

# Sıralama Algoritmaları Görselleştiricisi

201307026@kocaeli.edu.tr  
İsa Oğuz

201307046@kocaeli.edu.tr  
Mehmet Yusuf Albay

201307021@kocaeli.edu.tr  
Yunus Emre Dal

**Özet**—Bu proje, Python programlama dili ve PyQt5 grafiksel kullanıcı arayüzü kütüphanesi kullanılarak geliştirilen bir Sıralama Algoritmaları Görselleştiricisi'ni içermektedir. Bu yazılım, farklı sıralama algoritmalarının nasıl çalıştığını ve veri kümesinin nasıl sıralandığını görsel olarak sunar.

**Abstract**— This project includes a Sorting Algorithms Visualizer developed using the Python programming language and the PyQt5 graphical user interface library. This software visually presents how different sorting algorithms work and how the dataset is sorted.

**Keywords**—python, PyQt5, matplotlib, bubble sort, selection sort, insertion sort, merge sort, quick sort, virtualization

## I. GİRİŞ .

Sıralama algoritmaları, programlamada sıklıkla kullanılan ve verileri belirli bir düzene göre düzenleyen algoritmalar. Bu algoritmalar, veri analitiği, liste işlemleri ve genel sıralama problemleri gibi birçok alanda önemli bir rol oynar. Ancak bu algoritmaların işleyişini anlamak ve görsel olarak takip etmek bazen zorlu olabilir. İşte bu noktada, Python dilinin güçlü kütüphaneleri olan PyQt5 ve Matplotlib devreye girer. Bu proje, PyQt5 ve Matplotlib kütüphanelerini kullanarak sıralama algoritmalarını görselleştiren interaktif bir uygulamayı amaçlamaktadır. Kullanıcı, bubble sort, insertion sort, selection sort, quick sort ve merge sort gibi farklı sıralama algoritmalarından birini seçebilir ve bu algoritmanın işleyişini adım adım izleyebilir. Böylece, sıralama algoritmalarının nasıl çalıştığını daha iyi anlayabilir ve verilerin nasıl düzenlendiğini görsel olarak takip edebilir. Bu proje, kullanıcı dostu bir arayüz sağlamak için PyQt5 kütüphanesini kullanır. PyQt5, Python programlarına grafiksel bir kullanıcı ara yüzü eklemek için kullanılan popüler bir araçtır. Kullanıcı, sıralanacak verileri girebilir, sıralama algoritmasını seçebilir ve işlemi başlatabilir. Ayrıca, kullanıcının isteğine bağlı olarak sıralama hızını ayarlama veya verileri çoğaltma gibi özelleştirme seçenekleri de sunulmaktadır.

## II. KULLANILAN PROGRAMLAR

### A. Python dili ve Visual Studio Code Editörü

Python, projenin temel programlama dili olarak kullanılmıştır. Python, açık kaynaklı, kolay okunabilir ve anlaşılır bir dil olmasıyla öne çıkar. Geniş bir kütüphane ekosistemi ve güçlü veri işleme yetenekleriyle bilinen Python, sıralama algoritma görselleştiricisi projemizin ana dili olmuştur.

Visual Studio Code, bir kod editörüdür ve projenin geliştirilmesinde kullanılmıştır. VS Code, Python diline özel özellikler sunar ve kod yazımını kolaylaştıran birçok eklentiye sahiptir. Projenin yazılım kodlarının oluşturulması ve düzenlenmesi için kullanılan bir ortamdır.

Arayüz tasarımı ve grafik animasyonu için kullanılan kütüphaneler:

- PyQt5

Python dilinde GUI (Graphical User Interface - Grafiksel Kullanıcı Arayüzü) uygulamaları geliştirmek için kullanılan bir kütüphanedir. Bu projede, PyQt5 kütüphanesi aracılığıyla kullanıcı dostu bir arayüz tasarlanmıştır. PyQt5, zengin widget seti ve olay işleme yetenekleri ile GUI uygulamalarının oluşturulmasını kolaylaştırır.

