



**T.C.**

**Kocaeli Üniversitesi**

**Teknoloji Fakültesi**

**Bilişim Sistemleri Mühendisliği**

**Sıralama Algoritmaları Görselleştiricisi**

**Mehmet Arda YÜKSEL**

**Güven KAZICI**

**Levent KIRIMGERİ**

**2022-2023 Bahar Dönemi**

**Yazılım Geliştirme Laboratuvarı-II 2.Proje**

**Özet—** Geliştirilen masaüstü uygulaması, kullanıcıdan alınan listeye 5 farklı sıralama algoritmasının uygulanışını, 3 farklı grafik türü ile animasyon halinde görselleştirilmesini sağlamaktadır.

**Anahtar Kelimeler-** Python, PyQt5, Matplotlib, Qt Designer, Git, GitBash, GitHub

## I. TANITIM

Geliştirilen masaüstü uygulaması, 5 farklı sıralama algoritmasının (selection, bubble, insertion, merge, quick), 3 farklı grafik türü (scatter, bar, stem) ile animasyon şeklinde görselleştirilmesini sağlamaktadır.

Masaüstü uygulamasında, arayüz üzerinden, kullanıcıların sıralama yapılmasını istedikleri listeyi manuel olarak veya rastgele şekilde girmeleri sağlanmaktadır. Sıralanacak liste girildikten sonra, sıralama algoritması ve grafik türü de ilgili radio buton'lardan seçilerek görselleştirme ve animasyon, ilgili butonlar üzerinden başlatılabilmektedir. Kullanıcı, sıralayacağı listeyi, manuel şekilde tek tek metin kutusuna tamsayı değerler girerek ya da sıralanacak listenin minimum ve maksimum değerleri ile eleman adedini olacak şekilde 3 değer girerek rastgele şekilde oluşturabilmektedir. Sonrasında sıralamanın hangi tür algoritma ile yapılacağı ve grafik türü ilgili radio butonlar üzerinden seçilip, 4 farklı kontrol butonu arasından oluştur butonuna basılarak girilen liste için grafik oluşturulabilmektedir. Arayüzde 4 farklı kontrol butonu bulunmaktadır. Bunlar; Oluştur, Başlat, Durdur ve Sıfırla butonlarıdır. Durdur butonu Durdur-Devam Et şeklinde çift yönlü çalışmaktadır. Oluştur butonu, liste girildikten sonra animasyon başlatılmadan ilgili grafik türü ile listenin sıralanmamış değerlerinin gösterilmesini sağlamaktadır. Başlat butonu, girilen sıralama algoritmasına göre sıralama işlemini gerçekleştirip, grafik üzerinde sıralama algoritmasının bütün basamaklarının animasyon şeklinde gösterilmesi işlemlerini başlatmaktadır. Durdur-Devam Et butonu animasyon esnasında, animasyonun durdurulmasını ve kaldığı yerden devam etmesini sağlamaktadır. Sıfırlama butonu ise tüm değerleri sıfırlayarak, grafiği başlangıç durumuna geri döndürmektedir.

Arayüz üzerinde ayrıca, ister animasyon başlamadan önce ister animasyon esnasında animasyonun hızını değiştirebilecek bir slider bulunmaktadır. Arayüzün en alt kısmında girilen listeye ve seçilen sıralama algoritmasına göre karmaşıklık analizi gösterilmektedir. En sağ alt kısımda ise animasyon devam ettiği sürece sıralama algoritması içerisinde kaç kez karşılaştırma yapıldığı gösterilmektedir.

