

Emir Çağlar Demirci / 2024-2025 Dönemi Veri Yapıları 1. Ödevi Raporu

B201210003

Bu ödevde, DNA dizileri üzerinde işlem yapabilmek amacıyla bağlı listeler kullanılmıştır. Ödevdeki ana hedef, bir dna.txt dosyasındaki verileri işlemek ve bu verileri bağlı listeler aracılığıyla düzenlemektir. Bu yaklaşımın kullanılması, büyük veri setlerinin verimli bir şekilde işlenmesine olanak tanımaktadır. Ayrıca, daha hızlı işlem süreleri elde etmek için kodun optimizasyonu üzerinde de durulmuştur. Bu noktada, programın genel performansını artıran önemli unsurlardan biri, veri ekleme ve çıkarma işlemlerinde bağlı liste yapısının sunduğu avantajlardan faydalanmak olmuştur.

Ödevin başlangıcında, bağlı liste yapısının temel kavramlarını öğrenmek ve uygulamak benim için önemli bir adımdı. Zamanla bu yapıyı ne kadar etkin bir şekilde kullanabileceğimi keşfettim ve üzerinde çalışarak büyük veri kümeleriyle verimli bir şekilde nasıl başa çıkabileceğimi öğrendim. Özellikle, veri ekleme ve silme işlemlerinde bağlı listenin işaretçi (pointer) mantığını kullanarak hafıza yönetimini daha doğru bir şekilde yapabilmeyi başardım. Bu deneyim, hem yazılım geliştirme sürecim açısından hem de veri yapıları konusundaki bilgimi derinleştirerek bana büyük bir katkı sağladı.

Odevde getline() ile txt dosyasını okurken, satir sonlarında delimiter olarak listeye '_' (alt tire) karakterini de ekledim. Böylece işlemler için listede gezinirken satir gecislerini '_' karakterini kontrol ederek sağladım.

Ödevde, genel olarak tek yönlü bağlı liste kullandım. Bu yapı, veri ekleme ve silme işlemlerinin hızlı bir şekilde gerçekleştirilmesini sağladı. Ancak, EkranaYaz fonksiyonunda, kromozomları sondan başa doğru taramak gerektiği için çift yönlü bağlı listenin "önceki" (previous) özelliğinden faydalandım. Bu sayede, bağlı listede her kromozom için kromozomdaki gen sayısı kadar işlem yapma yükünden kurtuldum.

Ödevin başından itibaren, zaman ve bellek karmaşıklıklarına dikkat ederek kodu yazmaya özen gösterdim. Özellikle büyük veri setleri üzerinde çalışırken, işlemlerin verimli bir şekilde yapılabilmesi için algoritma optimizasyonuna odaklandım. Örneğin, Caprazlama fonksiyonu için dosyadaki tarama sayısını en aza indirmek amacıyla, bağlı liste yapısını ve algoritma mantığını optimize ettim. Bu, programın daha hızlı çalışmasını ve işlem sürelerinin önemli ölçüde kısılmasını sağladı. Bu tür optimizasyonlar, yalnızca performansı artırmakla kalmadı, aynı zamanda projenin daha etkili bir şekilde yönetilmesini de mümkün kıldı.

OOP ile ilgili deneyimlerim bu projede pekişti.

Git ve GitHub üzerinde ilk defa çalışma fırsatım oldu. Özellikle, fetch ve branch işlemlerini yaparak sürüm kontrolünü öğrenmek, kodun daha düzenli ve güvenli bir şekilde geliştirilmesini sağladı. Bu süreç, takım çalışması ve projelerin zaman içinde nasıl daha etkili bir şekilde yönetileceği konusunda bana değerli bilgiler kazandırdı.

Sonuç olarak, bu ödev, sadece yazılım becerilerimi geliştirmekle kalmadı, aynı zamanda büyük veri setlerini işlerken karşılaşılan performans ve bellek yönetimi gibi sorunlara nasıl çözümler üretebileceğimi de gösterdi.