

Sıralama Algoritmaları Görselleştiricisi

Kocaeli Üniversitesi, Bilişim Sistemleri Mühendisliği Bölümü, Yazılım Geliştirme Laboratuvarı II, 2022-2023 Bahar, Proje II

Alperen Kuzhan
Kocaeli Üniversitesi
Bilişim Sistemleri Mühendisliği
191307031
kuzhanalperen@gmail.com

Hayri Batuhan Aral
Kocaeli Üniversitesi
Bilişim Sistemleri Mühendisliği
191307033
batuhanaral3@gmail.com

Hüseyin Oğuz Çetinkaya
Kocaeli Üniversitesi
Bilişim Sistemleri Mühendisliği
191307003
huseyinoguzc@gmail.com

Özet— Bu projede, kullanıcının sıralama algoritmalarını ve grafik türlerini kullanarak bir listenin sıralanmasını görsel olarak görmesini sağlayan bir arayüz geliştirildi. Kullanıcı, listenin boyutunu belirleyebilir ve istediği sıralama algoritmasını ve grafik türünü seçebilir. Arayüzde bulunan butonlar sayesinde liste oluşturulabilir, animasyon başlatılabilir, durdurulabilir ve sıfırlanabilir. Sıralama işlemi gerçekleştirilirken karşılaştırma sayısı ve analiz sonuçları da ekranda gösterilmektedir. Proje, Python programlama dili ve grafik animasyonları için farklı kütüphaneler kullanılarak geliştirilmiştir.

Anahtar kelimeler—sort algorithms, python, tkinter, matplotlib

I. GİRİŞ

Bu rapor, sıralama algoritmaları ve grafik türlerini kullanarak bir listenin görsel olarak sıralanmasını sağlayan bir projenin sunumudur. Bu projenin amacı, kullanıcının sıralama işlemlerini adım adım takip edebilmesini ve sıralamanın sonuçlarını görsel bir şekilde anlayabilmesini sağlamaktır.

Raporun ilerleyen bölümlerinde, projenin problem tanımı, geliştirme ortamı, uygulama geliştirme aşaması, uygulamanın tanıtımı ve kullanımı, karşılaşılan zorluklar ve projenin bize kattıkları gibi konular ele alınacaktır.

Projede, kullanıcı arayüzü(GUI) üzerinde birçok bileşen bulunmaktadır. Kullanıcı, sıralanacak listenin boyutunu ve oluşturulacak animasyonun hızını belirleyebilir. Ayrıca, farklı sıralama algoritmaları ve grafik tipleri arasından seçim yapabilir.

Arayüzde bulunan "Oluştur", "Başla", "Dur" ve "Sıfırla" butonları, kullanıcının işlemi kontrol etmesini sağlar. Sıralama işlemi gerçekleştirilirken, her adımda yapılan karşılaştırma sayısı gösterilir ve animasyon hızı kullanıcı tarafından ayarlanabilir.

Proje, Python programlama dili ve grafik animasyonları için çeşitli kütüphaneler kullanılarak geliştirilmiştir. GitHub platformu, proje sürüm kontrolü ve paylaşımı için kullanılmıştır.

Raporun devamında, projenin detayları ve aşamaları ayrıntılı bir şekilde açıklanacaktır.

II. PROBLEM TANIMI

Bu projenin temel amacı, sıralama algoritmalarını ve grafik türlerini kullanarak bir liste üzerinde gerçekleştirilen sıralama işlemlerini görselleştirmektir. Kullanıcıların sıralama algoritmalarının çalışma mantığını daha iyi anlaması ve sıralama işlemlerinin adımlarını görsel olarak takip etmesi hedeflenmektedir.

Sıralama algoritmaları, bir liste içindeki elemanları belirli bir kriter doğrultusunda sıralayan yöntemlerdir. Bu projede, kullanıcılar farklı sıralama algoritmalarını seçebilir ve bu

algoritmaların listeyi nasıl sıraladığını görsel olarak izleyebilirler.

Proje aynı zamanda grafik türlerini de desteklemektedir. Kullanıcılar, sıralama işlemlerinin sonuçlarını dağılım (scatter), sütun (bar) veya kök (stem) grafikleriyle görselleştirebilirler. Bu şekilde, sıralama işleminin sonucunu daha rahat bir şekilde anlayabilirler.