- Matplotlib

Matplotlib, Python dilinde grafik çizimi ve veri görselleştirmesi için kullanılan bir kütüphanedir. Bu projede, Matplotlib kullanılarak sıralama algoritmalarının her adımında verilerin grafiksel olarak görselleştirilmesi sağlanmıştır. Matplotlib, çizgi grafikleri, sütun grafikleri, dağılım grafikleri gibi farklı grafik türlerini oluşturmayı destekler.

### B. Designer

Designer, PyQt5 kütüphanesiyle birlikte gelen bir araçtır ve PyQt5 tabanlı kullanıcı ara yüzlerinin görsel olarak tasarlanmasını sağlar. Bu projede, Designer aracılığıyla kullanıcı ara yüzü tasarlanmış ve ardından Python kodlarına dönüştürülmüştür. Designer, hızlı ve verimli bir şekilde GUI bileşenlerini yerleştirmeye ve düzenlemeye olanak tanır.

### C. Github

Bu çalışmada, Sıralama Algoritmaları Görselleştiricisi kodları GitHub üzerinde depolanmıştır. GitHub, yazılım geliştiricileri için kod depolama, paylaşma, inceleme ve iş birliği yapma imkanı sağlayan bir web tabanlı sürüm kontrol sistemi ve yazılım geliştirme platformudur. Bu nedenle, GitHub'u kullanarak, Sıralama Algoritmaları Görselleştiricisi kodlarını merkezi bir yerde depolayarak, herhangi bir cihazdan erişilebilir hale getirdik. Ayrıca, GitHub'un hata takip sistemi ve proje tahtası gibi araçları, yazılım geliştirme sürecimizi yönetmemize yardımcı oldu. Bu araçlar, kodların daha iyi yönetilmesini ve geliştirme sürecindeki iş birliğini kolaylaştırdı.

### III. KULLANILAN METODOLOJİ

Projenin başlangıcında, kullanıcıların sıralama algoritmalarını görselleştirmek ve tercihlerine göre grafikleri özelleştirmek istedikleri belirlendi. Bu ihtiyaçlar, proje gereksinimlerini sağlamıştır.

#### A. Proje Yönetimi

Projeye başlarken, kullanıcıların sıralama algoritmalarını görselleştirmek ve tercihlerine göre grafikleri özelleştirmek istedikleri belirlendi. Bu ihtiyaçlar, proje gereksinimlerini belirlemek için kullanıldı.

Proje yönetimi sürecinde, projenin hedefleri ve kapsamı belirlendi. Projenin zaman çizelgesi oluşturuldu ve projedeki görevler ve sorumluluklar tanımlandı. Ayrıca, projenin geliştirilme sürecinde kullanılacak kaynaklar ve gereksinimler değerlendirildi.

Proje geliştirme sürecinde kullanılacak kaynaklar, Python, PyQt5 ve Matplotlib gibi teknolojik kaynaklar olarak belirlendi. Bu kaynakların projede uygun bir şekilde kullanılması sağlandı. Ayrıca, projede kullanılan kodların ve grafiklerin saklanması için GitHub gibi bir sürüm kontrol sistemi kullanıldı.

Projenin geliştirilmesi için takım üyeleri arasında görevler atanarak sorumluluklar paylaşıldı. Her bir görevin zaman çizelgesi belirlendi ve takip edildi. Bu sayede, proje sürecinin verimli bir şekilde ilerlemesi sağlandı.

Proje sürecinde, takım üyeleri düzenli olarak ilerlemelerini güncelledi ve iletişim halinde kaldı. Zaman çizelgesine uyum sağlamak ve proje hedeflerine ulaşmak için düzenli toplantılar ve ilerleme raporları düzenlendi.

Proje sürecinde, her bir sıralama algoritmasının doğru bir şekilde çalıştığından emin olmak için testler gerçekleştirildi. Hata ayıklama ve kalite kontrolü için test senaryoları oluşturuldu ve her bir senaryonun başarılı bir şekilde geçilmesi sağlandı.