## II. KULLANILAN TEKNOLOJİLER

### A. Python

Bu proje, Python programlama dili kullanılarak geliştirilmiştir. Python yüksek seviyeli, yorumlamalı, girintilere dayalı öğrenmesi kolay basit söz dizimli, nesne yönelimli bir programlama dilidir. İlk çıkışı 1990 yılına dayanmaktadır. Öncesinde gereksiz angarya olarak görülen işlerin otomatikleştirilmesinde kullanılırken, günümüzde en önemli programlama dillerinden biri haline gelmiştir. Özellikle veri bilimi ve makine öğrenmesi için yaygın olarak tercih edilen Python ile, masaüstü, web, mobil uygulamaları gibi pek çok alanda farklı framework'ler ile geliştirme yapılabilmektedir.

Proje geliştirme süresince ilk olarak, "python.org" sitesi üzerinden Python'un 3.11.3 sürümü indirilmiştir. Proje geliştirme ortamı olarak ise Visual Studio Code kullanılmıştır. VSCode içerisinde Extensions kısmından Python geliştirmede intellisense sayesinde kolaylık sağlaması açısından Microsoft'un yayınladığı eklenti indirilmiştir. Proje, "SortingAlgorithmsVisualizer" adındaki bir klasör altında geliştirilmiştir. Temel dosya olarak "Main.py" oluşturulmuş ve neredeyse tüm kodlar bu dosya içerisinde yazılmıştır. Ayrıca ikonlar ve fontlar için birer klasör oluşturulmuştur. Main.py dosyası içerisinde en başta proje için tüm gerekli modüller import edilmiştir. Bu modüllerden başlıcaları: PyQt5, math, threading, matplotlib paketleridir.

### B. PyQt5

Python ile masaüstü uygulaması yapmak için birden fazla kütüphane bulunmaktadır. Bunlar PyQt5, Tkinter, wxPython gibi kütüphanelerdir. Bu projede ise PyQt5 kullanılmıştır.

PyQt5'in kullanılabilmesi için öncelikle harici olarak indirilmesi gerekmektedir. Bunun için iki yol mevcuttur. Bir yolu sitesi üzerinden .exe uzantılı bir dosya ile kurmak, diğer yolu ise Python'un yazılım deposu olan "pypi.org" üzerinden komut satırı aracılığıyla indirmektir. Bu proje için pypi yöntemi tercih edilmiştir. Python yorumlayıcısı üzerinden "pip install PyQt5" komutu ile kolaylıkla PyQt5 kurulumu yapılmıştır. Pypi üzerinden indirildiği için içerisinde QtDesigner uygulaması da dahili olarak gelmektedir. Bu kurulum içerisinde aynı zamanda ileride bahsedilecek olan ".ui" uzantılı dosyaları ".py" uzantısına çevirmede ve ".qrc" uzantılı dosyaları "\_rc.py" uzantılı dosyaları dönüştürmeyi sağlayan "pyuic5" ve "pyrcc5" exe'leri de kurulum ile indirilmektedir.

Projede main.py dosyası içerisinde modüllerin import edildiği yerde PyQt5 import edilmiştir. PyQt5 içerisinden bütün arayüz elemanları QtWidgets adındaki alt sınıfın içerisinde bulunmaktadır.

### C. Matplotlib

Python’da grafik görselleştirme için diğer alanlarda olduğu gibi birden fazla kütüphane mevcuttur. Bunlar: Matplotlib, Seaborn, Plotly gibi kütüphanelerdir. Bu projede Matplotlib kütüphanesi kullanılmıştır. PyQt5’de olduğu gibi pypi üzerinden “pip install matplotlib” komutu ile indirilmiştir. Projedeki main.py dosyasına “matplotlib.pyplot” plt alias’ı ile import edilmiştir. Ayrıca animasyon işlemleri için “matplotlib.animation” içerisinde “FuncAnimation” sınıfı projeye dahil edilmiştir.