Bu projede karşılaşılan temel problem, sıralama algoritmalarının ve grafik türlerinin kullanıcı dostu bir arayüz üzerinde nasıl görselleştirileceği ve sıralama işleminin adımlarının nasıl takip edileceğidir. Ayrıca, sıralama işleminin hızını ayarlamak ve kullanıcının istediği zaman duraklatmak veya yeniden başlatmak gibi interaktif kontrollerin sağlanması da önemli bir problem olarak ortaya çıkmaktadır.

Bu proje, kullanıcılara sıralama işlemlerini anlamaları ve görsel olarak takip etmeleri için bir platform sunarak, sıralama algoritmalarıyla ilgili öğrenme sürecini kolaylaştırmayı hedeflemektedir.

III. GELİŞTİRME ORTAMI

Bu projenin geliştirme süreci için çeşitli araçlar ve kütüphaneler kullanılmıştır. Geliştirme ortamı, Python programlama dilini temel alarak, Tkinter, Numpy ve Matplotlib gibi kütüphaneleri içermektedir. Ayrıca, Visual Studio Code IDE'si ve PyInstaller aracı projenin geliştirilmesi ve dağıtılması aşamalarında kullanılmıştır.

Python, proje için seçilen programlama dilidir. Yüksek seviyeli ve kullanımı kolay bir dil olması, projenin hızlı bir şekilde geliştirilmesini sağlamıştır. Python, zengin kütüphane desteği sayesinde sıralama algoritmalarının uygulanması ve grafiklerin oluşturulması gibi işlemlerde büyük bir kolaylık sağlamaktadır.

Tkinter, Python için standart bir GUI (Grafiksel Kullanıcı Arayüzü) kütüphanesidir. Bu proje için kullanıcı arayüzünün oluşturulmasında Tkinter kullanılmıştır. Tkinter, kullanıcı dostu bir arayüzün hızlı bir şekilde oluşturulmasına imkan sağlayarak projenin kullanıcı deneyimini artırmıştır.

Numpy kütüphanesi, bilimsel hesaplamalar için kullanılan güçlü bir kütüphanedir. Proje kapsamında Numpy, liste işlemleri için kullanılmıştır. Özellikle rastgele liste oluşturma işleminde Numpy'nin sunduğu fonksiyonlar kullanılmıştır.

Matplotlib, Python tabanlı bir veri görselleştirme kütüphanesidir ve projede grafik türlerinin oluşturulması için kullanılmıştır. Scatter, bar ve stem grafikleri gibi çeşitli grafik türleri, Matplotlib'in sağladığı fonksiyonlar kullanılarak projede entegre edilmiştir. Kullanıcıların sıralama işlemlerini daha iyi anlamalarını sağlamak amacıyla Matplotlib'in görselleştirme yetenekleri büyük bir avantaj sağlamıştır.

Visual Studio Code, projenin geliştirme ortamı olarak kullanılmıştır. Bu entegre geliştirme ortamı (IDE), Python

kodunun hızlı bir şekilde yazılmasına, hata ayıklanmasına ve proje dosyalarının yönetilmesine olanak tanımıştır. Visual Studio Code'un kullanımı kolay arayüzü ve gelişmiş özellikleri, projenin etkin bir şekilde geliştirilmesine katkı sağlamıştır.

PyInstaller, Python programını .exe dosyasına dönüştüren bir araçtır. Projede, uygulamanın Windows işletim sisteminde çalışabilir bir .exe dosyası olarak dağıtılması hedeflenmiştir. PyInstaller, Python kodunu bağımsız bir çalıştırılabilir dosyaya dönüştürerek kullanıcılara uygulamayı kurma ve çalıştırma kolaylığı sağlamıştır. Bu sayede kullanıcılar, Python ortamına ihtiyaç duymadan uygulamayı kullanabilirler.

Geliştirme ortamı olarak Python, Tkinter, Numpy, Matplotlib, Visual Studio ve PyInstaller kullanmanın avantajları şunlardır:

- Python'un kolay ve anlaşılır sözdizimi, hızlı geliştirme süreci sağlar.
- Tkinter, kullanıcı dostu bir arayüzün hızlı bir şekilde oluşturulmasını sağlar.
- Numpy, veri manipülasyonu ve rasgele liste oluşturma gibi işlemleri kolaylaştırır.
- Matplotlib, çeşitli grafik türlerinin oluşturulması için güçlü bir araç sağlar.
- Visual Studio, hata ayıklama ve proje yönetimi gibi özellikleriyle verimli bir geliştirme ortamı sunar.
- PyInstaller, uygulamanın bağımsız bir .exe dosyasına dönüştürülerek kullanıcılar için kolaylık sağlar ve Python ortamına ihtiyaç duymadan uygulamanın çalıştırılabilmesini sağlar.