Proje sürecinde, proje hakkında detaylı dokümantasyon oluşturuldu. Kullanıcılar ve gelecekteki geliştiriciler için projenin nasıl çalıştığı, nasıl özelleştirileceği ve kullanılacak olan sıralama algoritmalarının işleyişi hakkında bilgi içeren bir kullanım kılavuzu hazırlandı.

#### B. Proje Detaylı Tanıtımı

##### 1. Designer:

PyQt5 Designer, görsel bir araç olarak kullanıldı ve projenin kullanıcı ara yüzünün tasarımı için kullanıldı. Ara yüz, sıralama algoritma görselleştiricisi projenin kullanıcıya sunulan önemli bir bileşenidir ve kullanıcının projeyi etkileşimli bir şekilde kullanmasını sağlar.

Ara yüzün başlangıcı, PyQt5 Designer arayüzü üzerinde uygun widget'ların yerleştirilmesiyle başlar. Kullanıcının veri girişi yapabileceği ve sıralama algoritmasını seçebileceği bileşenler tasarlanır. Ayrıca, grafik tercihlerini belirlemek için gerekli bileşenler eklenir.

Widget, özelliklerine göre özelleştirilir. Örneğin, kullanıcıya veri girişi yapması için bir metin kutusu eklenir ve metin kutusunun boyutu, rengi ve diğer özellikleri ayarlanır. Butonlar, açılır listeler ve diğer bileşenler de aynı şekilde özelleştirilir.

Widget'lar, kullanıcının ara yüzü daha kolay ve düzenli kullanabilmesi için bir düzen içinde yerleştirilir. PyQt5 Designer, farklı düzen seçenekleri sunar (örneğin, düğme, yatay, dikey, ızgara düzeni vb.). Bu düzenler ara yüzün estetik ve kullanılabilirliğini artırmak için kullanılır. Designer'da yapılan tasarım, gerçek zamanlı olarak görsel ön izleme olarak görüntülenebilir. Bu, tasarımın nasıl görüneceğini ve kullanılacağını kontrol etmek için tasarım sürecinde değerli bir geri bildirim sağlar.



Resim I: Designer Ara yüz Tasarımı

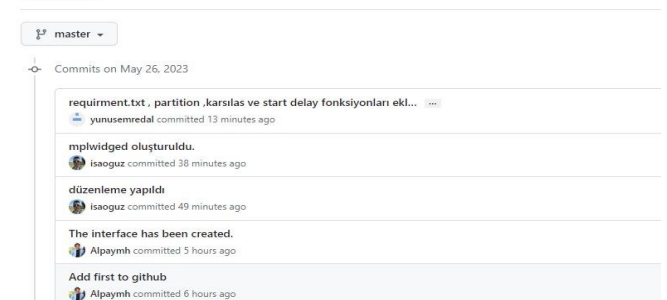
Sıralama Algoritmaları görselleştiricisi uygulamasının Designer programında tasarımı Resim-1de gösterilmiştir.

##### 2. Github:

Proje grup üyeleri İsa Oğuz, Yunus Emre Dal ve Mehmet Yusuf Alpay ile paylaşılan bir gizli repo, Github platformu üzerinde açıldı. Proje planı kapsamında, Visual Studio Code kullanılarak yapılacak görevler belirlendi ve bunlar için al, gönder ve yorumlar yapıldı. GitHub, projede geri bildirim alma ve iş birliği yapmayı kolaylaştırdı. Kullanıcılar, projenin depo sayfasında sorunlar açarak hataları bildirebilir veya önerilerde bulunabilirler. Ayrıca, proje ekibi, diğer kullanıcılar tarafından yapılan katkılara açık olacak şekilde projenin katkıda bulunma politikasını belirledi. Proje geliştirme sürecinde, projenin farklı özellikleri veya hatalarının üzerinde çalışmak için dal (branch) kullanıldı. Her bir dal, belirli bir özellik veya sorunu ele almak için oluşturuldu ve kod değişiklikleri bu dallarda yapıldı. Daha sonra, bu değişiklikler ana dala (master veya main) birleştirildi. Proje grubu ekibi, GitHub üzerindeki iş akışını belirledi ve bu iş akışına göre projenin yönetimini sürdürdü. Kod incelemeleri, birleştirme (merge) süreci ve işleri tamamlama (resolve) süreci gibi adımlar belirlendi ve bu adımlara uygun olarak işleyiş sağlandı.

