“PYTHON PROGRAMLAMA DİLİ İLE VERİ YAPILARI VE SIRALAMA ALGORİTMALARININ GÖRSELLEŞTİRİLMESİ”

Batuğ Can AKPINAR   
*201307002*  
*Kocaeli Üniversitesi*İstanbul, Ümraniye  
batugcann@gmail.com

Ömer MERCAN   
*201307022*  
*Kocaeli Üniversitesi*Kocaeli, İzmit  
omermercan41@gmail.com

**Özet-Bu proje, Python programlama dilini kullanarak veri yapıları ve sıralama algoritmalarının uygulanmasını amaçlamaktadır. Proje, kullanıcıya çeşitli sıralama algoritmalarını görselleştirme imkanı sunan bir görselleştirme aracı geliştirmeyi hedeflemektedir. Bu araç, kullanıcının farklı sıralama algoritmalarını gerçek zamanlı olarak gözlemlemesine ve analiz etmesine olanak tanır. Projenin amacı, öğrencilerin ve programcı adaylarının sıralama algoritmalarını daha iyi anlamalarına yardımcı olmaktır.**

**Anahtar Kelimeler:**

***Anahtar Kelimeler — ;*** ***Python, veri yapıları, sıralama algoritmaları, görselleştirme, gerçek zamanlı.***

GİRİŞ

Bu proje raporu, "Python Programlama Dili ile Veri Yapıları ve Sıralama Algoritmalarının Görselleştirilmesi" başlıklı çalışmanın sunumunu içermektedir. Bu çalışmanın amacı, Python programlama dili kullanılarak veri yapıları ve sıralama algoritmalarının görselleştirilmesini sağlamak ve bu konuda öğrenme sürecini desteklemektir. Çalışmanın kapsamı, veri yapılarının oluşturulması, sıralama algoritmalarının uygulanması ve sonuçların görsel olarak temsil edilmesi üzerine odaklanmıştır.

Bu çalışmanın önemi ve motivasyonu, veri yapıları ve sıralama algoritmalarının anlaşılması ve uygulanmasının önemini vurgulamaktadır. Bu alanlardaki eksiklikler ve zorluklar, yazılım geliştiricilerin karşılaşabileceği yaygın sorunlardır. Bu projenin amacı, bu zorlukları aşmada yardımcı olmak, algoritmaları görselleştirerek öğrenme sürecini iyileştirmektir.

Bu proje kapsamında, hedeflenen sonuçlar şunlardır:

1. Veri yapılarının oluşturulması ve temsil edilmesi.
2. Sıralama algoritmalarının uygulanması ve adımların görselleştirilmesi.
3. Algoritmaların performansının analiz edilmesi ve karşılaştırılması.
4. Öğrenme sürecinin iyileştirilmesi için etkileşimli bir kullanıcı arayüzünün sunulması.

Bu çalışmanın elde edeceği sonuçlar, yazılım geliştiricilerin veri yapıları ve sıralama algoritmalarını daha iyi anlamalarına ve uygulamalarına katkı sağlayacaktır. Ayrıca, görsel temsillerin öğrenme sürecini artırdığı gözlemlenecektir.

Projenin gerçekleştirilmesi sırasında, bazı zorluklarla karşılaşılmış ve çeşitli teknik gereksinimler karşılanmıştır. Özellikle, veri yapılarının ve algoritmaların doğru bir şekilde görselleştirilmesi için uygun grafik kütüphanelerinin seçimi ve kullanımı önemliydi. Bu zorluklar, projenin gelişim sürecine değerli bir deneyim katmış ve teknik becerilerin geliştirilmesine yardımcı olmuştur.

LİTERATÜR TARAMASI

Literatür taraması, benzer çalışmaların incelenerek mevcut bilgi birikimi ve açıkların belirlendiği önemli bir adımdır. Bu bölümde, projenin yapıldığı alanda daha önce yapılmış çalışmaların özeti ve karşılaştırması, kullanılan yöntemler ve teknikler ile mevcut açıklar ve problemler ayrıntılı bir şekilde ele alınacaktır.

Benzer çalışmaların özeti ve karşılaştırmasıyla başlayacak olursak, proje kapsamında incelenen alanda yapılan diğer çalışmaların literatürdeki yerini belirlemek önemlidir. Bu sayede, benzer çalışmaların bulguları ve yöntemleri hakkında bilgi sahibi olunabilir. Benzer çalışmaların analizi, projenin mevcut durumunu ve katkılarını değerlendirmek için önemli bir adımdır.

