

Yazılım Geliştirme Lab.

Proje-2

Selim Metehan AKTAŞ - 201307063

Batuhan KANBER - 201307048

Mehmet Berk BAYTÜRK - 201307044

Bilişim Sistemleri

Mühendisliği

Kocaeli Üniversitesi

Problem Tanımı

Bu proje, kullanıcıların bir listenin sıralama işlemini gözlemlemek için sıralama algoritmalarını ve çeşitli grafik türlerini kullanmalarını sağlayan bir görsel arayüz sunmayı amaçlamaktadır. Proje, kullanıcılara bir liste girmek veya belirli bir boyutta rasgele bir liste oluşturmak için manuel olarak izin verir. Arayüz, boyut girişi, animasyon hızı kontrolü, sıralama algoritmaları seçimi, grafik türleri seçimi ve görselleştirmenin başlatılması, durdurulması ve sıfırlanması gibi çeşitli bileşenleri içerir.

Anahtar Kelimeler— Sıralama algoritmaları, görsel grafik türleri , görselleştirme, Python, Tkinter

1.1. Python

Python, genel amaçlı, yüksek seviyeli, etkileşimli ve yorumlanabilir bir programlama dilidir. Guido van Rossum tarafından geliştirilmeye başlanan Python, basit ve anlaşılır sözdizimiyle öne çıkar. Dilin tasarımı okunabilirlik ve kullanılabilirlik üzerine odaklanırken, güçlü bir dil olmasını da sağlamaktadır. Python, geniş bir standart kütüphane ve birçok üçüncü taraf modülle birlikte gelir. Bu modüller, çeşitli alanlarda işlevsellik sağlar ve hızlı bir şekilde uygulama geliştirmeyi destekler. Python ayrıca açık kaynaklı bir dildir, bu da geliştiricilerin kaynak kodlarına erişip değiştirme veya kendi projelerinde kullanma özgürlüğüne sahip olmalarını sağlar. Python 3.x, Python programlama dilinin en son sürüm ailesini ifade eder. Python 3.x sürümleri, Python 2'nin yerini alarak dilin geliştirilmiş ve daha tutarlı bir versiyonunu sunar. Python 3.x, dilin temizlenmesi, Unicode desteği, geliştirilmiş hata ayıklama, güncellenmiş kütüphaneler ve bir dizi yeni özellik içerir. [1]

1.2. Tkinter

Tkinter, Python programlama dilinin standart kütüphanesinde bulunan bir modüldür. Tkinter, GUI (Graphical User Interface - Grafiksel Kullanıcı Arayüzü) uygulamaları oluşturmak için kullanılan bir araçtır. Tkinter, kullanıcıların pencere, düğme, metin kutusu, etiket, liste kutusu gibi grafiksel bileşenleri kullanarak interaktif ve görsel uygulamalar geliştirmesini sağlar.

Tkinter, Tk GUI toolkit'ini (Tkinter = Tk + Interface) Python programlama diline bağlayan bir arayüzdür. Tkinter, Tk'nın işlevlerine erişimi kolaylaştırır ve Python ile birlikte kullanıldığında hızlı bir şekilde GUI uygulamaları geliştirmeyi sağlar.

Tkinter, kullanıcı arayüzü bileşenlerini oluşturma, düzenleme, konumlandırma ve etkileşimlere tepki verme gibi işlevleri sağlar. Ayrıca, Tkinter uygulamaları için olay işleme, renklendirme, grafikler ve kullanıcı girişi gibi çeşitli özellikler sunar.[2]

1.3. Visual Studio Code

VS Code, bir projeyi yönetmek, kod yazmak, hata ayıklamak, sürüm kontrolü yapmak ve diğer geliştirme işlemlerini kolaylaştıran bir dizi araç sunar. Bunlar arasında otomatik tamamlama, kod renklendirme, hata ayıklama, otomatik biçimlendirme, git entegrasyonu ve uzantılarla genişletilebilirlik gibi özellikler bulunur.

