



SAKARYA
ÜNİVERSİTESİ

TC. SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ PR. (İÖ)

VERİ YAPILARI

1. ÖDEVİ

Dr.Öğr.Üyesi KAYHAN AYAR

Adı Soyadı: SELİM ALTIN

Öğrenci No: G231210558

2. Öğretim C

Genel Bakış

Ödevde, bir **single linked list** (tek yönlü bağlı liste) kullanarak DNA verileriyle işlem yapmaya yönelik bir yapı oluşturdum. Çalışmam iki ana listeye dayanıyor: **Kromozom listesi** ve **Popülasyon listesi**. Kromozomlar genlerle doldurulurken, her gen GenEkle fonksiyonu ile listeye eklenir. Tüm kromozomlar ise KromozomEkle fonksiyonu ile popülasyon listesine eklenir. Veriler dosyadan getline ile okunarak genlere ayrılır ve sırasıyla bu listelere eklenir.

Yaptıklarım

1. Veri Okuma ve Listeleme:

- Dosyadan okunan veriler önce genlere ayrılır, sonra kromozom olarak listeye eklenir.
- **Performans Testleri:**
 - **10 bin satır için: yaklaşık 35 ms.**
 - **800 bin satır için: yaklaşık 630 ms.**
- Bu süreler, single linked list yapısının kabul edilebilir bir performans sunduğunu gösterdi. Bellek tasarrufu amacıyla double linked list yerine single linked list kullandım.

2. Çaprazlama İşlemi:

- Kullanıcıdan kromozom indeksleri alınarak KromozomBul fonksiyonu ile listeden erişim sağlandı.
- Çaprazlama işlemiyle iki kromozomun genleri birleştirilerek yeni kromozomlar oluşturuldu.
- Yeni kromozomlar, popülasyon listesine eklendi.
- Çaprazlama işlemi hem kullanıcı girişiyle hem de parametre göndererek uygulanabilir şekilde tasarlandı.

3. Mutasyon İşlemi:

- Belirli bir kromozom ve gen seçildikten sonra GenBul fonksiyonu ile gene erişim sağlandı.
- İlgili gen, mutasyonla 'X' karakterine dönüştürüldü.

4. Otomatik İşlemler:

- IslemleriOkuVeUygula fonksiyonu ile bir dosyadan işlemler okunup sırayla uygulandı.
- Dosyadan okunan veriler sstream kullanılarak işlenip çaprazlama ve mutasyon işlemleri için parametreler gönderildi. Kod tekrarından kaçınarak aynı çaprazlama ve mutasyon fonksiyonlarını kullandım.

5. Ekran Yazma:

- Popülasyon listesindeki her kromozomun genleri üzerinde işlem yaparak en büyük gen sağdan sola tarama ile bulundu.
- Sonuçlar sırayla ekrana yazdırıldı.

6. Bellek Yönetimi:

- Sınıf destructor'ları ile dinamik bellek temizliği sağlandı.
- Örneğin, popülasyondaki tüm kromozomlar ve genler program sonunda doğru şekilde serbest bırakılıyor.

Zorlandığım Kısımlar

- **Gen Ekleme Performansı:**

- GenEkle fonksiyonunda, listenin sonuna ulaşmak için her seferinde tüm listeyi dolaşıyordum ($O(n)$ maliyet). Bu sorun, Kromozom sınıfına bir tail pointer eklenerek çözüldü. Artık genler doğrudan son düğüme ekleniyor ($O(1)$ maliyet).

- **Performans Testleri:**

- Büyük dosyalarla çalışırken performansı artırmak için getline ve verimli listeleme algoritmalarını kullanmak gerekti. Bununla birlikte, double linked list kullansaydım performans biraz daha iyileşebilirdi. Ancak bellek kullanımını optimize etmek için single linked list yeterliydi.

Eksik Bıraktığım Yerler

- **Double Linked List Kullanımı:**

- Performans avantajı sunmasına rağmen, bellek kullanımını artırdığı için tercih edilmedi.