### D. Qt Designer

Masaüstü uygulamasını geliştirirken, tüm arayüz elemanlarını tek tek kod ile yapmamak ve süreci kolaylaştırmak için arayüzün geliştirilmesinde, sürükle bırak yöntemi ile widget’ların arayüze yerleştirilebilmesini ve properties penceresinden özelliklerinin değiştirilebilmesini sağlayan bir tasarımcı uygulaması olan QtDesigner kullanılmıştır. Temelinde Qt uygulamaları için C++ dili ile masaüstü uygulaması yapmakta kullanılsa da “pyuic5” dönüştürücüsü sayesinde PyQt5 ile de kullanılabilir. Arayüz elemanları, bu uygulama üzerinden yerleştirilmiş ve tasarımı yapılmıştır. Yalnızca yazı tipi ve dinamik değişmesi gereken tasarımlar için kod tarafından ekstra müdahale edilmiştir. Oluşturulan arayüz “.ui” uzantılı bir dosya olarak kaydedilmektedir. Python projesinde çalıştırılabilmesi için “pyuic5” exe dosyası ile “.ui” uzantılı dosyayı “.py” uzantılı Python dosyasına dönüştürmek gerekmektedir. Yapılan tüm tasarımlar en nihayetinde Python dili karşılığı ile projede çalıştırılmaktadır.

### E. Git, GitBash ve GitHub

Git, açık kaynak kodlu bir sürüm/versiyon ve kaynak kod kontrol-yönetim sistemidir. Normalde bir proje geliştirilirken projenin kaynak kodu, yazılımcının isteğine göre harddiskinde veya bulut ortamı gibi yerlerde saklanabilmektedir. Proje geliştirilmeye devam edildikçe, projenin sürümleri oluşmaktadır ve depolandığı yerlerde bir karmaşa söz konusu olmaktadır. Ayrıca, örneğin harddiskte depolanan bir projenin güvenliği de şüphelidir. Bir projenin geriye döndürülebilirliği ve de oldukça önemli bir diğer konu olan ekip ile çalışma esnasındaki kodların paylaşılabilirliği açısından da geleneksel yöntemler çok sıkıntılıdır. İşte bu ve bunun gibi tüm sorunlara çözüm olarak Git kullanılmaktadır.

Git sayesinde projelerin versiyon kontrolü çok daha kolay yapılabilmekte, ekip çalışması ve eipteki yazılımcıların eş zamanlı proje üzerinde çalışmalarında kodların paylaştırılması ve birleştirilmesi oldukça kolaylaşmakta ve tüm bu işlemlerin hızlı ve pratik bir şekilde gerçekleştirilmesi sağlanmaktadır. Proje geliştirilmesi esnasında herhangi bir aşamada istenilen değişikliklerin geri ve ileri alınabilmesini sağlamaktadır.

Kullanabilmek için öncelikle, git-scm.com sitesinden GitBash ile birlikte indirilmesi gerekmektedir.

**GitBash**, Git özelliklerinin kullanılabildiği bir komut satırıdır. Pek çok Unix komutlarının da kullanılabildiğini sağlamaktadır. Git’i projede kullanabilmek için GitBash komut satırı harici olarak kullanılabildiği gibi Visual Studio Code içerisinde terminal kısmından dahili olarak editör

içerisinden de erişilerek de kullanılabilir. Git kullanımının temelinde 3 farklı ortam yer almaktadır. Bunlardan birincisi “**working directory**” ikincisi “**staging area**” üçüncüsü ise “**local repository**”dir. Git kullanımında kaynak kodlar bu üç farklı bölgede ileri geri hareket ettirilmektedir. Working Directory(çalışma dizini) kaynak kodların yer aldığı dosyaların dizinidir. Staging Area(Geçiş Bölgesi-Index Bölgesi) dosyaların tutulduğu ara bir bölgedir. Local Repository ise proje kodlarının lokal ortamda tutulabileceği son duraktır.