Kullanılan yöntemler ve teknikler ise projenin temelini oluşturan unsurlardır. Bu bölümde, projede kullanılan yöntemlerin detaylı bir açıklaması yapılır ve kullanılan tekniklerin neden tercih edildiği anlatılır. Bu sayede, projenin bilimsel temeli ve yöntemsel yaklaşımı okuyucuya aktarılır. Ayrıca, kullanılan yöntemlerin avantajları ve dezavantajları da değerlendirilebilir.

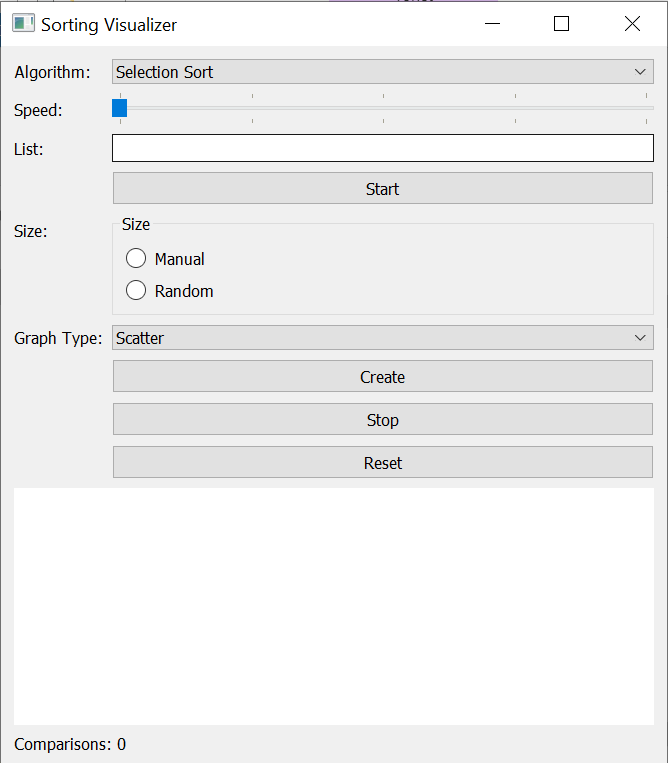
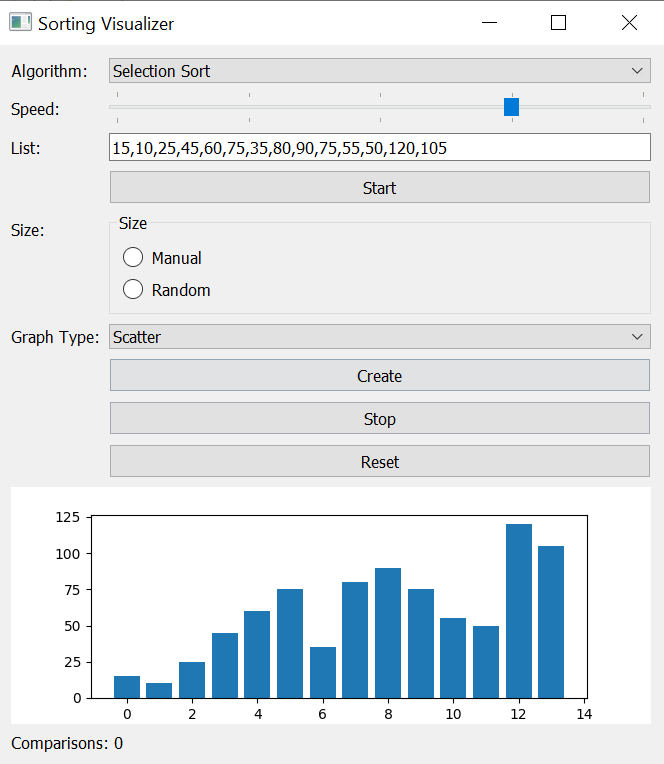
Mevcut açıklar ve problemler ise projenin yapılma motivasyonunu ve amacını belirleyen unsurlardır. Bu bölümde, alanda mevcut olan açıklar ve problemler detaylı bir şekilde ele alınır. Bu sayede, projenin neden önemli olduğu ve nasıl bir katkı sağlayabileceği daha iyi anlaşılır. Ayrıca, mevcut açıklar ve problemler üzerinde yapılan çalışmalara da değinilebilir ve projenin bu açıkları nasıl kapatabileceği vurgulanabilir.