Github platformu üzerinde yapılan değişikliklerin beyanları ve yorumları için commitler atıldı. Bu commitler, Resim-2'de gösterildi.

Proje takımının, yapılacak işler hakkındaki görüşlerini ve yaptıkları değişiklikleri takip etmek için Github platformu kullanması oldukça faydalı bir yöntemdir.



Resim II: Github Raporları

### C. Arayüz Tasarımı ve Uygulama Kullanımı:

Uygulama ekranının sol tarafında kullanıcı liste girişi tercihi, liste, boyut, hız, grafik tipi, sıralama algoritmaları yer almaktadır. Orta kısmında grafik animasyonu, oluştur butonu, sıfırla butonu, başla butonu ve dur butonu yer almaktadır. Ekranın sağ kısmında ise karşılaştırma sayısı ve zaman karmaşıklığı sonuçları bulunmaktadır.

Kullanıcının isteğine göre liste değerlerini ‘kullanıcı liste girişi tercihi’ ile ister rastgele değer ister de manuel olarak giriş yapabilmektedir. Otomatik giriş tercih edildiğinde ‘liste’ kısmının enable özelliği pasif olmaktadır. Manuel giriş seçildiğinde ise ‘boyut’ kısmının enable özelliği pasif olmaktadır ve listeye kullanıcının istediği sayı kadar girilebilmektedir. Boyut kısmı 0 ile 250 değerleri arasında değişmektedir. Hız kısmında kullanıcı dilerse grafik animasyonunu yavaşlatabilir isterse de hızlandırabilmektedir.

Grafik tipi için combobox kullanılmıştır. Bu combobox da sütun, dağılım ve kök grafikleri yer almaktadır. Sıralama algoritmaları bölümünde beş adet sıralama algoritması bulunmaktadır. Bunlar kabarcık (bubble) sıralama, eklemeli (insertion) sıralama, hızlı (quick) sıralama, seçme (selection) sıralama ve birleştirme (merge) sıralamasıdır.

Kullanıcı giriş tercihi, boyut, hız ve grafik tipi bilgilerini girdikten sonra oluştur butonuna basmalıdır. Sonrasında bilgilere göre grafik animasyonu oluşmaktadır. Oluşan grafik animasyonunda çizgi değerleri sıralanmamış haldedir. Sıralanması için sıralama algoritmalarından birini seçmelidir. Grafik animasyonunda çizgiler sıralanırken dur butonuna basılır ise sıralanma durmaktadır. Başla butonuna basıldığında ise sıralanma devam etmektedir. Sıfırla butonu ise girilen tüm bilgileri ve karşılaştırma sayısı, zaman karmaşıklığı sonuçlarını da temizlemektedir.

Karşılaştırma sayısı grafik animasyonunda çizgilerin birbirleri ile karşılanması sonucunda elde edilen sonuçlardır. Zaman karmaşıklığı bölümünde liste değerleri ve sıralama algoritmasına göre zaman karmaşıklığı sonucunu vermektedir. Uygulamanın ara yüzü Resim-3’de gösterildi.