IV. UYGULAMA GELİŞTİRME AŞAMASI

Bu bölümde uygulama geliştirme aşamasında yapılan çalışmalar ve kullanılan fonksiyonlar hakkında açıklamalar bulunmaktadır. Aşağıda, proje geliştirme sürecinde yer alan temel adımlar ve bu adımlarda kullanılan önemli fonksiyonlar verilmiştir:

A. Kullanıcı Arayüzünün Oluşturulması:

Tkinter kütüphanesi kullanılarak kullanıcı arayüzü tasarlanmıştır. Tkinter, Python için bir GUI kütüphanesi olduğu için arayüz bileşenleri oluşturmak ve kullanıcının etkileşimde bulunabileceği kontrolleri sağlamak için kullanılmıştır.

Kullanıcı arayüzünde, sıralama algoritmalarının seçimi, grafik türünün belirlenmesi, liste boyutunun ve animasyon hızının ayarlanması gibi işlevleri gerçekleştirmek için gerekli bileşenler tasarlanmıştır.

Arayüzde oluştur, başla, dur ve sıfırla butonları bulunmaktadır. Ayrıca kullanıcının listenin boyutunu ve oluşacak animasyonun hızını belirlemesi için gerekli olan iki adet scale (ölçeklendirme) bulunmaktadır. Yine kullanıcının seçeceği sıralama algoritması ve grafik tipi seçimi için combobox alanları yer almaktadır.

B. Liste Oluşturma:

Numpy kütüphanesi kullanılarak rasgele bir liste oluşturmak için gerekli fonksiyonlar kullanılmıştır. Kullanıcı, listenin boyutunu belirleyebilir. Kullanıcının belirlemiş olduğu boyuttaki liste 1 ve 100 arasındaki rastgele değerlerden oluşur. Liste boyutu en az 10, en çok 100 olacak şekilde belirlenebilir.

C. Sıralama Algoritmalarının Uygulanması:

Proje kapsamında beş farklı sıralama algoritması uygulanmıştır: Seçme Sıralaması (Selection Sort), Kabarcık Sıralaması (Bubble Sort), Ekleme Sıralaması (Insertion Sort), Birleştirme Sıralaması (Merge Sort) ve Hızlı Sıralama (Quick Sort). Bu sıralama algoritmalarının çalışma mantığı araştırılarak kodları python kodlama diline aktarılmıştır.

Her bir sıralama algoritması için ayrı bir Python fonksiyonu oluşturulmuştur. Bu fonksiyonlar, sıralanacak listeyi alır ve ilgili algoritmayı kullanarak sıralama işlemini gerçekleştirir.

D. Grafiklerin Oluşturulması:

Matplotlib kütüphanesi kullanılarak sıralama adımlarını görsel olarak temsil eden grafikler oluşturulmuştur.

Seçilen grafik türüne göre Dağılım (Scatter), Sütun (Bar), Kök (Stem), sıralama işlemi adımlarını yansıtan grafikler kullanıcı arayüzünde gösterilmektedir.

Grafik oluşturulduğunda ve başlat butonuna basıldığı anda aynı listenin sıralanması için oluşturulan dizi kopyalanıp sıralama algoritmalarına girdi olarak gönderilmiştir.

Her oluşturulan liste birbirinden benzersizdir. Oluştur butonuna her basıldığında karşımıza gelecek liste farklı olacaktır.

E. Animasyon ve Karşılaştırma Sayısının Güncellenmesi:

Animasyon, sıralama algoritmalarının adımlarını göstermek için kullanılan bir işlemdir. Her adımda, sıralama işlemi gerçekleştirilirken karşılaştırmaların, yer değiştirmelerin ve güncel durumun görsel olarak takip edilmesi sağlanır.

Sıralama algoritması, sıralama işlemini gerçekleştirdiği sırada matplotlib kütüphanesi ile açılan pencere sürekli olarak güncellenip bu sayede bir animasyon elde edilmektedir.

Her adımda gerçekleşen karşılaştırma sayısı, arka planda hesaplanır ve ana panelde görüntülenir. Bu sayede kullanıcı, sıralama işleminin karmaşıklığını anlayabilir. Ayrıca sıralama işlemi bittiğinde algoritmanın karmaşıklık analizi sonucu ortalama durum olarak arayüzde gösterilmektedir.