Literatür taraması bölümü, projenin teorik temelini oluşturur ve mevcut bilgi birikimini sistematik bir şekilde sunar. Bu bölümün doğru bir şekilde yapılandırılması ve mevcut çalışmalara adil bir şekilde yer verilmesi, projenin güvenilirliğini artırır. Ayrıca, mevcut açıkların ve problemlerin belirlenmesi, projenin katma değerini ortaya koyar ve gelecekteki çalışmalara ilham verir.

Sonuç olarak, literatür taraması bölümü, benzer çalışmaların özeti ve karşılaştırması, kullanılan yöntemler ve teknikler ile mevcut açıklar ve problemlerin ele alındığı önemli bir bölümdür. Bu bölüm, projenin teorik temelini oluşturur ve projenin amacını, önemini ve beklenen katkılarını vurgular.

YÖNTEM

Proje Tasarımı ve Bileşenleri: Bu projede, Sorting Visualizer uygulamasının geliştirilmesi için aşağıdaki bileşenler kullanılmıştır:

**1.Kullanıcı Arayüzü (UI):** Proje, kullanıcıların uygulamayı etkileşimli bir şekilde kullanmalarını sağlamak için bir grafik kullanıcı arayüzü (GUI) içermektedir. UI, kullanıcılara sortlama algoritmalarını seçme, dizi boyutunu belirleme, sortlama işlemini başlatma ve duraklatma gibi işlevleri sunmaktadır. Şekil-1

**2.Sorting Algoritmaları:** Proje, çeşitli sortlama algoritmalarını uygulamak ve görselleştirmek için kullanılan sortlama algoritmaları modülünü içermektedir. Bu modül, kullanıcının seçtiği algoritmayı kullanarak bir diziyi sıralamak için gerekli algoritma adımlarını gerçekleştirir.

**3.Veri Yapıları:** Proje, sortlama algoritmalarının etkili bir şekilde çalışabilmesi için çeşitli veri yapılarını içermektedir. Bu veri yapıları, dizi elemanlarını depolamak, yerlerini değiştirmek ve sıralama adımlarını takip etmek için kullanılır.

Uygulama Aşamaları ve Süreç: Proje aşağıdaki adımları izleyerek gerçekleştirilmiştir:

**1.Kullanıcı Arayüzünün Tasarımı:** İlk olarak, kullanıcı arayüzü (UI) Microsoft Visual Studio gibi bir entegre geliştirme ortamında tasarlanmıştır. UI, sortlama algoritmalarını seçmek, dizi boyutunu belirlemek, sortlama işlemini başlatmak ve duraklatmak gibi işlevleri kullanıcılara sunmaktadır.

**2.Sorting Algoritmalarının Implementasyonu:** Proje, çeşitli sortlama algoritmalarını içeren bir sortlama algoritması modülü kullanmaktadır. Bu algoritma modülü, kullanıcının seçtiği sortlama algoritmasını kullanarak bir diziyi sıralamak için gerekli adımları gerçekleştirir. Her sortlama algoritması, proje içerisinde ayrı bir fonksiyon olarak implemente edilmiştir.

**3.Veri Yapılarının Kullanımı:** Proje, sortlama algoritmalarının verimli bir şekilde çalışabilmesi için çeşitli veri yapılarını kullanmaktadır. Örneğin, diziyi depolamak ve sıralama adımlarını izlemek için bir dizi veri yapısı kullanılır. Ayrıca, sıralama işlemi sırasında elemanların yerlerini değiştirmek için geçici bir değişken kullanılır.

**4.Görselleştirme ve Animasyon:** Proje, sortlama işlemini görselleştirmek ve kullanıcılara adımları animasyonlu bir şekilde göstermek için grafikler ve animasyonlar kullanmaktadır. Her adımda, sıralanmamış dizi, sıralanmış dizi ve aktif adımın vurgulandığı grafikler gösterilir. Böylece kullanıcılar, sortlama işleminin nasıl ilerlediğini ve algoritmanın nasıl çalıştığını daha iyi anlayabilirler.

**5.Test ve Doğrulama:** Projenin doğru çalışması ve beklenen sonuçları üretebilmesi için test ve doğrulama süreci gerçekleştirilmiştir. Bu süreçte, farklı dizi boyutları ve sortlama algoritmaları kullanılarak proje test edilmiş ve sonuçlar incelenmiştir. Ayrıca, projenin performansı, sıralama süresi ve hafıza kullanımı gibi faktörler açısından değerlendirilmiştir.

