SIRALAMA ALGORİTMALARI GÖRSELLEŞTİRİCİSİ

Proje II

Binnur Özcan  
Kocaeli Üniversitesi  
Bilişim Sistemleri Mühendisliği *191307059*

Mustafa Bulut  
*Kocaeli Üniversitesi  
Bilişim Sistemleri Mühendisliği*  
201307062

Muhammed Emin Babayiğit  
*Kocaeli Üniversitesi  
Bilişim Sistemleri Mühendisliği*  
201307005

*Özet* —Bu rapor, PyQt5 ve Matplotlib kütüphanelerini kullanarak sıralama algoritmalarının görselleştirilmesini sağlayan bir Python uygulamasını tanıtmaktadır. Kullanıcılar, liste boyutunu belirleyebilir, bir sıralama algoritması seçebilir ve animasyon hızını ayarlayabilir. Uygulama, oluşturulan listenin grafiksel temsiline dayalı olarak sıralama algoritmalarının adımlarını animasyonlu bir şekilde gösterir.

Anahtar Kelımeler—PyQt5, Matplotlib, Sıralama algoritmaları, Görselleştirme, Timer, Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, Quick Sort

# GİRİŞ

Sıralama algoritmaları, bir dizi veya liste içindeki öğeleri belirli bir sıraya göre düzenlemek için kullanılan temel algoritmalar olarak karşımıza çıkar. Bu algoritmaların çalışma prensiplerini ve adımlarını anlamak, programcılar için önemli bir konudur. Görselleştirme, sıralama algoritmalarını daha iyi anlamak ve adımlarını izlemek için etkili bir yol sağlar.

Bu rapor, bir Python uygulaması aracılığıyla sıralama algoritmalarının görselleştirilmesini ele almaktadır. Uygulama, PyQt5 ve Matplotlib kütüphanelerini kullanarak kullanıcı arayüzü oluşturur ve sıralama algoritmalarının adımlarını animasyonlu bir şekilde gösterir. Kullanıcılar, liste boyutunu belirleyebilir, bir sıralama algoritması seçebilir ve animasyon hızını ayarlayabilir. Oluşturulan liste, Matplotlib tarafından çizilen bir çubuk grafikle görselleştirilir ve seçilen sıralama algoritmasının adımları görsel olarak gösterilir.

Bu uygulama, sıralama algoritmalarının çalışma prensiplerinin daha iyi anlaşılmasına yardımcı olurken aynı zamanda PyQt5 ve Matplotlib gibi popüler Python kütüphanelerinin alternatif kullanımlarının gösterilmesini sağlamıştır. Ayrıca, PyQt5 ile kullanıcı arayüzü oluşturma, etkileşimli grafik gösterimi ve animasyonlar gibi konuları da ele almaktadır.

# SIRALAMA ALGORİTMALARI

1. Seçme Sıralaması (Selection Sort):

* Bu algoritma, listedeki en küçük veya en büyük elemanı seçerek sıralama işlemini gerçekleştirir.
* İlk adımda, listedeki en küçük veya en büyük eleman bulunur ve listenin başıyla yer değiştirilir.
* İkinci adımda, ikinci en küçük veya en büyük eleman bulunur ve listenin ikinci elemanıyla yer değiştirilir.
* Bu işlem listedeki tüm elemanlar sıralanana kadar devam eder.

1. Kabarcık Sıralaması (Bubble Sort):

* Bu algoritma, komşu elemanları karşılaştırarak sıralama işlemini gerçekleştirir.
* İlk adımda, listedeki ilk iki eleman karşılaştırılır ve gerektiğinde yer değiştirilir.
* Ardından, ikinci ve üçüncü eleman karşılaştırılır ve gerektiğinde yer değiştirilir.
* Bu işlem listedeki tüm elemanlar sıralanana kadar devam eder. Her adımda en büyük eleman sağ tarafa doğru ilerler.

1. Ekleme Sıralaması (Insertion Sort):

* Bu algoritma, elemanları sıralı bölgeye doğru tek tek ekleyerek sıralama işlemini gerçekleştirir.
* İlk adımda, listenin ilk elemanı sıralı bölge olarak kabul edilir.
* Diğer elemanlar sırasıyla seçilir ve sıralı bölgedeki uygun konuma yerleştirilir.
* Elemanların yerleştirme işlemi sırasında, sıralı bölgedeki elemanlar kaydırılır ve uygun konuma eleman eklenir.

