



YAZILIM GELİŞTİRME LABORATUVARI

Dr. Öğr. Üyesi Adnan Sondaş

Öğr. Gör. Alper Metin

SIRALAMA ALGORİTMALARI GÖRSELLEŞTİRİCİSİ

Berra Oğuz - 201307028

Eren Torun - 201307008

Mert Gökyar - 201307051

SIRALAMA ALGORİTMALARI GÖRSELLEŞTRİCİSİ

*Yazılım Geliştirme Laboratuvarı-II

Berra Oğuz (201307028)
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
berraoguz14@gmail.com

Eren Torun (201307008)
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
beneyim@gmail.com

Mert Gökyar (201307051)
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
gokyarmert01@gmail.com

Özet—Sıralama algoritmaları, bir dizi veya koleksiyon içerisindeki öğeleri küçükten büyüğe ya da büyükten küçüğe doğru sıralamak için kullanılan matematiksel işlemlerdir. En sık kullanılan sıralama algoritmaları arasında kabarcık sıralama, seçim sıralaması, eklemeli sıralama, hızlı sıralama ve birleştirme sıralaması bulunmaktadır. Hangi algoritmanın kullanılacağına karar verilirken, sıralanacak verilerin boyutu ve türü gibi faktörler dikkate alınmalıdır. Bu algoritmalar, birçok farklı uygulama alanında yaygın olarak kullanılmaktadır. Örneğin veritabanı sorgulama, bilgisayar grafikleri, yapay zeka ve bilgisayar programlama gibi alanlarda yaygın olarak kullanılmaktadır.

Anahtar Kelimeler—Sıralama, algoritma, veri, programlama

Abstract—Sort annotations are conservative operations used to sort items in a series or collection from smallest to largest or largest to smallest. The most commonly used sort sorts include list sort, selection sort, insertion sort, quicksort, and merge sort. Factors such as the size and type of data to be sorted should be considered when deciding which values to use. These descriptions are widely used in many different application areas. For example, it is commonly used in fields such as dictionary querying, computer graphics, artificial intelligence and computer programming.

Keywords—Sorting, algorithm, data, programming

I. GİRİŞ

Sıralama algoritmaları görselleştiricisi, bir sıralama algoritmasının görsel olarak simüle edilmesine yardımcı olan bir araçtır. Bu araç, öğrencilerin ve programcıların sıralama algoritmalarını daha iyi anlamasına yardımcı olmaktadır. Görselleştirici, bir dizi elemanı farklı sıralama algoritmaları kullanarak sıralar ve adım adım değişen elemanları görsel olarak göstermektedir. Sıralama algoritmaları görselleştiricisi, farklı sıralama algoritmalarını karşılaştırmak ve hangisinin daha hızlı olduğunu görmek için de kullanılmaktadır. Sıralama algoritmaları görselleştiricisi, web tarayıcılarında veya özel programlarda kullanılmaktadır. Bazı görselleştiriciler, örnek verileri kullanarak sıralama algoritmalarının nasıl çalıştığını anlatan adım adım talimatlar sağlamaktadır.

Bu raporun konusu olan Sıralama Algoritmaları Görselleştiricisi kullanıcının belirlediği liste boyutuna göre rasgele bir liste oluşturmaktadır. Dilerse kullanıcı sıralanacak listeyi manuel olarak da girebilmektedir. Oluşturulan animasyonun da hızı ölçeklenebilir. Bu sayede kullanıcının sıralama algoritmalarını ve grafik türlerini kullanarak bir listenin sıralanmasını ve sıralama işleminin adımlarını görsel bir şekilde takip edebilmesini sağlamaktadır. Sıralama Algoritmaları Görselleştiricisi ile ilgili gerçekleştirilebilecek bütün işlemlerin yer bulacağı bu projemizi takip edebileceğiniz bu raporda sizlere tanıtacağız.

II. PROJE HAKKINDA

A. Arayüz Tasarımı

Projemizin arayüz tasarımını yapmak için Python programlama dilinde kullanılan bir GUI kütüphanesi olan Tkinter kütüphanesini kullandık. Tkinter, basit ve orta düzeyde kullanıcı arayüzleri oluşturmak için yaygın olarak tercih edilen bir seçenek olduğu için “Sıralama Algoritmaları Görselleştiricisi” konulu projemize en uygun arayüz tasarımının bu olduğuna karar verdik.