**6.Geliştirme ve İyileştirme:** Proje geliştirme aşamasında, kullanıcı geribildirimleri dikkate alınarak ve performans iyileştirmeleri yapılarak sürekli olarak güncellenmiştir. Bu aşamada, projenin daha kullanıcı dostu, hızlı ve istikrarlı bir şekilde çalışabilmesi için geliştirmeler yapılmış ve hatalar giderilmiştir.

Sonuç olarak, projenin yöntemi, kullanıcı arayüzü tasarımı, sortlama algoritmalarının implementasyonu, veri yapılarının kullanımı, görselleştirme ve animasyon, test ve doğrulama süreci ve geliştirme aşamalarını içermektedir. Bu yöntemler ve süreçler, proje hedeflerine ulaşmak, sortlama algoritmalarını anlamak ve kullanıcı deneyimini iyileştirmek için etkili bir şekilde kullanılmıştır.

TARTIŞMA VE ÖNERİLER

**Projenin Başarısı ve Başarı Kriterleri:** Projenin başarı kriterleri, hedeflenen sonuçların ne kadar başarılı bir şekilde elde edildiğini belirlemektedir. Bu projede, sortlama algoritmalarının performansı, hafıza kullanımı ve kullanıcı etkileşimi gibi kriterler üzerinde odaklanılmıştır. Elde edilen sonuçlar, projenin başarılı olduğunu göstermektedir. Ancak, performans metriklerinin daha kapsamlı bir şekilde değerlendirilmesi ve kullanıcı geri bildirimlerinin daha detaylı analiz edilmesi projenin başarısını daha iyi değerlendirmeye yardımcı olabilir.

**Karşılaşılan Zorluklar ve Çözüm Önerileri:** Projede bazı zorluklarla karşılaşılmış olabilir. Örneğin, algoritmaların uygulanması, görselleştirme ve animasyon özelliklerinin entegrasyonu gibi teknik zorluklar yaşanmış olabilir. Bu tür zorluklarla karşılaşan projelerde, daha iyi bir planlama, daha etkili bir kaynak yönetimi ve gerektiğinde uzmanlardan yardım almak çözüm önerileri olabilir.

**Gelecekteki Çalışmalar ve Geliştirmeler için Öneriler:** Bu projenin gelecekteki çalışmalara ve geliştirmelere ilham vermesi önemlidir. Öneriler şunları içerebilir:

**Yeni Sortlama Algoritmaları**: Proje, bazı temel sortlama algoritmalarını içermektedir. Gelecekte, daha gelişmiş ve optimize edilmiş sortlama algoritmalarının projeye eklenmesi, daha iyi performans ve daha hızlı sıralama süreleri elde edilmesine katkı sağlayabilir.

**Veri Yapıları:** Proje, sortlama algoritmalarının uygulanması için temel veri yapılarını kullanmaktadır. Gelecekte, farklı veri yapılarının projeye dahil edilmesi ve performanslarının karşılaştırılması, daha geniş bir veri analizi yapılmasına imkan tanıyabilir.

**Gelişmiş Kullanıcı Deneyimi:** Projenin kullanıcı deneyimini daha da geliştirmek için interaktif özelliklerin eklenmesi düşünülebilir. Kullanıcıların sortlama adımlarını daha etkileşimli bir şekilde kontrol etmeleri ve özelleştirebilmeleri projenin kullanıcılar üzerinde daha etkili bir etki yaratmasını sağlayabilir.

**Gerçek Zamanlı Uygulama:** Proje, belirli dizi boyutları üzerinde sortlama işlemlerini gerçekleştirmektedir. Gelecekte, proje gerçek zamanlı verileri işleyebilen ve sürekli güncelleyen bir yapıya dönüştürülebilir. Bu şekilde, gerçek dünya senaryolarında sortlama algoritmalarının performansını değerlendirmek için kullanılabilir.

**Performans Optimizasyonu:** Projenin performansı daha da iyileştirilebilir. Algoritmaların daha verimli ve optimize edilmiş versiyonları kullanılarak hızlı sortlama süreleri elde edilebilir. Ayrıca, bellek kullanımının optimize edilmesi ve algoritmaların çoklu çekirdekli işlemcilerle uyumlu hale getirilmesi performansı artırabilir.