1. Birleştirme Sıralaması (Merge Sort):

* Bu algoritma, "böl ve fethet" stratejisine dayanır ve rekürsif olarak çalışır.
* İlk adımda, liste ikiye bölünür ve her iki parça için aynı sıralama algoritması uygulanır.
* Ardından, iki sıralı parça birleştirilirken doğru sıralama sağlanır.
* Bu işlem parçalar tekrar birleştirilene kadar devam eder.

1. Hızlı Sıralama (Quick Sort):

* Bu algoritma da "böl ve fethet" stratejisini kullanır ve rekürsif olarak çalışır.
* İlk adımda, bir "pivot" elemanı seçilir. Listenin diğer elemanları pivot elemanına göre küçük veya büyük olarak ayrılır.
* Pivot elemanının sağında ve solunda yer alan alt listeler için aynı sıralama algoritması uygulanır.
* Bu işlem alt listerler sıralanana kadar devam eder.

Bu şekilde, proje için belirtilen sıralama algoritmaları kullanılmıştır. Her bir algoritmanın kendine özgü bir çalışma prensibi vardır ve listenin elemanlarını farklı şekillerde sıralar. Seçme Sıralaması, Kabarcık Sıralaması, Ekleme Sıralaması, Birleştirme Sıralaması ve Hızlı Sıralama algoritmaları, çeşitli senaryolarda kullanılan etkili sıralama yöntemleridir. Bu algoritmaların seçimi, projenin gereksinimlerine ve performans beklentilerine göre yapılabilir.

# KOD AÇIKLAMALARI

Bu çalışma bir PyQt5 uygulamasıdır ve sıralama algoritmalarının görselleştirilmesini sağlar.

1. İlk olarak gerekli kütüphaneler import edilir:

sys: Sistemle ilgili işlemler için kullanılır.

random: Rastgele sayılar oluşturmak için kullanılır.

PyQt5.QtWidgets: PyQt5 ile kullanıcı arayüzü bileşenlerini oluşturmak için kullanılır.

PyQt5.QtCore: PyQt5'in çekirdek bileşenlerini içerir, zamanlayıcı için kullanılır.

PyQt5.QtGui: PyQt5'in grafiksel bileşenlerini içerir, sayı girişlerinin kontrolü için kullanılır.

matplotlib.pyplot: Grafik çizimleri için kullanılır.

matplotlib.backends.backend\_qt5agg.FigureCanvasQTAgg: Matplotlib grafiklerini PyQt5 uygulamasında görüntülemek için kullanılır.

1. SortingVisualization adında bir sınıf tanımlanır ve QWidget sınıfından türetilir.
2. Sınıfın \_\_init\_\_ metodu, sınıf özelliklerini ve kullanıcı arayüzünü oluşturur:

Başlık ve düzen oluşturulur.

Animasyon hızı ve zamanlayıcı tanımlanır.

Giriş paneli, algoritma paneli, grafik paneli ve düğme paneli oluşturulur.

1. create\_input\_panel metodu, giriş panelini oluşturur:

QHBoxLayout kullanılarak yatay bir düzen oluşturulur.

Liste boyutu etiketi ve metin girişi eklenir.

1. create\_algorithm\_panel metodu, algoritma panelini oluşturur:

QHBoxLayout kullanılarak yatay bir düzen oluşturulur.

Sıralama algoritmaları için bir açılır liste oluşturulur.

1. create\_graph\_panel metodu, grafik panelini oluşturur:

matplotlib.figure.Figure sınıfından bir figür oluşturulur.

FigureCanvas kullanılarak figürün PyQt5 uygulamasında görüntülenmesi için bir tuval oluşturulur.

1. create\_button\_panel metodu, düğme panelini oluşturur:

QHBoxLayout kullanılarak yatay bir düzen oluşturulur.

"Create", "Start", "Stop" ve "Reset" adında dört düğme eklenir.

Hız ölçeği/slider'ı oluşturulur.

1. update\_animation\_speed metodu, animasyon hızını günceller:

Değer değiştiğinde çağrılır ve yeni animasyon hızını alır.

Timer'ın zaman aralığını günceller.