Git kullanımında öncelikle ilgili proje klasörü içerisinde GitBash üzerinden “git -init” komutu ile local repo’nun oluşturulması gerekmektedir. Bu komut ile çalışma dizininde .git uzantılı gizli bir klasör oluşturulmaktadır. Bu klasör projenin Local Repository’si bir diğer deyişle zaman makinesidir. Git kullanımında en sık kullanılan komutlardan biri de “git status” komutudur. Bu komut ile dosyaların güncel durumu sorgulanabilmektedir. Git kullanımında öncelikle tüm dosyalar, “git add” komutu ile working directory’den staging directory’e aktararak izlenmeye başlanmaktadır. Staging Directory’den Local Repo’ya ise “git commit” komutu ile aktarılmaktadır. Bu işlemler ilgili git komutlarıyla tersine de çevrilebilmekte, istenilen tek bir işlem geri alınabildiği gibi belli bir işlemin olduğu yere kadarki tüm değişiklikler de geriye ya da ileriye döndürülebilmektedir.

**GitHub** ise esasında **Remote Repository**’lerden en popüler olanıdır. Yazılımcılar arasında kaynak kodların paylaşıp depolandığı bir yerdir. Aynı zamanda bir sosyal ağ görevi de görmektedir. Git kullanımıyla beraber GitHub içerisinde bir repo oluşturulmakta ve bunu local repo ile bağlayarak versiyon yönetiminin bulut ortamında gerçekleştirilmesi sağlanmaktadır. GitHub ortamı, ekip çalışmasında da elbette Git ile beraber kullanımıyla projenin eş zamanlı geliştirilmesinde kolaylık sağlamaktadır.

## III. GELİŞTİRME AŞAMALARI

### A. Yeni Python-PyQt5 Projesi

Proje, “SortingAlgorithmsVisualizer” klasörü altında geliştirilmiştir. Tüm ana kodlar “Main.py” adındaki dosya içerisinde yazılmıştır. Bu dosya içerisinde en üstte daha önce de kullanılan teknolojilerde bahsedilen kütüphaneler import edilmiştir. PyQt5 masaüstü uygulamasını oluşturmak için “QMainWindow” sınıfından kalıtım alan, “Main” adında bir sınıf oluşturulmuştur ve tüm kodlar bu sınıf altında yazılmıştır. QMainWindow sınıfı, PyQt5 altında QtWidgets paketi içerisinde yer almaktadır. Tüm arayüz elemanları, sınıfları buradan gelmektedir. Projenin çalışması için en temel iki sınıftan biri QMainWindow iken diğeri yine aynı paketten import edilen QApplication sınıfıdır. Main.py içerisinde en altta window adında bir fonksiyon oluşturulmuştur. Bu fonksiyon içerisinde QApplication sınıfından ve QMainWindow sınıfından kalıtım alan “Main” sınıfından birer nesne türetilmektedir. Bu nesne üzerinden “show()” metodu çalıştırılmaktadır. Bu metod sayesinde masaüstü uygulaması başlatılmaktadır. Oluşturulan Main sınıfı içerisinde “\_\_init\_\_” metodu ile constructor yazılmıştır. Constructor metodu içerisinde QtDesigner’dan

dönüştürülen arayüzün tanımlanması, arayüz elemanlarının başlangıç değerleri, signal-slot bağlantıları gibi pek çok tanımlama gerçekleştirilmektedir. Main dosyasının devamında ise amaca göre farklı işlemler gerçekleştiren pek çok metod yazılmıştır.

PyQt5 uygulamalarında arayüz elemanlarına signal-slot bağlantısı uygulanmaktadır. Buradaki signal, elemanın hangi durumlarda tetikleneceği anlamına gelmektedir. Slot ise tetiklendiğinde hangi metodun çalıştırılacağını belirtmektedir. Tüm arayüz elemanları bu mantık ile çalışmaktadır. Yazılan metodlar, ilgili widget'ların istenilen sinyaline (örneğin mouse sol tıklaması gibi) bağlanmaktadır.