B. Proje Tanıtımı

Bu proje, kullanıcının sıralama algoritmalarını ve grafik türlerini kullanarak bir listenin sıralanmasını ve sıralama işleminin adımlarını görsel bir şekilde görmesini sağlamayı amaçlamaktadır. Kullanıcı arayüzü, sol panel ve ana panel olmak üzere iki bölümden oluşmaktadır.

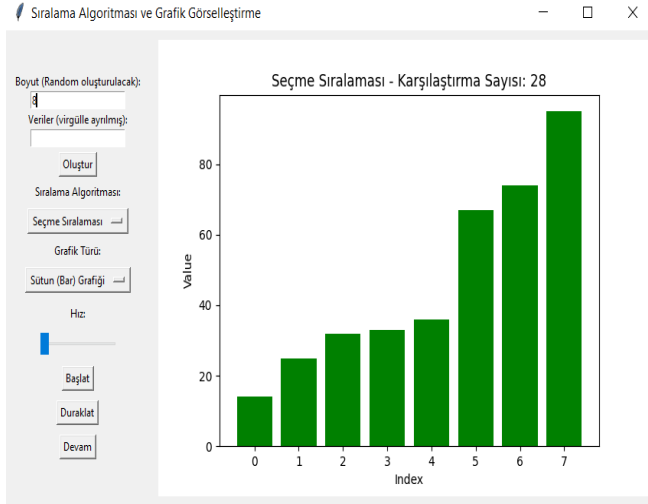
Sol panelde kullanıcı, sıralanacak listenin boyutunu belirleyebilmektedir veya listenin elemanlarını manuel olarak girebilmektedir. Ayrıca animasyon hızını ölçeklendirebilmektedir. Sıralama algoritmaları olarak Seçme Sıralaması, Kabarcık Sıralaması, Ekleme Sıralaması, Birleştirme Sıralaması ve Hızlı Sıralama seçenekleri sunulmaktadır. Grafik tipleri olarak Dağılım (Scatter) Grafiği, Sütun (Bar) Grafiği ve Kök (Stem) Grafiği seçenekleri bulunmaktadır. Ayrıca, Oluştur, Başla, Dur ve Sıfırla butonları kullanıcıya sunulmaktadır.

Ana panelde ise sol panelde yapılan seçimlere göre görsel bir arayüz oluşturulmaktadır. Seçilen algoritmaya göre

sıralama gerçekleştirilirken renk kodları kullanılarak anlaşılabilirlik artırılmaktadır. Sıralama adımları görsel olarak gösterilirken, karşılaştırma sayısı da arka planda tutulur ve belirli periyotlarla güncellenerek ana panelde gösterilmektedir. Sıralama işlemi tamamlandığında karşılaştırma sayısı ve algoritmanın karmaşık analizi sonucu ekrana yazdırılmaktadır.

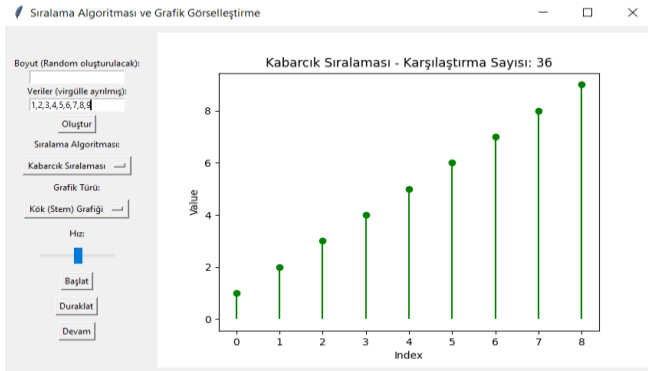
Bu proje, kullanıcılara sıralama algoritmalarının nasıl çalıştığını ve sıralama işleminin adımlarını görsel olarak anlama imkanı sunarak, sıralama algoritmaları hakkında daha iyi bir anlayış kazanmalarını sağlamayı hedeflemektedir. Ayrıca, farklı grafik türleriyle görsel olarak verileri temsil etme yeteneği de sunarak kullanıcılara esneklik sağlamaktadır.

