

# SIRALAMA ALGORİTMALARI GÖRSELLEŞTİRİCİSİ

Yakup Yaşa  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
191307028@kocaeli.edu.tr

**Özet—** Bu projenin amacı, kullanıcıların sıralama algoritmalarını ve farklı grafik türlerini kullanarak bir listenin sıralama işlemini gözlemlemeleri için görsel bir arayüz sağlamaktır. Proje, kullanıcıların manuel olarak bir liste girmesine veya belirli bir boyutta rasgele bir liste oluşturmaya izin verir. Arayüz, boyut girişi, animasyon hızı kontrolü, sıralama algoritmalarının seçimi, grafik türlerinin seçimi ve görselleştirmenin oluşturulması, başlatılması, durdurulması ve sıfırlanması için düğmeler gibi çeşitli bileşenleri içerir. Proje, Python 3.8.16'yi programlama dili olarak kullanmakta ve grafiksel kullanıcı arayüzünün (GUI) oluşturulması için Tkinter'i, grafik görselleştirmesi için ise Matplotlib'i kullanmaktadır.

**Anahtar Kelimeler—** Sıralama algoritmaları, görsel kullanıcı arayüzü, görselleştirme, Python, Tkinter, Matplotlib.

**Abstract —** The purpose of this project is to provide a visual interface for users to observe the sorting process of a list using sorting algorithms and different types of graphs. The project allows users to manually enter a list or generate a random list of specified size. The interface includes various components such as size input, animation speed control, selection of sorting algorithms, selection of graph types, and buttons for generating, starting, pausing, and resetting the visualization. The project utilizes Python 3.8.16 as the programming language and employs Tkinter for creating the graphical user interface (GUI) and Matplotlib for graph visualization.

**Keywords—** sorting algorithms, graphical user interface, visualization, Python, Tkinter, Matplotlib.

## I. GİRİŞ

Bu proje, kullanıcılara çeşitli sıralama algoritmalarını ve grafik türlerini kullanarak bir listenin sıralama işlemini gözlemlemelerine olanak sağlayan görsel bir arayüz sunmayı amaçlamaktadır. Proje, Python 3.8.16, Tkinter ve Matplotlib kullanarak grafiksel kullanıcı arayüzünü (GUI) geliştirir ve sıralama adımlarının görselleştirilmesini sağlar.

Proje, kullanıcıların manuel olarak bir liste girmesine veya belirli bir boyutta rasgele bir liste oluşturmaya izin verir. GUI, Seçme Sıralaması, Kabarcık Sıralaması, Ekleme Sıralaması, Birleştirme Sıralaması ve Hızlı Sıralama gibi sıralama algoritmalarını seçme seçenekleri sunar. Kullanıcılar ayrıca Dağılım (Scatter), Sütun (Bar) ve Kök (Stem) grafikleri de dahil olmak üzere farklı grafik türleri arasından seçim yapabilir.

GUI, animasyon oluşturma, başlatma, duraklatma ve sıfırlama düğmelerine sahiptir. "Oluştur" düğmesine tıklandığında, liste değerlerine ve grafik türüne dayalı olarak arayüz oluşturulur. "Başlat" düğmesi, belirtilen gereksinimlere göre animasyonu başlatır. "Duraklat" düğmesi, kullanıcılara animasyonu geçici olarak durdurma imkanı verir ve ardışık tıklamalarda animasyona devam etme seçeneği sunar. "Sıfırla" düğmesi, liste değerlerini temizler.

GUI'nin ana paneli, kullanıcının seçimlerinin görsel bir temsiliyi sunar. Anlaşılabilirliği artırmak için karşılaştırmalar arasında renk kodları kullanılır. Örneğin, karşılaştırılan iki değeri vurgulamak için aynı renk kullanılır. Ayrıca, sıralanmış ve sıralanmamış değerler farklı renk kodlarıyla ayırt edilir.

Sıralama süreci boyunca, her adımda yapılan karşılaştırmaların sayısı takip edilir ve ana panelde görüntülenir. Karşılaştırma sayısı belirli periyotlarla güncellenir ve kullanıcılara ilerlemenin genel bir özetini sunar. Sıralama süreci tamamlandığında, toplam karşılaştırma sayısı ve algoritmanın karmaşıklık analizi sonuçları ekranda görüntülenir.

Proje, Python programlama dili, özellikle Python 3.8.16 kullanılarak geliştirilmiştir. Grafik animasyonları, Matplotlib gibi kütüphaneler kullanılarak uygulanmıştır. Proje, GitHub üzerinde barındırılmaktadır, böylece sürüm kontrolü sağlanmakta ve işbirliğini kolaylaştırmaktadır.