Resim III: Uygulama Ekranı

#### a) Uygulamanın Sol Paneli:

- **Kullanıcı Giriş Tercihi:** Kullanıcının liste değerlerini rastgele veya manuel olarak girmesini sağlar.
- **Liste:** Manuel giriş seçimi yapıldığında listeye değer girmesini sağlamaktadır.
- **Boyut:** Rastgele liste seçimi yapıldığında listenin kaç adet değeri olması gerektiğini belirler.
- **Hız:** Grafik animasyonunda sıralama yapma hızını belirlemektedir.
- **Grafik Tipi:** Grafik animasyonun kök, dağılım veya sütun olması gerektiğini belirler.
- **Sıralama Algoritmaları:** Kullanıcının hangi sıralama algoritmasını seçmesini belirler. Bu sıralama algoritmaları kabarcık, hızlı, eklemeli, seçme ve birleştirmeli sıralamadır.

#### b) Uygulamanın Ana Paneli:

- **Grafik Animasyonu:** Kullanıcının istediği giriş tercihi, boyut, hız ve grafik tipi bilgilerini girdikten sonra oluşan grafiği gösterir.
- **Oluştur Butonu:** Grafik bilgileri girildikten sonra grafiği oluşturmak için oluştur butonuna basılması gerekmektedir.
- **Sıfırla Butonu:** Girilen bilgileri, karşılaştırma sayısı, zaman karmaşıklığı ve grafik animasyonunu temizler.
- **Dur Butonu:** Grafik animasyonunda sıralama yapmasının durması isteniliyorsa dur butonuna basılması gerekmektedir.

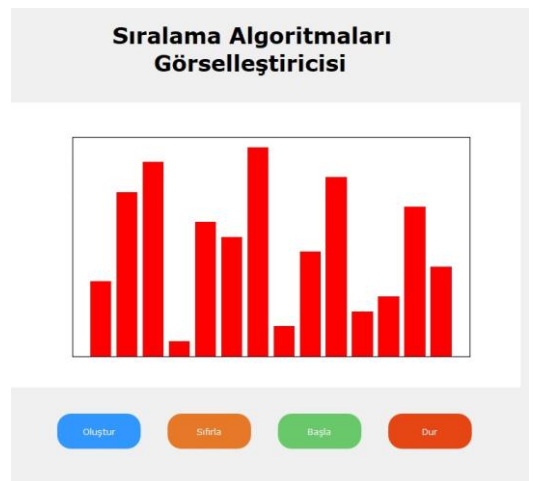
#### c) Uygulamanın Sağ Paneli:

- **Karşılaştırma Sayısı:** Grafik animasyonunda çizgilerin birbiriyle karşılaştırılma sonucu elde edilen sayıyı ekrana yazmaktadır.
- **Zaman karmaşıklığı:** Liste değerleri ve sıralama algoritmasına göre zaman karmaşıklığı sonucunu ekrana yazmaktadır.

#### i. Sütun Animasyon Grafiği:

Sütun grafik türünde, veriler sütunlar şeklinde temsil edilir ve her sütunun yüksekliği verinin değerine karşılık gelir. Sütun grafiği, sıralama işlemi sırasında verilerin nasıl değiştiğini anlamamızı sağlar. Sıralama algoritması her adımında verilerin yer değiştirdiği, swap işlemlerinin gerçekleştiği ve sıralama durumunun nasıl ilerlediği sütun grafiği üzerinde net bir şekilde görülebilir. Her sütunun yüksekliği, verinin değerine bağlı olarak belirlenir. Verilerin sıralanmamış hali için sütunlar rastgele bir düzende yerleştirilir. Sıralama algoritması her adımında verilerin yer değiştirdiği ve sıralama durumunun değiştiği gösterilir. Sütunların hareket ettiği, yer değiştirdiği veya renklerinin değiştiği animasyonlarla sıralama işlemi adım adım takip edilebilir. Swap işlemleri, sıralama işlemi sırasında verilerin yer değiştirdiği adımlardır. Sütun grafiği üzerinde, swap işlemleri vurgulanarak kullanıcının bu adımları daha iyi anlaması sağlanabilir.

Sütun Animasyon Grafiği Resim-4’de gösterildi.