1. create\_list metodu, bir liste oluşturur:

Kullanıcı tarafından girilen liste boyutunu alır.

random.sample kullanılarak 1 ile 100 arasından rastgele sayılarla bir liste oluşturur.

show\_graph metodunu çağırarak grafiği görüntüler.

1. show\_graph metodu, grafiği görüntüler:

figure.clear() ile figür temizlenir.

Bir çubuk grafik oluşturulur ve listeyi gösterir.

canvas.draw() ile çizilen grafikleri görüntüler.

1. start\_animation metodu, sıralama algoritması animasyonunu başlatır:

Seçilen sıralama algoritmasını alır.

Algoritma türüne göre ilgili animasyon metodu çağırılır.

1. bubble\_sort\_animation metodu, Bubble Sort algoritmasının animasyonunu başlatır:

Animasyon adımını ve timer'ı başlatır.

1. animate\_step metodu, animasyon adımını gerçekleştirir:

Animasyon adımı listenin sonuna ulaşırsa timer'ı durdurur.

Bubble Sort algoritması adımlarını gerçekleştirir ve her adımda grafik güncellenir.

QApplication.processEvents() çağrısıyla uygulamanın güncellenmesi sağlanır.

Bir sonraki adıma geçmek için return kullanılır.

1. Diğer sıralama algoritması animasyon metotları buraya eklenebilir.
2. stop\_animation metodu, animasyonu durdurur:

Timer'ı durdurur.

Animasyon adımını sıfırlar.

Liste ve metin girişini temizler.

Grafiği görüntüler.

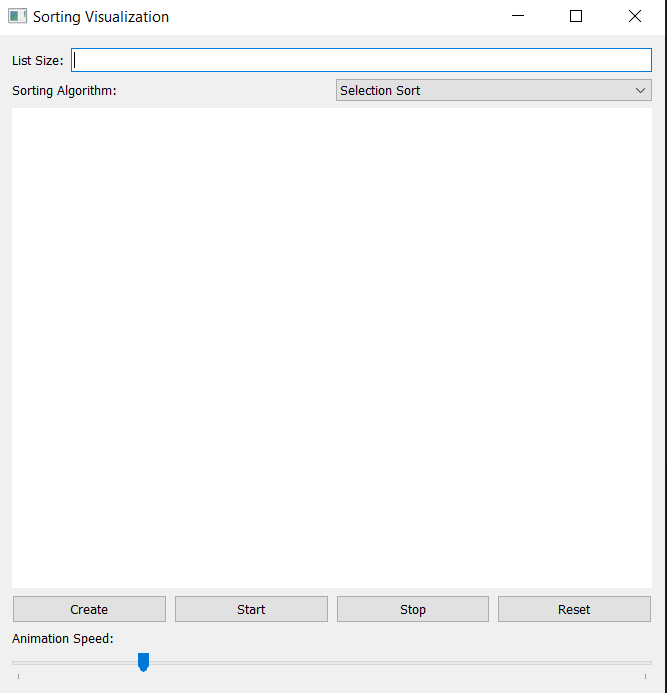
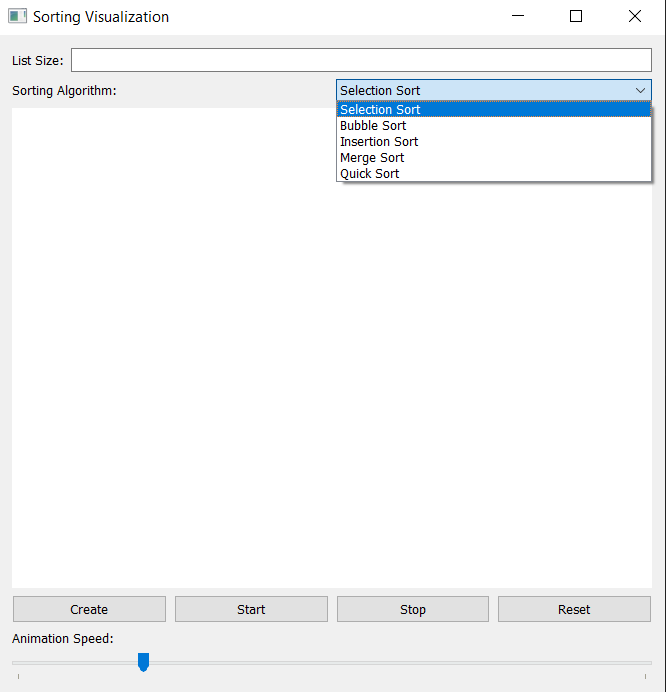
1. Ana program kısmı:

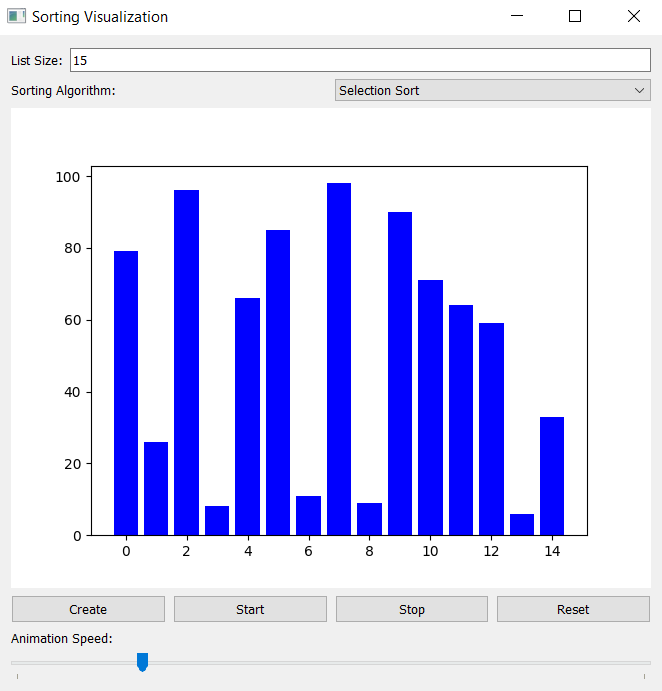
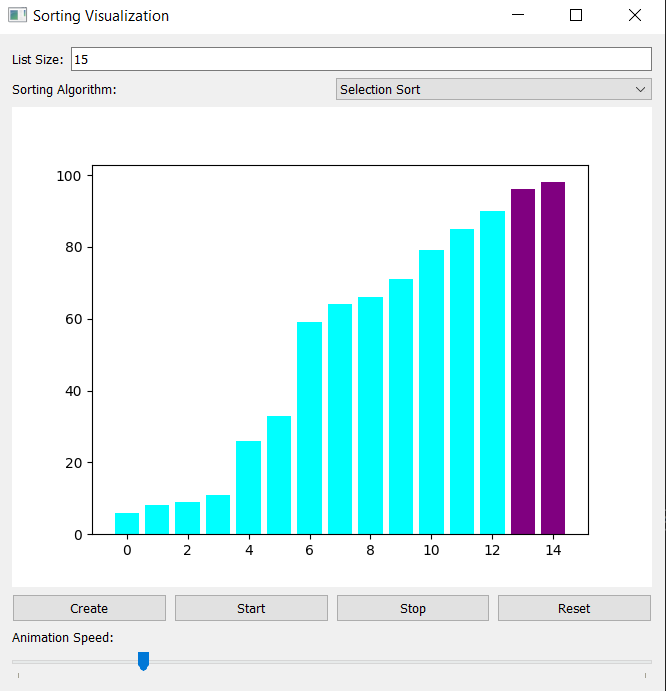
QApplication ve SortingVisualization sınıfından bir örnek oluşturulur.

Uygulama penceresini görüntüler.

Bu kod, PyQt5 kullanarak sıralama algoritmalarının görselleştirilmesini sağlar. Kullanıcı, liste boyutunu belirleyebilir, bir sıralama algoritması seçebilir ve animasyon hızını ayarlayabilir. Daha sonra, oluşturulan listenin grafiği gösterilir ve seçilen algoritmanın animasyonu başlatılır. Animasyon adımları, belirtilen hızda gerçekleşir ve grafiği günceller. Kullanıcı ayrıca animasyonu durdurabilir veya sıfırlayabilir.

Uygulamadan Görseller :





KAYNAKÇA

<https://www.youtube.com/watch?v=aubOM9dOy6c>

<https://bilgisayarkavramlari.com/2008/08/09/siralama-algoritmalari-sorting-algorithms/>

<https://www.geeksforgeeks.org/sorting-algorithms/>

<https://betterexplained.com/articles/sorting-algorithms/>

<https://www.geeksforgeeks.org/sorting-algorithms-in-python/>