F. PyInstaller ile Uygulamanın Dönüştürülmesi:

PyInstaller, projeyi bağımsız bir çalıştırılabilir dosya haline getirmek için kullanılan bir araçtır. Bu sayede Python ortamına ihtiyaç duymadan uygulamanın kullanılabilirliği artar.

Uygulamanın .exe formatına dönüştürülmesi, kullanıcıların uygulamayı kolaylıkla çalıştırmasını sağlar ve Python kurulumu gereksinimini ortadan kaldırır.

Bu adımlar ve kullanılan fonksiyonlar, projenin geliştirme aşamasında önemli rol oynamış ve uygulamanın istenen özellikleriyle birlikte başarılı bir şekilde tamamlanmasını sağlamıştır.

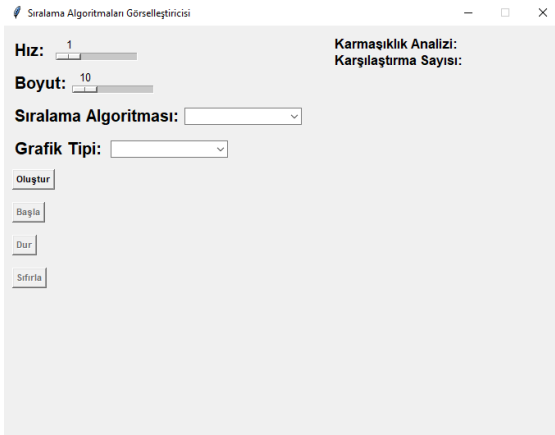
V. UYGULAMANIN TANITIMI VE KULLANIMI

Bu bölümde, geliştirilen uygulamanın tanıtımı ve kullanımı hakkında detaylı bilgiler verilmektedir. Aşağıda, uygulamanın nasıl kullanılacağı adım adım açıklanmıştır ve ilgili adımları desteklemek için görseller eklenmiştir.

Uygulamayı başlatmak için, .exe dosyasını çift tıklayarak çalıştırın.

Uygulama başladığında, kullanıcı arayüzü görüntülenecektir. Arayüz, sıralama algoritmalarının seçimi, grafik türünün belirlenmesi, liste boyutunun ve animasyon hızının ayarlanması gibi özellikleri içermektedir.

Aşağıdaki görselde, uygulama arayüzünün örnek bir görünümü bulunmaktadır:



Resim 1

Görüldüğü gibi uygulama ilk açıldığında sadece oluştur butonu aktiftir. Oluştur butonuna basmadan önce hız, boyut, sıralama algoritması ve grafik tipi eksiksiz şekilde seçilmelidir, aksi takdirde gerekli uyarı ekranda belirip kullanıcı uyarılacaktır.

"Hız" bölümünde, oluşturulacak animasyonun hızını belirleyebilirsiniz.

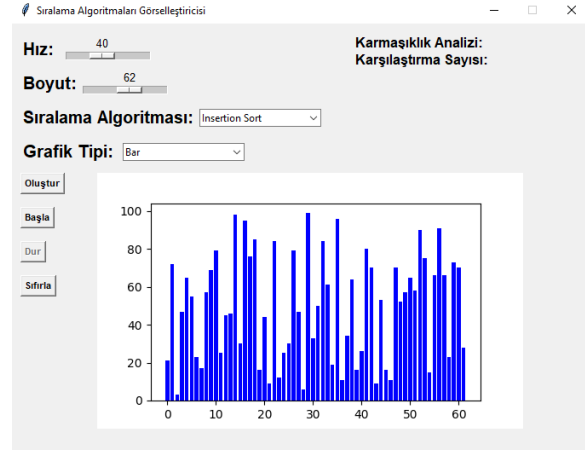
"Boyut" bölümünde, liste boyutunu belirleyebilirsiniz. Belirlediğiniz boyuta göre, rasgele bir liste oluşturulacaktır.

"Sıralama Algoritmaları" bölümünde, beş farklı sıralama algoritması arasından birini seçebilirsiniz. Ayrıca, "Grafik Tipleri" bölümünde, dağılım (scatter), sütun (bar) veya kök (stem) grafik türlerinden birini seçebilirsiniz.

Tüm giriş değerleri eksiksiz seçilip oluştur butonuna basıldığında kullanıcının belirlemiş olduğu boyutta Numpy kütüphanesi ile 1 ve 100 arasında rastgele değerlerden bir liste

oluşturulur. Oluşturulan karmaşık liste Matplotlib kütüphanesi ile görselleştirilir ve ekranda gösterilir.