#### B. Sıralanacak Listenin Manuel Şekilde Oluşturulması

Projede, kullanıcı sıralanacak listeyi manuel ya da rastgele olacak şekilde iki yolla girebilmektedir. Sıralanacak listenin, manuel şekilde girilebilmesi için, arayüz üzerinde sol panele QScrollArea ve içerisine de QLineEdit QtWidget'ları eklenmiştir. QLineEdit'e kullanıcı listenin elemanlarını tam sayı olarak girebilmekte ve her enter veya space tuşuna basmasıyla QScrollArea içerisinde yeni bir QLineEdit elemanı üretilmektedir. Bu işlem kod kısmında dinamik bir şekilde gerçekleştirilmektedir. Arayüz üzerinde başlangıçta tek bir QLineEdit bulunmaktadır. Kullanıcı istediği sayıda elemanı teker teker girebilmektedir. QScrollArea ise yana doğru artan QLineEdit elemanları taşımaya sebep olacağı için elemanlar arasında gezinebilmeyi sağlamaktadır. QScrollArea altında kullanıcı isterse girdiği liste elemanlarını birer birer ya da tamamını silebileceği iki buton yerleştirilmiştir. Kod kısmında bir for döngüsüyle bütün QLineEdit'ler turlanarak içerisindeki tamsayı değerler, programın devamında kullanılacak listeye eklenmektedir.

#### C. Sıralanacak Listenin Rastgele Şekilde Oluşturulması

Kullanıcının, sıralanacak listeyi rastgele oluşturabilmesi için yine sol panelde manuel liste oluşturmanın altında 3 adet QLineEdit yerleştirilmiştir. Bunlar sıralanacak listenin; minimum, maksimum değerlerinin ve eleman adedinin girileceği input alanlarıdır. Buradan girilen değerlere göre math modülü kullanılarak rastgele değerler üretilmektedir ve algoritmalarda kullanılacak listeye eklenmektedir. İki yöntemde de QLineEdit içerisinde tamsayı değerleri için PyQt5.QtGui içerisinde QIntValidator sınıfı kullanılarak validasyonları kontrol edilmektedir. Aynı zamanda tamsayının kaç karakter olacağı, minimum ve maksimum sınırları da belirtilmektedir.

#### D. Matplotlib Grafiğinin PyQt5 Arayüzünde Uygulanışı

Grafik kütüphanesi olan Matplotlib ile bir grafik oluşturulduğunda bunu "figure" olarak açmaktadır. PyQt5 ile masaüstü arayüzünde bu figürü entegre etmek için özel bir widget oluşturmak ve Matplotlib'in farklı sınıflarını kullanmak gerekmektedir. Bu işlemler için öncelikle QtDesigner'dan boş bir widget eklenmiştir ve "promote to" denilerek kod tarafında onu yönetecek "MplWidget" sınıfına bağlanmıştır. "MplWidget.py" adında bir dosya oluşturulmuştur. MplWidget içerisinde Matplotlib kütüphanesinden FigureCanvas, Figure ve Pyplot sınıfları

import edilmiştir. Oluşturulan Figure nesnesi FigureCanvas içerisine tanımlanmakta ve FigureCanvas nesnesi de arayüzdeki widget'a atanmaktadır. Böylelikle oluşturulacak grafiklerin arayüzdeki widget ile bağlantısı sağlanmış olmaktadır.

#### E. Sıralama Algoritmalarının Uygulanışı

Sıralama işleminde tercihe bağlı olacak şekilde Selection, Bubble, Insertion, Merge, Quick algoritmaları kullanılmıştır. Hangi algoritmanın seçileceği arayüz üzerindeki radio butonları ile kullanıcı tarafından seçilmektedir. Bütün algoritmalar karşılaştırma temeline dayanmaktadır. Selection-Bubble-Insertion ikili kıyaslama yaparak basit ve verimsiz şekilde sıralama gerçekleştirirken, Merge-Quick algoritmaları daha performanslı şekilde recursive mantığıyla sıralamaları gerçekleştirmektedir.