C. Görsellerle Tanıtım



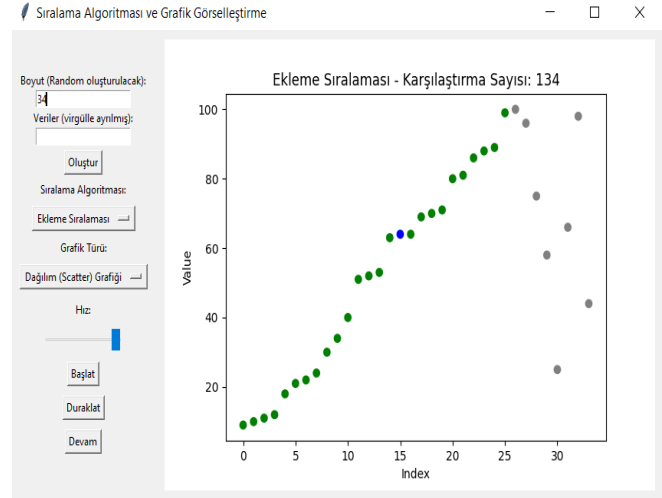
Şekil 1

Sıralama Algoritması ve Grafik Görselleştirme projemizde Şekil 1’de görüldüğü gibi sol panelinde Boyut (random oluşturur.), Veriler(verileri manuel bir şekilde elimizle bir çok sayı gireriz ve virgüllerle ayırırız.). İstedğimiz Sıralama Algoritmasını (Seçme Sıralaması, Kabarcık Sıralaması, Ekleme Sıralaması Birleştirme Sıralaması, Hızlı Sıralama) ve istediğimiz Grafik Türünü (Sütun Grafiği, Dağılım Grafiği, Kök Grafiği) seçtikten sonra Oluştur ve Başlat butonuna tıklarız. Grafik görselleştirken hızını öncesinde ya da o sırada ayarlayabilir, duraklatabilir ve daha sonrasında devam edebiliriz. Grafiğimiz görselleştikten sonra yukarıda Karşılaştırma Sayısı bilgisini de görebiliriz.



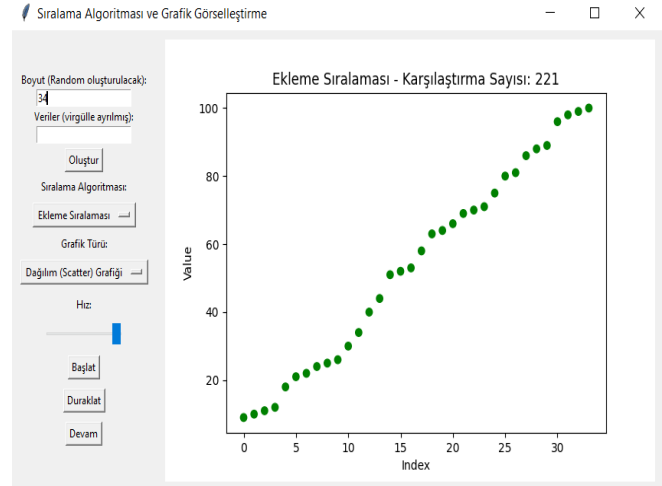
Şekil 2

Şekil 2’de de görüldüğü üzere (Kabarcık Sıralaması ve Kök Grafiği seçilmiştir.) Hızını arttırdık ve Veriler kısmından manuel bir şekilde grafikte görmek istediğimiz verileri girdik. Karşılaştırma Sayısı da 36 olarak görünüyor.



Şekil 3

Ekleme Sıralaması ve Dağılım Grafiğini seçtikten sonra grafiğimizin oluşturulma anı Şekil 3’te görülüyor.



Şekil 3.1

Ekleme Sıralaması ve Dağılım Grafiği bitmiş hali Şekil 3.1’de görülüyor.