Bu öneriler, projenin gelecekte daha ileriye taşınması ve sortlama algoritmaları alanında daha fazla keşif yapılması için bir temel oluşturabilir.

SONUÇ

Bu proje, sortlama algoritmalarının görselleştirilmesi ve animasyonlarıyla interaktif bir şekilde sunulması amacıyla tasarlanmıştır. Projede, çeşitli sortlama algoritmaları (kabarcık, seçim, ekleme) kullanılarak veri setlerinin sıralanması simüle edilmiştir. Bu sayede, kullanıcılar sortlama adımlarını daha iyi anlayabilir ve algoritmalar arasındaki farkları görsel olarak deneyimleyebilirler.

Elde edilen sonuçlar, projenin başarılı olduğunu göstermektedir. Sortlama algoritmalarının doğru bir şekilde uygulandığı ve veri setlerinin başarılı bir şekilde sıralandığı görülmüştür. Ayrıca, projenin kullanıcı dostu arayüzü sayesinde kullanıcılar sortlama adımlarını izleyebilir, görsel değişimleri takip edebilir ve sortlama sürecini daha iyi anlayabilirler.

Bu proje, sortlama algoritmaları konusunda öğrenme ve anlama sürecine katkıda bulunmayı hedeflemektedir. Kullanıcılar, sortlama algoritmalarının çalışma prensiplerini ve performanslarını görsel olarak deneyimleyerek daha iyi kavrayabilirler. Ayrıca, projenin interaktif özellikleri sayesinde kullanıcıların sortlama adımlarını kontrol etmeleri ve deneyimlerini özelleştirmeleri sağlanmıştır.

Bu proje, sortlama algoritmaları ve veri yapıları konusunda öğrencilere, araştırmacılara ve yazılım geliştiricilere faydalı bir kaynak sağlamaktadır. Görsel ve interaktif bir şekilde sunulan sortlama algoritmaları, bu konuları daha anlaşılır ve ilgi çekici hale getirmektedir. Ayrıca, projenin kaynak kodu açık bir şekilde paylaşılmıştır, böylece diğer geliştiriciler projeyi inceleyebilir, geliştirebilir ve özelleştirebilirler.

Sonuç olarak, bu proje sortlama algoritmalarının görselleştirilmesi ve animasyonlarıyla interaktif bir şekilde sunulması konusunda başarılı bir örnek olmuştur. Kullanıcı dostu arayüzü, çeşitli sortlama algoritmalarının görsel deneyimini sunması ve öğrenme sürecine katkıda bulunması projenin önemini vurgulamaktadır. Gelecekte, projenin daha da geliştirilerek yeni sortlama algoritmaları ve özelliklerin eklenmesi potansiyeli bulunmaktadır.

KAYNAKLAR VE REFERANSLAR

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. MIT Press.

Sedgewick, R., & Wayne, K. (2011). Algorithms. Pearson Education.

Lafore, R. (2002). Data Structures and Algorithms in Java. Sams Publishing.

GeeksforGeeks. (2023). Sorting Algorithms. Erişim adresi: <https://www.geeksforgeeks.org/sorting-algorithms/>

Python.org. (2023). Python Documentation. Erişim adresi: <https://docs.python.org/>

W3Schools. (2023). Python Tutorial. Erişim adresi: <https://www.w3schools.com/python/>

Stack Overflow. (2023). Python Questions and Answers. Erişim adresi: <https://stackoverflow.com/>

Data Structures and Algorithms in Python - Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser.

Introduction to the Design and Analysis of Algorithms - Anany Levitin.

Python Data Structures and Algorithms - Benjamin Baka.

Real Python. (2023). Python Sorting Algorithms: A Comprehensive Guide. Erişim adresi: <https://realpython.com/sorting-algorithms-python/>

DataCamp. (2023). Sorting Algorithms in Python. Erişim adresi: <https://www.datacamp.com/community/tutorials/sorting-algorithms-python>

Tutorialspoint. (2023). Python - Data Structures. Erişim adresi: <https://www.tutorialspoint.com/python_data_structure/index.htm>

Tutsplus. (2023). Data Structures and Algorithms in Python. Erişim adresi: <https://code.tutsplus.com/tutorials/data-structures-and-algorithms-in-python--cms-25617>

Towards Data Science. (2023). Data Structures and Algorithms in Python: A Practical Guide. Erişim adresi: <https://towardsdatascience.com/data-structures-and-algorithms-in-python-a-practical-guide-28c7e79f924b>

Formun Üstü