

Sude Özkanoglu

Veri Yapıları 2. Ödev

Verilen ödevin bitirilmesi ile, ödev isterleri içerisinde kazanılması beklenen yetkinlikler incelendiğinde ilk olarak AVL ağaçlarının, C++ içerisinde entegre edilmesinin yine veri yapıları dersi bünyesinde işlenen diğer yapılara göre daha zorlayıcı olduğu görüldü. AVL ağaçlarının başarılı bir şekilde projeye entegre edilebilmesi için ders bünyesinde hocamız tarafından gösterilen ve işlenen kodlar çok yararlı oldu. Bu kodlar referans alınarak geliştirilen AVL ağaçlarının başarılı bir şekilde çalışabilmesi için dosya okuma kısmı ile ödevde başlandı.

Dosya okuma 1.ödev ile benzer bir mantık içerdiği için zorlanılmadan, istenilen değerler alınarak AVL ağaçlarının içerisine yerleştirilmeye çalışıldı. Bu kısımda ödev genelinde bakılacak olursa en zorlayıcı kısım kesinlikle ağaçlarının dengesizliği ve döndürme işlemleri idi. AVL ağaçlarının oluşturulması, adres referanslarının yaratılması, ağacın elemanlarının gezilmesi ve elemanların yerleştirilmesi yapıldıktan sonra en çok uğraşılan kısım ağacın dengesizlik durumunun tespiti ve döndürülmesiydi. Bu kısım birçok hata ve dengesiz bastırılan ağaçlar sonrasında başarı ile düzeltilerek yeni adıma geçildi.

Yeni adımda yaprakların postorder okuma biçiminde yığınlar içerisine yerleştirilmesi istenildiği için, AVL ağaçlarını doğru bir şekilde dolaşabilmek adına, tek yönlü bağlı listeler kullanılarak, AVL ağaçları tek tek gezilerek, yaprakları postorder bir şekilde yığınların içerisine başarılı bir şekilde eklendi. Yapılan bu ekleme işlemi sırasında büyük bir problemle karşılaşmadı. Yığınlar içerisinde ödev dahilinde en zorlayıcı kısım ise kesinlikle bir adet yığının işi bittikten sonra silinmesi ve diğer yığınların yeni aşamaya ilk halinde başlaması kısmı oldu. Proje dahilinde dosya incelendiğinde kaç satır eleman olduğu bilinse dahi, daha büyük dosyaların okunabilmesi için kod bloklarının bu probleme elverişli geliştirilmesi en zorlayıcı kısım olarak ayrıca öne çıktı. Yığınların içerisine yerleştirilen AVL ağaçlarının işleri bittikten, yığınları ile birlikte silinmesi büyük bir problem oluşturmada da, silinme kontrolünün yapılmasının ardından başarılı bir şekilde ekrana çıktı almak diğer bir zorlayıcı kısım olarak öne çıktı.

Ekrana çıktıların doğru bir şekilde alınabilmesi için, ilk olarak ascii değerlerinin başarılı bir şekilde hesaplanması ile birlikte ekrana yazdırma konusunda, kolay bir yöntem olması bakımından fazlaca while ve for döngüsü kullanılabilecekken, ödev dosyasındaki isterler takip edildiğinde 40 saniyenin üstünde bir çözüm olmaması için, ödev dahilinde bu kısımlarda çok fazla while ve for döngüsü kullanılmamaya çalışıldı. Problemler sıklıkla farklı yöntemlerle

özölmeye alışıldı. Ekranaya ıktıların doėru bir řekilde alınması ile birlikte, son olarak karşılaşılan en büyük problem ise, bir adet AVL ağacının kaldığını tespit etmek oldu. Bu problemin özümünde, sayısızca hafıza ve sonsuz döngü hataları ile uğraşıldıktan sonra bu noktada özöme kavuřturularak ödev başarılı bir řekilde tamamlandı.

Sonuç olarak, ödev dahilinde AVL ağalarının yapısının her bakımdan özellikleri güzelce kavranmakla beraber, yığıtların da yapıları ayrıca kazanım olarak değeriendirilmiş olundu. Gereklı aşamalar sonrası alışan uygulamanın gereksiz kullanılan while ve for döngüleri ile birlikte performansının ne kadar düşebileceėi de tespit edilmekle beraber, öp oluşması durumu da ayrıca incelenmiş olundu.