Bu proje, sıralama algoritmalarının ve grafik türlerinin kapsamlı bir görselleştirmesini sunarak kullanıcıların sıralama sürecini ve adımlarını anlamalarını geliştirmeyi amaçlamaktadır. Kullanıcı dostu bir GUI ve etkileşimli animasyonların birleşimi, etkileyici bir öğrenme deneyimi sunmaktadır.

Bu raporda, projenin detayları, proje geliştirme süreci, proje mimarisi ve kullanılan teknolojiler hakkında bilgiler sunulacaktır.

## II. PROJEDE KULLANILAN TEKNOLOJİLER



Şekil 2.1. Python programlama dilinin simgesi

Bu projenin geliştirilmesinde çeşitli teknolojiler ve kütüphaneler kullanılmıştır. Aşağıda, projede kullanılan temel teknolojiler ve kütüphaneler listelenmiştir:

1. Python (3.8.16): Proje, Python programlama dili kullanılarak geliştirilmiştir. Python, kolay okunabilir ve anlaşılabilir sözdizimiyle bilinen bir programlama dilidir ve geniş bir topluluk tarafından desteklenmektedir.

2. Tkinter: Python ile GUI (Graphical User Interface) uygulamaları geliştirmek için kullanılan bir kütüphanedir. Tkinter, kullanıcı arayüzü bileşenlerini oluşturmayı, düzenlemeyi ve kontrol etmeyi sağlar. Bu projede, kullanıcı arayüzünün oluşturulması için Tkinter kullanılmıştır.

3. Matplotlib: Veri görselleştirme için popüler bir Python kütüphanesidir. Matplotlib, çizgi grafikleri, dağılım

grafikleri, sütun grafikleri, pasta grafikleri ve daha fazlasını oluşturmak için kullanılır. Bu projede, sıralama adımlarının görselleştirilmesinde Matplotlib kullanılmıştır.

Bu temel teknolojiler ve kütüphaneler, projenin amaçlarına ve gereksinimlerine uygun bir şekilde kullanılarak, sıralama algoritmalarının ve grafik türlerinin görsel bir şekilde sunulmasını sağlamıştır.

### III. METODOLOJİ

Bu projenin geliştirilmesi için aşağıdaki metodoloji adımları takip edilmiştir:

**İhtiyaç Analizi:** İlk adımda, kullanıcıların sıralama algoritmalarını ve grafik türlerini görsel olarak gözlemlemelerini sağlayacak bir arayüze olan ihtiyaç belirlendi. Kullanıcıların beklenen işlevselliği ve kullanıcı arayüzü bileşenleri üzerinde yapılan analizlerle projenin gereksinimleri belirlendi.

**Teknoloji Seçimi:** Projenin gereksinimlerini karşılamak için kullanılacak teknolojiler belirlendi. Python programlama dili, Tkinter kütüphanesi ve Matplotlib kütüphanesi seçildi. Python, esneklik ve geniş kütüphane desteği sağlayan bir dil olduğu için tercih edildi. Tkinter, kullanıcı arayüzünün oluşturulması için uygun bir seçenek olarak değerlendirildi. Matplotlib, görsel gösterimlerin oluşturulması için ideal bir kütüphane olarak seçildi.

**Tasarım:** Kullanıcı arayüzünün tasarımı ve işlevselliği belirlendi. Arayüzde kullanılacak bileşenler, kullanıcı girdilerinin alınacağı alanlar, sıralama algoritmalarının seçimi, grafik türlerinin seçimi ve kontrol düğmeleri gibi öğeler planlandı. Tasarım sürecinde kullanıcı deneyimi ve kullanılabilirlik prensipleri göz önünde bulunduruldu.

**Uygulama Geliştirme:** Tasarım aşamasında belirlenen özellikler ve işlevler, Python programlama dili kullanılarak uygulama geliştirme sürecinde hayata geçirildi. Tkinter kütüphanesi kullanılarak kullanıcı arayüzü oluşturuldu ve gerekli bileşenler entegre edildi. Sıralama algoritmaları ve grafik gösterimleri Matplotlib kullanılarak uygulandı.

**Test ve Hata Ayıklama:** Geliştirilen uygulama, test aşamasında kullanıcı girişlerine, sıralama algoritmalarının doğruluğuna ve grafik gösterimlerinin doğruluğuna tabi tutuldu. Hata ayıklama süreciyle uygulamadaki hatalar tespit edildi ve düzeltmeler yapıldı.