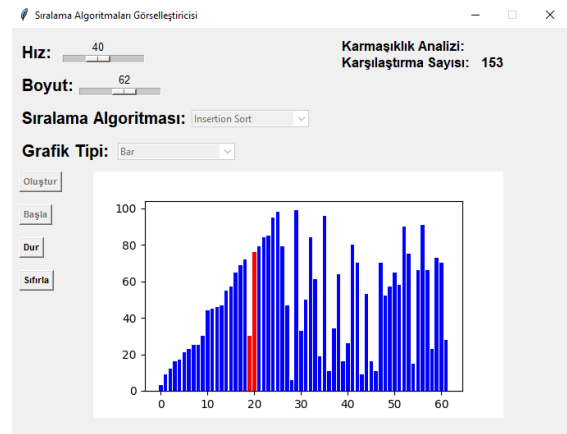
Oluştur butonuna her basıldığında farklı bir liste oluşturulup ekrana gelir. Aşağıda görselleştirilen örnek bir liste görülmektedir.



Resim 2

Oluştur butonuna tıklandıktan ve ekrana görselleştirici geldikten sonra başla ve sıfırla butonları aktif hale gelecektir. Başla butonuna tıkladığınızda, sıralama işlemi belirlediğiniz algoritma ve grafik türüne göre animasyonlu bir şekilde başlayacaktır. Dur ve sıfırla butonları da aktif hale gelecektir. Ancak oluştur butonuna basıldığında oluşturulan listenin eleman sayısı değiştirildiyse ve ardından başlat butonuna tıklandıysa eleman sayısı ve liste değiştiği için ekranda bir uyarı belirir ve başlat butonu pasif olur. Bu durumda liste değiştiği için tekrar oluştur butonuna tıklanıp yeni eleman sayısını içeren bir liste oluşturulmalıdır. Bu kontrol sağlanmıştır.

Aşağıda başlat butonuna tıklandıktan sonra gelen ekran görülmektedir.

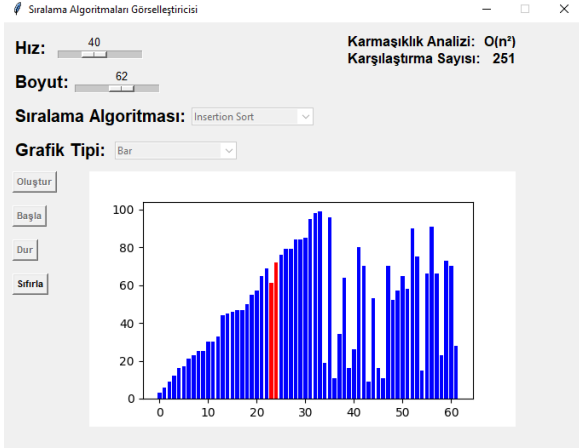


Resim 3

Dur butonuna tıkladığınızda, animasyon durdurulacaktır. Sıfırla butonu hariç diğer tüm butonlar pasif kalacaktır. Yani kullanıcı animasyonu durdurması halinde listeyi sıfırlamak durumunda olacaktır. Karşılaştırma sayısı ve karmaşıklık

analizi değerleri animasyonun durdurulduğu andaki değerler olarak ekrana gelecektir.

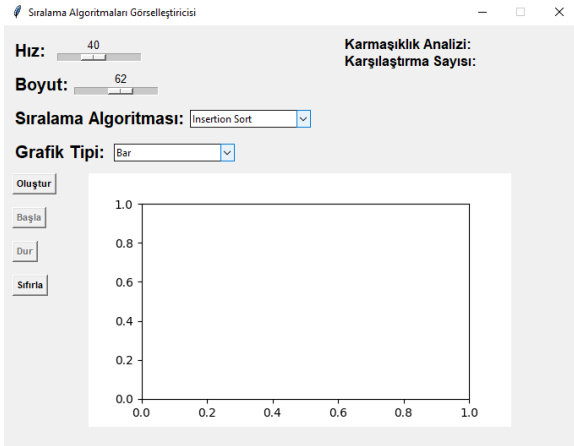
Aşağıda durdur butonuna basılması halinde karşılaşılabilecek olan ekran gözükmecektir.



Resim 4

Sıfırla butonuna tıkladığınızda liste temizlenecek, oluştur butonu aktif olacak ve uygulama yeni bir sıralama işlemi için hazır hale gelecektir.

Aşağıda sıfırla butonuna tıkladığında gelen ekran görülmektedir.