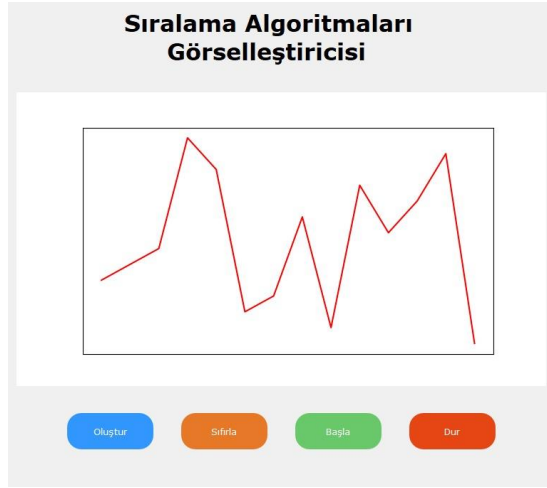
Resim IV: Sütun Grafik Animasyonu

Her adımda sıralama durumu görsel olarak temsil edilir. Sıralanmış verilerin yerleşimi, sütun grafiği üzerinde belli bir düzen oluşturur ve kullanıcının sıralama işleminin ilerlemesini net bir şekilde görmesini sağlar.

## ii. Kök Animasyon Grafiği:

Kök grafiği, sıralama algoritması görselleştirmesi için kullanılan bir grafik türüdür. Kök grafiği, verilerin sıralama işlemi sırasında nasıl değiştiğini daha iyi anlamamızı sağlar. Verilerin kökleri kullanılarak oluşturulan çubuklar, verilerin sıralanma sürecindeki değişimleri görsel olarak temsil eder. Her çubuğun yüksekliği, veri noktasının kök değeriyle orantılıdır. Sıralama işlemi adım adım gerçekleştirilirken, çubukların animasyonlu bir şekilde hareket etmesi veya geçişlerin gösterilmesi kullanıcının işlemi daha iyi takip etmesini sağlar.

Kök Animasyon Grafiği Resim-5’de gösterildi.



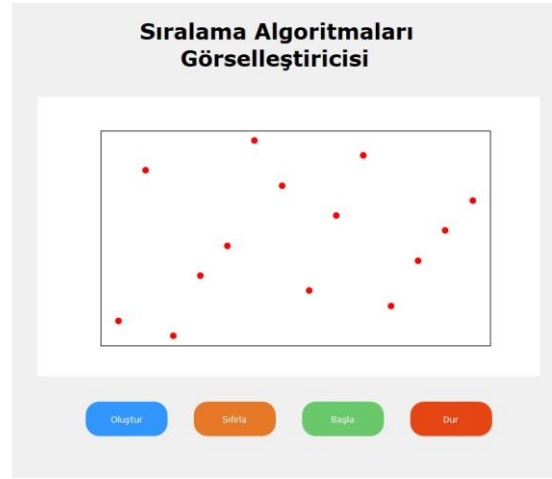
*Resim V: Kök Grafik Animasyonu*

Kök grafiği, sıralama algoritmasının veri üzerindeki etkisini daha net bir şekilde görselleştirmemize yardımcı olur. Verilerin köklerine dayanan bu grafik türü, kullanıcıların sıralama algoritmalarının performansını ve veri değişimlerini daha iyi anlamalarına yardımcı olur.

## iii. Dağılım Animasyon Grafiği:

Dağılım grafiği, sıralama algoritması görselleştirmesi için kullanılan bir grafik türüdür. Bu grafik türü, verilerin değerlerine dayalı olarak noktaların veya nokta gruplarının konumunu gösterir. Dağılım grafiği, verilerin sıralama işlemi sırasında nasıl dağıldığını ve hangi bölgelerde yoğunlaştığını anlamamıza yardımcı olur. Verilerin dağılımı, sıralama algoritmasının etkisini daha iyi gösterir ve hangi algoritmaların hangi durumlarda daha iyi performans gösterdiğini anlamamıza yardımcı olur. Hesaplanan dağılıma göre, grafik üzerinde noktaların veya nokta gruplarının konumlandırılması yapılır. Her nokta, verinin değerine ve dağılıma bağlı olarak konumlandırılır. Sıralama işlemi adım adım gerçekleştirildiğinde, noktaların veya nokta gruplarının hareket etmesi veya geçişlerin gösterilmesi kullanıcının işlemi daha iyi takip etmesini sağlar. Her adımda noktaların konumu ve yoğunluğu güncellenir.