##### 1) Selection Sort

Seçme sıralaması, sıralanmamış liste içerisinde döngüler halinde öncelikle en küçük elemanları bularak sıralama işlemini gerçekleştirmektedir. Her iterasyonda sıralanmamış kısımdan itibaren en küçük eleman aranmaktadır. Zaman karmaşıklığı  $O(n^2)$ 'dir.

##### 2) Bubble Sort

Kabarcık sıralaması, en basit sıralama algoritmalarından biridir. Liste elemanlarını ikili şekilde kontrol ederek, yanlış sırada olanları yer değiştirme mantığıyla çalışmaktadır. Zaman karmaşıklığı  $O(n^2)$ 'dir.

##### 3) Insertion Sort

Ekleme sıralaması, en basit sıralama algoritmalarından biridir. Seçme sıralamasına benzer bir mantıkla en küçük elemanları öncelikli olarak bulup sıralama yapmaktadır. Selection ve Bubble algoritmaları gibi küçük boyutlu listeler için uygundur. Zaman karmaşıklığı  $O(n^2)$ 'dir.

##### 4) Merge Sort

Birleştirme sıralaması, önceki 3 algoritmadan farklı olarak recursive yani yinelemeli şekilde çalışarak sıralama yapmaktadır.

Divide(Böl)-Conquer(Yönet)-Combine(Birleştir) prensibine dayanmaktadır. Problem öncelikle en küçük parçalara bölünmekte ve bu halde sıralama yapıp ardından birleştirilmektedir. Sıralama ve birleştirme en küçük elemandan listenin tamamına yinelemeli şekilde devam etmektedir. Performansı, klasik sıralama algoritmalarına göre özellikle büyük boyutlu listeler için çok daha iyidir. Zaman karmaşıklığı  $O(n \log(n))$ 'dir.

##### 5) Quick Sort

Hızlı sıralama, birleştirme sıralamasına çok benzer mantıkta çalışmaktadır. Recursive ve divide-conquer-merge işlemleri yapılmaktadır. Farkı, sıralama boyunca bir pivot elemanı üzerinden gidilmesidir. İlk başta bir pivot elemanı seçilmektedir. Bu listenin ilk elemanı, son elemanı ya da rastgele bir elemanı seçilebilir. Sonrasında seçilen pivot elemanına göre listenin diğer elemanları kıyaslanır ve pivot elemanından küçük olanlar sola, büyük

olanlar sağı olacak şekilde elemanların yerleri değiştirilir ve tekrarlı olarak diğer elemanlar pivot olarak seçilip bu işleme devam edilir. Zaman karmaşıklığı  $O(n^2)$ 'dir.

#### F. Grafik Türlerinin Uygulanışı

Kullanıcı sıralamanın görselleştirilmesinde hangi grafik tipi kullanılacağını seçebilmektedir. Sıralama algoritmalarında olduğu gibi radio butonlar ile bu bilgi alınmaktadır. 3 farklı grafik türü bulunmaktadır. Bunlar: Scatter(Dağılım), Bar(Sütun), Stem(Kök) grafikleridir. Radio butonlar ile hangisinin kullanılacağı seçildikten sonra matplotlib sınıfından daha önce oluşturulmuş "canvas" elemanının "axes" üstünden "scatter", "plot", "stem" isimlerindeki metodlardan ilgili olan çağrılmaktadır. İçeriğinde parametre olarak hepsi aynı değerleri almaktadır. İlk parametre olarak x eksenini, ikinci parametre olarak y eksenini oluşturacak elemanların listesini almaktadırlar. Bu listeler ise sıralama algoritmasına tabi tutulan kullanıcının girdiği liste elemanlarıdır. X eksenini olarak sıralanacak/sıralanan listenin eleman adedi girilirken, y eksenini listesine liste elemanlarının değerleri girilmektedir. Programın devamında, grafikteki y eksenine karşılık gelen değerlerin değişmesi ile sıralama işleminin görselleştirilmesi sağlanmaktadır.