D. Karşılaşılan Sorunlar

- Seçtiğimiz merge sort’ta color array’i create_graph_carray ile yazdırırken merge dolayısıyla veri bölündüğü için color array’in uzunluğundan dolayı sıkıntı çıktı. Bu durumu çözmek için de if len(color_array) != len(data): color_array = color_array[:len(data)] #Renk dizisini veri dizisiyle aynı boyuta getiriyoruz. Bu sayede de boyutları eşleştirmiş oluyoruz.

E. Faydalar

- Algoritmaların çalışma prensiplerini anlama: Görselleştirici, sıralama algoritmalarının adımlarını ve işleyişini görsel olarak gösterir. Bu sayede kullanıcılar, algoritmaların nasıl çalıştığını daha iyi anlayabilir ve mantığını kavrayabilir.

- Görsel takip imkanı: Görselleştirici, algoritmanın her adımını görsel olarak sunar. Kullanıcılar, adımları takip ederek her aşamada neler olduğunu görebilir ve algoritmanın nasıl ilerlediğini görsel olarak deneyimleyebilir.
- Hata tespiti ve verimlilik analizi: Görselleştirici, algoritmanın her aşamasında verilerin nasıl değiştiğini ve yer değiştirdiğini gösterir. Bu sayede kullanıcılar, olası hataları veya verimlilik sorunlarını daha iyi tespit edebilir ve algoritmanın performansını analiz edebilir.
- Karşılaştırmalı performans analizi: Görselleştirici, farklı sıralama algoritmalarının karşılaştırmalı performansını gösterir. Kullanıcılar, görsel olarak farklı algoritmaların nasıl çalıştığını ve hangi algoritmanın daha hızlı veya daha verimli olduğunu görebilir.
- Öğrenme kolaylığı: Görselleştirici, sıralama algoritmalarını öğrenen öğrencilere ve ilgilenen herkese öğrenme sürecini kolaylaştırır. Görsel olarak adımları takip etmek, algoritmaları daha iyi anlamayı ve öğrenmeyi sağlar.
- Esneklik ve özelleştirme: Görselleştirici genellikle kullanıcı dostu bir arayüz sunar ve kullanıcılara farklı seçenekleri özelleştirme imkanı tanır. Örneğin, listenin boyutunu veya öğelerini belirleme, grafik türünü seçme gibi seçenekler kullanıcıya sunulur.
- Görsel sunum ve paylaşım imkanı: Görselleştirici, sıralama algoritmalarının çalışma mantığını ve sonuçlarını görsel olarak sunar. Bu sayede kullanıcılar, algoritmanın çalışmasını ve sonuçları daha etkili bir şekilde sunabilir ve paylaşabilirler.

Bu nedenlerden dolayı, Sıralama Algoritmaları Görselleştiricisi, sıralama algoritmalarının daha iyi anlaşılmasını, öğrenme sürecini kolaylaştırmasını ve performans analizi yapılmasını sağlayarak kullanıcılara birçok fayda sağlar.

III. KULLANILAN TEKNOLOJİLER

A. PYTHON

Yüksek seviyeli, genel amaçlı bir programlama dilidir. Basit ve anlaşılır sözdizimiyle hızlı ve verimli bir yazılım geliştirme süreci sunmaktadır. Python, çeşitli veri tipleriyle çalışmayı kolaylaştırmaktadır ve geniş bir standart kütüphaneye sahiptir. Platform bağımsız olması, Python programlarının farklı işletim sistemlerinde çalışabilmesini sağlamaktadır. Veri analizi, yapay zeka, web geliştirme, bilimsel hesaplama, oyun geliştirme, otomasyon ve ağ programlama gibi birçok alanda yaygın olarak kullanılan Python, çok yönlülüğü ve kullanıcı dostu yapısıyla programcılara geniş bir uygulama alanı sunmaktadır.