VS Code, geniş bir programlama dilini destekler ve kullanıcıların dil desteğini genişletmek için çeşitli eklentileri kullanmasına izin verir. Özellikle web geliştirme alanında popülerdir ve HTML, CSS, JavaScript gibi diller için özel destek sunar. Ayrıca, Python, C++, Java, Go, Ruby ve daha birçok programlama dili için de geliştirme ortamı olarak kullanılabilir.

VS Code, kullanıcı dostu bir arayüze sahiptir ve özelleştirilebilir düzenleme seçenekleri sunar. Kullanıcılar, tema ve düzen ayarlarını kişiselleştirerek çalışma alanlarını istedikleri şekilde düzenleyebilirler.[3]

2.1.Sıralama Algoritmaları

Seçme Sıralaması (Selection Sort): Seçme sıralaması, bir diziyi sıralamak için kullanılan basit bir sıralama algoritmasıdır. Algoritma, dizideki en küçük veya en büyük elemanı bulup sıralı bölümün başına yerleştirerek çalışır. Bu işlem, sıralanmamış bölümdeki elemanların tamamı sıralanana kadar tekrarlanır. Seçme sıralaması, zaman karmaşıklığı açısından $O(n^2)$ olduğu için büyük veri setleri için genellikle tercih edilmeyen bir algoritmadır.

Kabarcık Sıralaması (Bubble Sort): Kabarcık sıralaması, bir diziyi sıralamak için kullanılan bir sıralama algoritmasıdır. Algoritma, komşu elemanları karşılaştırarak ve gerektiğinde yerlerini değiştirerek çalışır. Büyük elemanlar yavaş yavaş dizinin sonuna doğru "kabarcık" gibi yükselirken, küçük elemanlar da aynı şekilde dizinin başına doğru ilerler. Bu işlem, dizide hiç yer değişikliği yapılmayana kadar veya sıralı bölümün son elemanına ulaşılanaya kadar tekrarlanır. Kabarcık sıralamasının zaman karmaşıklığı $O(n^2)$ olduğu için büyük veri setlerinde verimli olmayabilir.

Ekleme Sıralaması (Insertion Sort): Ekleme sıralaması, bir diziyi sıralamak için kullanılan bir sıralama algoritmasıdır. Algoritma, bir elemanı sıralı bölüme doğru yerleştirerek çalışır. Başlangıçta, ilk eleman sıralı bölümdür ve sıralanmamış bölümdeki elemanlar sırasıyla alınarak doğru konumlarına yerleştirilir. Bu işlem, sıralanmamış bölümdeki elemanlar tamamı sıralanana kadar tekrarlanır. Ekleme sıralaması, neredeyse sıralı veya küçük veri setleri için iyi performans gösterirken, zaman karmaşıklığı $O(n^2)$ olduğu için büyük veri setlerinde verimli olmayabilir.

Birleştirme Sıralaması (Merge Sort): Birleştirme sıralaması, bir diziyi sıralamak için kullanılan bir sıralama algoritmasıdır. Algoritma, diziyi rekürsif olarak ikiye böler, ardından bu alt dizileri sıralar ve son olarak sıralı alt dizileri birleştirir. Birleştirme sıralaması, sıralanmış alt dizileri birleştirme işlemi sırasında karşılaştırarak çalışır. Bu işlem, dizinin tamamı sıralanana kadar devam eder. Birleştirme sıralamasının zaman karmaşıklığı $O(n \log n)$ olduğu için büyük veri setleri üzerinde daha iyi performans gösterir.