#### G. Animasyonun Oluşturulması

Programda, kullanıcının girdiği sıralanmamış listenin, sıralama algoritmalarından geçirilmesinin aşamalarını grafik üzerinde animasyon şeklinde gösterebilmek için, "TrackedArray" adında bir sınıf oluşturulmuştur. Bu sınıf, listenin, sıralama algoritmasındaki her bir aşamasını takip etmektedir. Sınıfın içerisinde "\_get\_" "\_set\_" magic metodları kullanılmıştır. Sıralama algoritmasına liste direkt gönderilmemektedir. Sıralanmamış liste önce TrackedArray'e parametre olarak gönderilip, bu sınıfın nesnesi, sıralama algoritmasına gönderilmektedir. Sıralama algoritmasında sıralanacak listeye her get ve set isteklerinde tanımlanmış magic metodlar tetiklenmekte ve listenin o anki değerleri yine TrackedArray içerisindeki bir listeye kaydedilmektedir. Bu sayede algoritmanın her bir aşamasında, daha sonra animasyonda kullanılmak üzere listenin aşama aşama tüm geçirdiği değişiklikler 2 boyutlu bir liste içerisinde tutulmaktadır. Animasyon işlemleri için daha önce bahsedilen "matplotlib.animation" modülünden import edilen "FuncAnimation" sınıfı kullanılmaktadır. Bu sınıfın içerisinde parametre olarak grafiğin çizileceği figür nesnesi, "AnimationUpdate" adında bir fonksiyon ve bu fonksiyonun kaç kez çalıştırılacağı, tamsayı olarak girilmektedir. Ayrıca interval parametresi ile animasyonun hızını float değer olarak belirtilebilmektedir. AnimationUpdate metodu belirtilen adet kadar çalıştırılmakta ve bu metod içerisinde grafik, her turda, listenin daha önce sıralama işleminde kaydedilmiş tüm aşamalarındaki değerlerine göre tekrar tekrar çizdirilmektedir. Bu sayede listenin sıralama aşamaları animasyon halinde görselleştirilmiş olmaktadır.

#### H. Animasyon Hızının Değiştirilebilmesi

Animasyonun hızının, başlangıçta veya animasyon esnasında dinamik bir şekilde değiştirilebilmesi için arayüze bir QSlider widget'ı eklenmiştir. Bu slider üzerinden kullanıcıyı barı sağı sola hareket ettirerek hız çarpanının değerini değiştirerek animasyon hızını yavaşlatıp, hızlandırabilmektedir. Buradan alınan input değeri bir değişkene aktarılarak, FuncAnimation içerisinde animasyon hızını belirleyen "interval" parametresinde, çarpan kısmına yazılmaktadır.

#### İ. Oluştur-Başlat-Durdur-Sıfırla Butonları ve İşlevleri

Programın çalışması boyunca grafiğin ve animasyonun başından sonuna kontrolü için 4 farklı buton oluşturulmuştur. Tüm butonlar başlangıç durumunda pasif durumdadır. Bu butonlardan ilki "Oluştur" butonudur. Oluştur butonu, kullanıcı manuel-rastgele bir liste girmeden aktifleşmemektedir. Diğer butonlar da aynı bu şekilde mantığa aykırı şekilde hareket etmeyecek şekilde tasarlanmış ve kontrolleri (Birinin aktifleşmesi ile diğer ilgili butonun kapatılması gibi...) yapılmıştır. Oluştur butonu aktifleştikten sonra basıldığında, ilgili metod tetiklenmekte ve girilen listenin sıralanmamış hali istenilen grafik türüne göre çizilmektedir. İkinci buton olan "Başlat" butonu, listeyi sıralama işlemine tabi tutmakta ve animasyonun başlamasını sağlamaktadır. "Durdur-Devam Et" butonları animasyon esnasında, animasyonu oluşturan "FuncAnimation" sınıfının nesnesini kullanarak, animasyonu durdurma ve kaldığı yerden devam ettirme işlemlerini gerçekleştirmektedir. 4. ve son buton olan "Sıfırla" butonu ise animasyon tamamen bittikten sonra aktifleştirilmekte ve basıldığında grafiği ve animasyonu başlangıç haline geri sıfırlamaktadır.