Resim 5

Kısaca özetlemek gerekirse uygulamada sıralama işlemi başladığında, adımlar görsel olarak takip edilecektir. Karşılaştırma yapılan değerler, aynı renk koduyla vurgulanacaktır.

Her adımda yapılan karşılaştırma sayısı, arka planda hesaplanacak ve ana panelde güncellenerek gösterilecektir.

Sıralama işlemi tamamlandığında, karşılaştırma sayısı ve algoritmanın karmaşık analizi sonucu elde edilen bilgiler ekrana yazdırılacaktır.

Kullanıcılar, uygulamanın sunduğu görsel takip ve bilgi paylaşımı sayesinde sıralama algoritmalarını daha iyi anlayabilir ve analiz edebilir.

Uygulamanın tanıtımı ve kullanımıyla ilgili adımlar yukarıda detaylı bir şekilde açıklanmıştır. Kullanıcılar, bu

adımları takip ederek uygulamayı kolaylıkla kullanabilir ve sıralama işlemlerini görsel olarak takip edebilirler.

VI. KARŞILAŞTIĞIMIZ ZORLUKLAR VE BİZE KATTIKLARI

Uygulamayı geliştirmeye ilk başladığımızda istekleri ve problemi iyi bir şekilde kavramaya çalıştık. Ardından problemi iki parçaya ayırdık. Bu parçalardan biri GUI (Grafiksel Kullanıcı Arayüzü) oluşturmak için gerekli kütüphane olan Tkinter'ı araştırıp öğrenmekti. Proje için gerekli olan diğer bir parça ise sıralama algoritmalarının mantığını kavramak ve bunları Python programa dili ile koda dökmekti. Yine bir Python kütüphanesi olan Matplotlib'i araştırıp yazdığımız sıralama algoritması kodlarını görselleştirdik. Bu bağlamda projeyi iki parça olarak geliştirdik.

Daha önce Python ile bir proje geliştirmemiş bir ekip olduğumuzdan ötürü bu programlama dilinin sözdizimine alışmakta güçlük çektik. Özellikle sıralama algoritmalarını koda dökerken mantığını anlamamıza rağmen bunları koda dökmekte zorluk çektik. Ayrıca proje içerisinde gerekli olan fonksiyonel işlemlerde değişkenlere ulaşmakta ve bu değişkenler üzerinde işlem yapmakta zorlandık.

Önceden Visual Studio ve C# ile masaüstü uygulamaları geliştirdiğimiz için uygulamanın tasarımını sürükleyip bırak yöntemiyle yapıyorduk fakat bu projede uygulama tasarımını tamamen kodla tasarladık. Bu da bizim için bir güçlük oldu.

Oluştur butonuna tıkladığında oluşturulan listeyi değişmeden sıralama algoritmalarına göndermekte zorlandık. Oluştur ve başla butonu farklı komutları çalıştırdığından dolayı aynı liste üzerinden işlem yapmak işimizi zorlaştırıyordu. Burada listeyi kopyalama yöntemine başvurduk ve oluşturulan listenin sıralama algoritmalarına değişmeden gönderilmesini sağladık.

Sıralama algoritmalarının karşılaştırma sayısını hesaplarken sayacı nerede arttıracığımız konusunda doğru yeri bulmakta zorlandık. Bu da yine Python dili ile daha önce proje gerçekleştirmediklerimizden kaynaklandığını düşünüyoruz.

Durdurulan animasyonun yeniden devam etmesi konusunda bir sonuca varamadığımız için animasyon durdurulduğunda sadece sıfırlama işlemini aktif ederek bu sorunu çözdük.

Sıralanan ve sıralanmamış değerleri animasyonda farklı renkte göstermek için yoğun çaba sarf ettik fakat yazmış olduğumuz koda bunu entegre edemedik. Bu yüzden animasyon sırasında sadece anlık olarak karşılaştırılan değerler farklı renkte gösterilmiştir.

KAYNAKLAR

- [1] bilgisayarkavramlari.com/2008/08/09/siralama-algoritmaları-sorting-algorithms/
- [2] docs.python.org/3/library/tk.html
- [3] pyinstaller.org/en/stable/
- [4] python.org
- [5] matplotlib.org
- [6] tutorialspoint.com/python_data_structure/python_sorting_algorithms.htm
- [7] youtube.com/@WriteandLearn/
- [8] youtube.com/@buildwithpython
- [9] youtube.com/@FelixTechTips

[10] [youtube.com/@NeuralNine](https://www.youtube.com/@NeuralNine)