



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

VERİ YAPILARI ÖDEV RAPORU

B221210309 – Yasin Enes Yılmaz 1-B

yasin.yilmaz17@ogr.sakarya.edu.tr

SAKARYA

Aralık, 2023

Veri Yapıları Dersi

Veri Yapıları –Avl Ağaçları ve Yığın

Yasin Enes Yılmaz

Özet

Bu ödevde Avl ağaçları ve yığın kullanarak veriler.txt dosyasındaki sayılar ile ilgili çeşitli işlemler yapılır. Txt dosyasındaki her bir satır için bir Avl ağacı oluşturulur ve satırdaki sayılar bu avl ağaçlarına eklenir. Oluşan ağaç postorder dolaşarak yaprak düğümleri yani en alttaki düğümler yığına eklenir. Yığına eklenmeyen sayılar toplanır ve ascii karşılığındaki harf hesaplanır. Her satır için bu işlemler yapılır ve oluşan ağaçlar Tekli bağlı listeye eklenir. Tüm ağaçlar oluştuktan sonra sayısal işlemler gerçekleşir. Tekli bağlı liste dolaşarak ağaçların içinde bulunan yığınlardan çıkmak üzere olan sayılar karşılaştırılır ve en küçüğü bulunur. Aynı mantık ile en büyük sayıya sahip Ağaç bulunur. Böylelikle bir en küçük bir en büyük sayı çıkarılarak işlemler devam eder. Yığında eleman kalmadığı zaman ilgili ağaç silinir ve ilk başta oluşturulan yığınlardan bu işlemler tek bir ağaç kalıncaya kadar devam eder ve son ağaç ekrana yazdırılır.

1.Kullanılan yapılar

- Avl Ağacı – Düğümleri bir arada tutar
- Düğüm – Satırlardaki her bir sayıyı tutar
- Yığın – Avl ağacındaki yaprakların verilerini tutar
- Tekli Bağlı Liste – Avl ağaçlarını bir arada tutar

2.Yaparken zorlandığım kısımlar

Doğru ödev çıktısını almak için çok uğraştım. Aslında yaptığım işlemler doğrudu ancak eksik yaptığım şeyler vardı. Ağaç silme işleminden sonra tur sıfırlanır ve tekrardan yığınlardan en küçük sayı sonra en büyük sayı çıkarılarak işlemlere devam eder. Ancak ben turu sıfırlamadım ve çıkarma işlemine kaldığım yerden devam ettim. Daha sonra bu hatayı fark ettim ancak çıktı yine yanlıştı. Bu sefer ağaç silme işleminden sonra yığınların ilk haline döndüğünü ve işlemlerin öyle gerçekleştiğini fark ettim. Bu yüzden iki yığın oluşturup birini yedeklemek için kullanmak istedim ama kopyalama işlemi gerçekleştirilemedi. Araştırdığımda shallow copy and deep copy kavramlarıyla karşılaştım. Ben ilk başta adresini kopyaladığım için istediğim işlemler olmadı. Çünkü shallow copy metodu ile kopyaladığımda ana yığının adresini kopyaladığım için ana yığın bozuluyordu ve yine istediğim çıktıyı elde edemiyordum. Sonra deep copy ile kopyalama yöntemini öğrendim ve bu yöntemi uyguladım. Deep copy yöntemi ile ana yığını kopyaladığımda başlangıçtaki yığının özellikleri kopyalanıyordu ve ana yığım bozulmuyordu. Bu sayede yedekleme işlemi de doğru bir şekilde gerçekleştirdim ve doğru çıktıyı elde ettim.

3.Sonuç

Bu ödev bir çok yapının bir arada kullanılmasını istiyordu ve ben bu ödev sayesinde bir çok yapının bir arada kullanılmasını öğrendim. Uygun algoritmalar ve yöntemler araştırarak ödevi doğru bir şekilde ve yüksek performansla çalışmasına özen gösterdim. Heap de oluşturduğum nesneleri bellekten silinmesine özen gösterdim ve bellek taşmalarını engelledim. Bu sayede program kısa sürede çalışıyor ve tamamlanıyor.

4.Referanslar

Uygulama geliştirilirken aşağıdaki kaynaklardan faydalanılmıştır:

<https://www.geeksforgeeks.org/introduction-to-avl-tree/>

<https://www.tutorialspoint.com/cplusplus-program-to-implement-avl-tree>

<https://unrealistic.dev/posts/implementing-a-stack-using-a-dynamic-array-in-c>

<https://www.geeksforgeeks.org/shallow-copy-and-deep-copy-in-c/>