**Belgeleme ve Raporlama:** Projenin belgelendirilmesi için README.md dokümanı oluşturuldu. Bu dokümanda proje özeti, geliştirme ortamı, projenin yüklenmesi ve çalıştırılması, kullanılan teknolojiler ve görseller gibi bilgiler yer aldı. Ayrıca, projenin raporu hazırlanarak projenin amaçları, kullanılan literatür kaynakları ve metodoloji adımları detaylı bir şekilde açıklandı.

Bu metodoloji adımları, projenin planlı bir şekilde ilerlemesini sağlayarak, ihtiyaçlara uygun bir arayüzün

geliştirilmesini ve sıralama algoritmalarının görsel olarak sunulmasını sağladı.

### IV. SONUÇ

Bu projenin amacı, kullanıcıların sıralama algoritmalarını ve grafik türlerini görsel olarak deneyimlemelerini sağlamaktır. Proje, Python programlama dili kullanılarak geliştirilen bir kullanıcı arayüzü ve Matplotlib kütüphanesi ile gerçekleştirilen grafik gösterimleri üzerine odaklanmıştır.

Proje sonucunda, kullanıcılar manuel olarak bir liste girebilmekte veya belirli bir boyutta rasgele bir liste oluşturabilmektedir. Kullanıcılar ayrıca seçme sıralaması, kabarcık sıralaması, ekleme sıralaması, birleştirme sıralaması ve hızlı sıralama gibi farklı sıralama algoritmalarını seçebilirler. Grafik türleri olarak da dağılım, sütun ve kök grafikleri kullanılabilir. Animasyon hızı ayarlanabilir ve sıralama işlemi adımları görsel olarak sunulmaktadır.

Projede kullanılan renk kodları, kullanıcının sıralama adımlarını daha iyi anlamasını sağlamak için kullanılmıştır. Karşılaştırılan değerler aynı renk ile vurgulanırken, sıralanmış ve sıralanmamış değerler farklı renklerle tanımlanmıştır. Ayrıca, her adımda yapılan karşılaştırma sayısı da takip edilerek gösterilmiştir.

Bu proje, kullanıcıların sıralama algoritmalarını ve grafik türlerini daha iyi anlamalarını ve görsel olarak deneyimlemelerini sağlamak için tasarlanmıştır. Kullanıcı dostu bir arayüz ve etkileşimli animasyonlar, öğrenme deneyimini arttırmaya yönelik olarak birleştirilmiştir.

Sonuç olarak, bu proje sayesinde kullanıcılar sıralama algoritmalarını görsel olarak deneyimleyebilir, algoritmaların nasıl çalıştığını daha iyi anlayabilir ve grafikler aracılığıyla sıralama adımlarını takip edebilirler. Projenin kullanıcılar üzerindeki etkisi, sıralama algoritmalarına ilişkin kavrayışlarını güçlendirmek ve veri görselleştirmesinin önemini vurgulamaktır.

Bu projenin geliştirilmesi, sıralama algoritmaları, veri görselleştirme ve kullanıcı arayüzü konularında daha fazla çalışmaların temelini oluşturabilir. Kullanıcı deneyimini iyileştirmek, daha fazla sıralama algoritmasını eklemek veya farklı grafik türlerini keşfetmek gibi ilerideki çalışmalar projenin genişletilmesine yönelik olabilir.



Şekil 4.1 Uygulama arayüzü

## Kaynaklar

- [1] <https://www.pythontutorial.net/tkinter/tkinter-ttk/>
- [2] <https://www.pythontutorial.net/tkinter/tkinter-slider/>
- [3] <https://pythonassets.com/posts/create-a-new-window-in-tk-tkinter/>
- [4] <https://www.pythontutorial.net/tkinter/tkinter-radio-button/>
- [5] <https://www.askpython.com/python-modules/tkinter/stringvar-with-examples>
- [6] <https://www.programiz.com/dsa/bubble-sort>
- [7] <https://python-course.eu/tkinter/entry-widgets-in-tkinter.php>
- [8] <https://www.programiz.com/dsa/selection-sort>
- [9] <https://www.programiz.com/dsa/insertion-sort>
- [10] <https://www.programiz.com/dsa/merge-sort>
- [11] <https://www.programiz.com/dsa/quick-sort>
- [12] [https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.animation.FuncAnimation.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.animation.FuncAnimation.html)
- [13] <https://www.geeksforgeeks.org/matplotlib-animation-funcanimation-class-in-python/>