Hızlı Sıralama (Quick Sort): Hızlı sıralama, bir diziyi sıralamak için kullanılan bir sıralama algoritmasıdır. Algoritma, bir elemanı "pivot" olarak seçer, daha küçük elemanları sol tarafa, daha büyük elemanları ise sağ tarafa yerleştirir. Ardından, sol ve sağ alt dizileri aynı işleme tabi tutarak sıralama işlemini tekrarlar. Hızlı sıralama, veriyi bölerek ve parçalayarak çalışır. Bu işlem, alt diziler tek elemandan oluşana kadar veya sıralı hale

gelene kadar devam eder. Hızlı sıralamanın ortalama zaman karmaşıklığı $O(n \log n)$ olduğu için genellikle diğer sıralama algoritmalarına göre daha hızlıdır. Ancak, en kötü durumda (örneğin, sıralı bir dizi için) zaman karmaşıklığı $O(n^2)$ olabilir. [4]

2.2. Grafik Tipleri

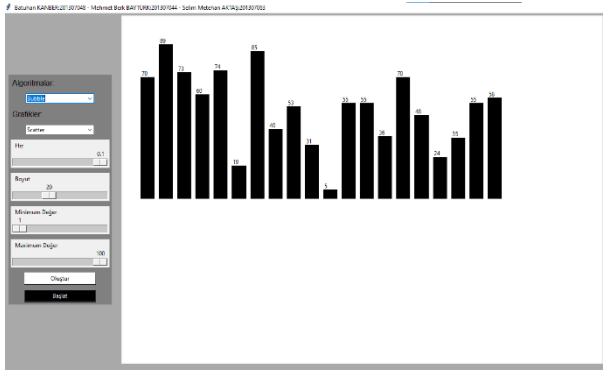
Dağılım (Scatter) Grafiği: Dağılım grafiği, veri noktalarının x ve y koordinatlarına dayalı olarak bir düzlemde gösterildiği bir grafik türüdür. Her veri noktası, grafiğin üzerinde bir nokta olarak temsil edilir.

Sütun (Bar) Grafiği: Sütun grafiği, kategorik verilerin farklı kategoriler arasındaki karşılaştırmalarını göstermek için kullanılan bir grafik türüdür. Her kategori, dikdörtgen sütunlarla temsil edilir ve sütunların yükseklikleri, ilgili kategorinin değerini gösterir.

Kök (Stem) Grafiği: Kök grafiği, genellikle zaman serileri veya sayısal verilerin dağılımlarını görselleştirmek için kullanılan bir grafik türüdür. Veri noktaları, düz bir çizgi üzerinde dikey çizgilerle temsil edilir.[5]

3.1. Proje Deneyimleri

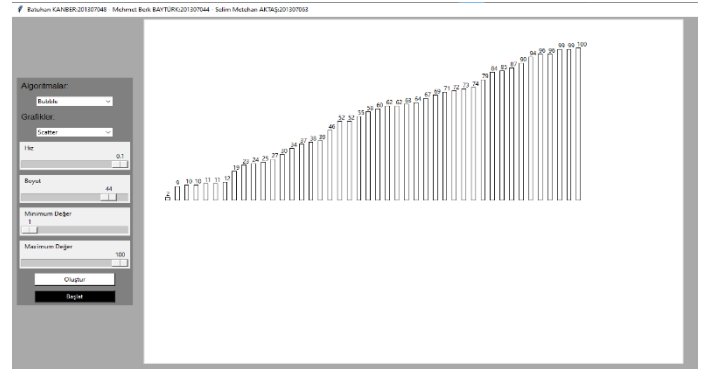
Görsel 3.1.1 Kullanıcının Seçim Yaptığı Panel



Görsel 3.1.2 Karışık Haldeki Değerler

4.1 KAZANIMLAR

Bu proje, kullanıcılara sıralama algoritmalarını görsel olarak deneyimleme ve algoritmaların çalışma mantığını daha iyi anlama fırsatı sunan bir arayüz sağlamayı amaçlamaktadır. Kullanıcılar, projeyi kullanarak sıralama adımlarını grafikler aracılığıyla takip edebilirler.



Görsel 3.1.3 Sıralanmış Haldeki Değerler

5.1 REFERANSLAR

- [1] <https://www.geeksforgeeks.org/python-gui-tkinter/>
- [2] <https://wiki.python.org/moin/PythonBooks>
- [3] <https://www.w3schools.com/python/>
- [4] <https://www.geeksforgeeks.org/sorting-algorithms/>
- [5] https://en.wikipedia.org/wiki/Sorting_algorithm