B. TKINTER

Python'un standart kütüphanesinde yer alan bir GUI (Grafiksel Kullanıcı Arayüzü) kitaplığıdır. Tkinter, Python ile masaüstü uygulamaları geliştirmek için kullanılan en popüler ve yaygın olarak kullanılan araçtır. Tkinter, kullanıcı dostu bir arayüz sağlamaktadır ve kolay öğrenilebilir bir API sunmaktadır. Platform bağımsız olması, farklı işletim sistemlerinde çalışabilen uygulamalar geliştirmeyi mümkün kılmaktadır. Tkinter, çeşitli widget'larla (pencere, düğme, etiket, giriş kutusu vb.) birlikte gelir ve bu widget'lar uygulamanın görünümünü ve davranışını özelleştirmek için kullanılabilir. Tkinter, diğer Python kütüphaneleri ve araçları ile entegre çalışabilmektedir, bu da daha karmaşık uygulamaların geliştirilmesini sağlamaktadır. Overall, Tkinter, Python ile GUI uygulamaları geliştirmek için kullanışlı, güçlü ve esnek bir araçtır.

C. MATPLOTLIB

Python'da popüler bir veri görselleştirme kütüphanesidir ve grafikler, diyagramlar ve görsel öğelerle verileri temsil etmek için kullanılmaktadır. Çizgi grafikleri, bar grafikleri, scatter plotlar, histogramlar ve daha fazlası gibi geniş bir grafik türü yelpazesini desteklemektedir. Matplotlib, esnek ve özelleştirilebilir bir API sunmaktadır, böylece grafikleri renkler, çizgi stilleri, etiketler, eksenler ve başlıklar gibi özellikleriyle kişiselleştirebilmektedir. İnteraktif veya statik grafikler oluşturabilir ve grafikleri çıktı dosyaları olarak kaydedebilmektedir. Veri analizi, bilimsel hesaplama, makine öğrenimi, finansal analiz ve daha birçok alanda kullanılan Matplotlib, diğer Python kütüphaneleriyle de uyumlu çalışabilmektedir ve geniş bir topluluğa ve belgelendirme kaynaklarına sahiptir. Overall, Matplotlib, verileri görselleştirmek için güçlü ve esnek bir araçtır ve Python programcılarının grafik oluşturma ihtiyaçlarını karşılamada önemli bir rol oynamaktadır.

D. GITHUB

Dağıtık bir versiyon kontrol platformudur ve yazılım geliştirme projeleri için kullanılan bir hizmettir. Geliştiriciler, kaynak kodlarını saklamak, paylaşmak, işbirliği yapmak ve sürüm geçişini yönetmek için GitHub'ı tercih etmektedir. GitHub, git adı verilen versiyon kontrol sistemi üzerine kurulmuştur ve projelerin merkezi sunucularda depolanmasını sağlamaktadır. İşbirliğini teşvik etmek için çeşitli özellikler sunan GitHub, geliştiricilerin projelerini paylaşmasını, geri bildirim almasını ve projeleri birlikte geliştirmesini sağlamaktadır. Grafik ve istatistiklerle sürüm geçişini görselleştirebilir, eşzamanlı çalışmayı kolaylaştıran dal ve birleştirme özelliklerine sahiptir. Açık kaynak projelerin yaygın olarak paylaşıldığı bir platform olarak da kullanılan GitHub, entegrasyon seçenekleriyle diğer araçlarla birlikte çalışabilmektedir. Tüm bu özellikleriyle GitHub, yazılım geliştirme süreçlerini daha verimli hale getirir ve projelerin daha iyi yönetilmesini sağlar.

IV. GENEL YAPI VE MİMARİ

Projemizde programlama dili olarak Python kullandık. Projemizin arayüz tasarımını ise Python programlama dilinde kullanılan bir GUI kitaplığı olan Tkinter kitaplığını kullandık. Tkinter kitaplığı sayesinde projemize orta düzeyde ve anlaşılır bir kullanıcı arayüzü oluşturduk.

Projemizde kullanılan sıralama algoritmaları arasında Seçme Sıralaması (Selection Sort), Kabarcık Sıralaması (Bubble Sort), Ekleme Sıralaması (Insertion Sort), Birleştirme

Sıralaması (Merge Sort) ve Hızlı Sıralama (Quick Sort) yer almaktadır.

Projemizde matplotlib kütüphanesi ve matplotlib.animation modülü de kullanılmıştır. Bu kütüphane ve modüller, sıralama işlemini görselleştirmek için grafikler oluşturmaktadır.