Dağılım Animasyon Grafiği Resim-6’da gösterildi.



*Resim VI: Dağılım Grafik Animasyonu*

Dağılım grafiği, sıralama algoritmasının veriler üzerindeki etkisini daha iyi görselleştirmemize yardımcı olur. Verilerin dağılımı, hangi bölgelerde yoğunlaştığı ve nasıl değiştiği gibi bilgileri kullanıcıya aktarır. Bu sayede farklı sıralama algoritmalarının performansını değerlendirmek ve verilerin nasıl etkilendiğini anlamak daha kolay hale gelir.

## IV. SONUÇ

Kullanıcılar, seçtikleri sıralama algoritmasını uygularken grafik üzerinde gerçek zamanlı olarak görsel değişiklikleri gözlemleyebildiler. Bu sayede, algoritmaların nasıl çalıştığını ve verilerin nasıl sıralandığını daha iyi anlama imkanı elde ettiler. Grafik üzerindeki çubukların hareketi, sıralama işleminin adım adım nasıl gerçekleştiğini gösterdi.

Kullanıcılar, farklı sıralama algoritmalarını seçerek bu algoritmaların performansını karşılaştırma fırsatı buldular. Her bir algoritma, farklı zaman karmaşıklıklarına ve sıralama stratejilerine sahip olduğundan, kullanıcılar verilerin nasıl sıralandığını ve işlem sürelerini karşılaştırarak algoritmalar arasında farkları gözlemleyebildiler.

Kullanıcılar, projenin ara yüzü üzerinden grafik özelliklerini özelleştirebildiler. Sütun, kök ve dağılım grafiği seçenekleri arasından istediklerini seçerek grafik türünü belirleyebildiler. Ayrıca, renk, çizgi kalınlığı ve etiketler gibi görsel özellikleri ayarlayarak grafikleri kendi tercihlerine göre özelleştirebildiler.

Kullanıcılar, projenin ara yüzü üzerinden veri çoğaltma ve sıralama hızı ayarları yapabildiler. Veri çoğaltma özelliği, daha büyük veri setleri üzerinde sıralama algoritmalarının etkisini daha iyi gözlemlemelerine olanak sağladı. Ayrıca, sıralama hızını ayarlayarak sıralama işlemini daha ayrıntılı bir şekilde veya daha hızlı bir şekilde takip etme seçeneği sunuldu.

Sonuç olarak, bu sıralama algoritma görselleştiricisi projesi, kullanıcılara sıralama algoritmalarını daha iyi anlamaları ve görsel olarak takip etmeleri için etkili bir araç sunmaktadır. Proje, Python programlama dilinin gücünü ve PyQt5 ve matplotlib gibi kütüphanelerin esnekliğini kullanarak geliştirilmiştir. Kullanıcı dostu arayüzü ve grafiklerin etkileşimli gösterimi, proje deneyimini zenginleştirir ve kullanıcıların sıralama algoritmalarıyla ilgili bilgilerini artırır.

## REFERENCES

- [https://tr.wikipedia.org/wiki/Siralama\\_algoritması](https://tr.wikipedia.org/wiki/Siralama_algoritması)
- <https://www.w3schools.com/python/>
- <https://birhankarahasan.com/pyqt-nedir-qt-designer-nedir-python-arayuz-olusturma>
- <https://medium.com/datarunner/matplotlibkutuphanesi-1-99087692102b>

Github: <https://github.com/Alpaymh/Python-Sorting-Algorithms-Visualizer>