#### J. Qt Designer Üzerinden Tasarımın Yapılması

Arayüz tasarımında, daha önce de bahsedildiği gibi QtDesigner uygulaması kullanılmıştır. Arayüz elemanları, QVBoxLayout ve QHBoxLayout widget'ları ile yatayda ve dikeyde hizalanmaları sağlanmıştır. QtDesigner, stillendirme sayfası ile tüm elemanlara css uygulanabilmesini sağlamaktadır. Çoğu css kodu central widget adındaki bütün widget'ları kapsayan merkez widget üzerinden tanımlanmıştır. QtWidgets için, elemanlara özgü farklı seçiciler bulunmaktadır. Bu özel seçiciler ile QScrollArea, Slider gibi stillendirmesi zor olabilecek elemanlara css komutları tanımlanabilmektedir.

#### K. Karmaşıklık Analizinin Yazdırılması

Sıralama algoritmalarının her birinin karmaşıklık analizleri, arayüzde, grafiğin çizdirildiği widget'ın altına bir footer oluşturularak bunun içerisine yazdırılmıştır. Her bir algoritmanın karmaşıklık analizinde, en iyi, ortalama ve en kötü durumların formülleri ve altlarına sıralama işlemine sokulan listenin eleman adedine göre değerleri math modülü ile hesaplanıp yazdırılmaktadır.

#### L. Karşılaştırma Sayısının Yazdırılması

Karşılaştırma sayısı, algoritma karmaşıklığının sağına yazdırılmaktadır. Animasyon boyunca listenin sıralama algoritmasındaki karşılaştırma adedi anlık olarak yakalanıp yazdırılmaktadır. Animasyonun başında 0 iken bittiğinde toplam karşılaştırma sayısı ekranda gösterilmektedir. Bunu yakalayabilmek için sıralama algoritmalarında karşılaştırmanın yapıldığı yerlerde listenin o anki bir kopyası başka bir listeye kaydedilmektedir. Animasyon devam ederken, animasyon için kullanılan sıralama aşamalarındaki her bir değerini tutan listenin anlık bulunduğu “frame”den itibaren karşılaştırma anlarındaki listenin görüntüsü içerisinde arama yapılarak kaç adet görüntünün eşleştiği sayılıp yazdırılmaktadır. Bu animasyonun o anına kadar kaç karşılaştırma yapıldığını yakalamaktadır.

#### IV. KAYNAKLAR

- [1] <https://www.btkakademi.gov.tr/portal/course/sifirdan-ileri-seviye-python-programlama-5877>
- [2] <https://www.python.org/>
- [3] <https://pypi.org/>
- [4] <https://matplotlib.org/>
- [5] <https://doc.qt.io/>
- [6] <https://yapayzekalabs.blogspot.com/>
- [7] [https://www.youtube.com/watch?v=jUcEr\\_ZM9wY&list=PLOl6SW8nLgJx9guRvfyIVwrMXIginZhin&index=1](https://www.youtube.com/watch?v=jUcEr_ZM9wY&list=PLOl6SW8nLgJx9guRvfyIVwrMXIginZhin&index=1)
- [8] <https://code.visualstudio.com/>
- [9] <https://git-scm.com/>
- [10] <https://fonts.google.com/>
- [11] <https://stackoverflow.com/>
- [12] <https://www.w3schools.com/>
- [13] <https://www.geeksforgeeks.org/>
- [14] <https://tailwindcolor.com/>
- [15] <https://medium.com/>
- [16] <https://fontawesome.com/>
- [17] <https://bilgisayarkavramlari.com/2008/08/09/siralama-algoritmali-sorting-algorithms/>