sayede yığının tepe değerinin indeksinin ne kadar geri gittiğini hesapladım.
- RestartStacks fonksiyonunu yığıtları başlangıçtaki durumuna geri getirmek için kullandım.
- RestartStacks fonksiyonu içinde yığıtların tepe değeri indeksine counter değerini ekleyip tepe değeri indeksine eşitleyerek yığıtları başlangıçtaki durumlarına döndürdüm.
- Son kalan ağacın numarası ve karakter karşılığı printTable fonksiyonu ile yazdım.
- Sınıfların yıkıcı fonksiyonlarında kalan silme işlemlerini yaptım.

## ÖDEVDE ZORLANDIĞIM KISIMLAR

- Yığıtları karşılaştırma kısmında zorlandım
- Bazı algoritma tasarım kısmında zorlandım
- İşaretçiler ile ilgili bazı problemler yaşadım