

Proje Raporu

Bu projede, Veri Yapıları dersine yönelik bir uygulama geliřtirdim. Amaç, baėlı liste veri yapısını kullanarak bir sistem kurmak ve her bir düėümüne bir ikili arama aėacı (BST) eklemektir. Bu sayede, verileri hem baėlı liste üzerinde hem de ikili arama aėacında tutarak işlemler yapabilmeyi hedefledim. Bu proje, veri yapılarını nasıl bir arada kullanabileceğimi ve iki farklı veri yapısının birbirleriyle nasıl entegre olabileceğini görmem açısından faydalı oldu.

Projenin başlangıcında, baėlı listeyi tasarladım. Baėlı liste, her düėümün bir sonraki düėümüne işaret etmesiyle çalışan bir yapı. Burada, listenin başını ve sonunu ilk ve son işaretçileriyle takip ettim. Her düėümde, o düėümüne ait veri ve bir işaretçi bulunuyor. Ayrıca, kullanıcının işlem yapabilmesi için bir mevcut işaretçisi ekledim. Bu işaretçi, listenin hangi düėümünde olduğumuzu takip ediyor ve kullanıcıya listenin üzerinde gezinti yapma imkânı sunuyor.

Baėlı listeyi oluşturduktan sonra, her düėümün içine bir ikili arama aėacı ekledim. İkili arama aėacı, verilerin düzenli bir şekilde saklanmasını ve hızlı bir şekilde işlenmesini sağlıyor. Her düėümdeki ikili arama aėacı, veri ekleme, silme ve arama işlemleri gibi temel işlemleri verimli bir şekilde yapabiliyor. Bu aėacın içinde veri toplama, aėacın aynalanması (tersine çevrilmesi) ve düėüm silme gibi işlemler gerçekleřtirdim.

Projede dosyadan veri okuma işlemiyle başladım. Dosyadaki her satırı bir düėüm olarak baėlı listeye ekledim. Satırlardaki veriler de her düėümdeki ikili arama aėacına işlendi. Bu sayede, dışarıdan gelen verileri kullanarak baėlı listeyi dinamik bir şekilde oluşturabildim. Dosyadaki her satır, sırasıyla bir düėüm ve bu düėümde bir ikili arama aėacına dönüřtü. Bu adımda, veri giriři sağlam bir şekilde yapıldı ve proje için temel yapı kuruldu.

Kullanıcı etkileřimini sağlamak için bazı tuřlar ekledim. Kullanıcı, A ve D tuřlarıyla baėlı listeyi gezebiliyor. A tuřu, baėlı listeyi baştan sona doğru gezmesini sağlarken, D tuřu ile geriye doğru hareket edilebiliyor. W tuřu ile mevcut düėümdeki ikili arama aėacı aynalanabiliyor. Aynalama, aėacın sıralamasını tersine çeviriyor ve bu işlemi aėacın tüm düėümlerinde uyguluyorum. S tuřu ise, mevcut düėümdeki verileri siliyor. Bu tuřlar, kullanıcıya listeyi rahatça gezme ve üzerinde işlem yapma imkânı sunuyor.

Projede, ikili arama aėacının aynalanması önemli bir yer tuttu. Bu işlemde, aėacın her seviyesindeki verilerin yer deėiřtirilmesini sağladım. Aynalama işlemi, aėacın yapısını tersine çeviriyor ve verilerin sıralamasını deėiřtiriyor. Bu, aėacın her düėümündeki veriyi kontrol edip, yer deėiřtirilmesini sağlayan bir algoritma ile gerçekleştirildi.

Projeyi yaparken karşılaştığım zorluklardan biri, bağılı liste ve ikili arama ağacının uyumlu bir şekilde çalışmasını sağlamaktı. Bağılı liste ve ikili arama ağacı iki farklı yapı olduğundan, bunları bir arada kullanmak bazen zorlayıcı oldu. Her düğümde hem bağılı listenin veri tutması hem de ikili arama ağacının verileri düzenlemesi gerekiyordu. Bu iki yapıyı birleştirirken dikkat edilmesi gereken önemli noktalar vardı, fakat bu süreci başarılı bir şekilde tamamladım.

Sonuç olarak, bu proje bana veri yapıları konusunda değerli bir deneyim kazandı ve yazılım geliştirme sürecinde önemli beceriler kazandım. Bu projeyle, veri yapılarının birleşimlerini nasıl kullanabileceğimi daha iyi öğrendim ve bu deneyimi gelecekteki projelerde uygulamayı planlıyorum.