Kodda, Tkinter penceresi ve arayüzü oluşturulur ve kullanıcıya sıralama algoritmasını ve grafik türünü seçme imkanı verilir. Ardından, seçilen algoritma ve verilerle sıralama işlemi gerçekleştirilir ve sonuç görselleştirilir. Kodun tam çalışması için Tkinter ve matplotlib kütüphanelerinin yüklü olması gerekmektedir.

get_speed() fonksiyonu, hız kaydırıcısından alınan değere göre sıralama hızını belirler. Hız, 11 ile 1 arasında bir değerdir ve -0.01 ile çarpılarak sıralama hızına dönüştürülür.

selection_sort() fonksiyonu, seçme sıralaması algoritmasını uygular. Bu algoritma, her adımda sıralanmış olmayan veri kümesinden en küçük elemanı seçer ve uygun konuma yerleştirir.

bubble_sort() fonksiyonu, kabarcık sıralaması algoritmasını uygular. Bu algoritma, komşu elemanları karşılaştırarak sıralanmamış elemanları birbirleriyle yer değiştirir ve en büyük elemanın sağa doğru kaydığını garanti eder.

insertion_sort() fonksiyonu, yerleştirme sıralaması algoritmasını uygular. Bu algoritma, bir elemanı sıralı bölgeye yerleştirerek sıralanmamış elemanları sıralı bölgede doğru konuma yerleştirir.

merge_sort_with_color() fonksiyonu, renklendirilmiş birleştirme sıralaması algoritmasını uygular. Bu algoritma, veri kümesini rekürsif olarak böler, ardından birleştirerek sıralar.

merge_sort_helper() fonksiyonu, birleştirme sıralamasının yardımcı fonksiyonudur. Veri kümesini ikiye böler ve her bir yarıyı ayrı ayrı sıralamak için kendini çağırır. Ardından, iki sıralı yarıyı birleştirerek sonuç verir.

merge() fonksiyonu, birleştirme sıralamasında iki sıralı alt diziyi birleştirir.

quicksort_with_color() fonksiyonu, renklendirilmiş hızlı sıralama algoritmasını uygular. Bu algoritma, bir pivot eleman seçerek veri kümesini ikiye böler, ardından pivotun sağında ve solunda rekürsif olarak sıralar.

V. SONUÇLAR

Geliştirilen proje kapsamında kullanıcılar, sıralama algoritmalarını ve grafik türlerini kullanarak bir listenin sıralanmasını ve sıralama işleminin adımlarını görsel olarak görebilecekler. Projenin sol panelinde, kullanıcılar listenin boyutunu belirleyebilir, manuel olarak bir liste girebilir veya rasgele bir liste oluşturabilirler. Ayrıca animasyon hızını ayarlayabilirler. Seçilebilen sıralama algoritmaları arasında Seçme Sıralaması, Kabarcık Sıralaması, Ekleme Sıralaması, Birleştirme Sıralaması ve Hızlı Sıralama bulunmaktadır. Grafik tipleri olarak ise Dağılım (Scatter) Grafiği, Sütun (Bar) Grafiği ve Kök (Stem) Grafiği seçilebilir. Oluştur, Başla, Dur ve Sıfırla butonlarıyla kullanıcılar arayüzü yönetebilirler. Ana panelde seçilen algoritma ve grafik türüne göre görsel bir arayüz sunulur. Sıralama işlemi sırasında yapılan karşılaştırma sayısı takip edilir ve belirli periyotlarla güncellenerek gösterilir. Sıralama işlemi tamamlandığında karşılaştırma sayısı ve algoritmanın karmaşık analizi sonuçları ekrana yazdırılır. Bu projenin sonucunda kullanıcılar, sıralama algoritmalarını daha iyi anlayabilir ve görsel bir şekilde sıralama işlemini takip edebilirler.

VI. KAYNAKÇA

- [1] [Python](#) (20/05/2023)
- [2] [Python Tutorial \(w3schools\)](#) (20/05/2023)
- [3] [Tkinter Tutorial](#) (21/05/2023)
- [4] [Git Tutorial](#) (21/05/2023)
- [5] [Python GUI Programming](#) (22/05/2023)
- [6] [Matplotlib Tutorial](#) (22/05/2023)