



**BURSA TEKNİK ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BLM0314 Algoritma Analizi ve Tasarımı**  
**Proje Ödevi**

**Arama ve Sıralama Algoritmaları için Arayüz Geliştirme**

Öğrenci Adı Soyadı: Nil DUMAN

Öğrenci Numarası: 18360859026

OCAK 2021

## İçindekiler

1. GİRİŞ.....	2
1.1 Algoritma Analizi ve Tasarımı .....	3
2. SIRALAMA (SORT) ALGORİTMALARI .....	3
2.1 Sokuşturma Sıralama (Insertion Sort) Algoritması .....	3
2.2 Birleşen Sıralama (Merge Sort) Algoritması .....	4
2.3 Yığın Sıralama (Heap Sort) Algoritması .....	5
2.4 Hızlı Sıralama (Quick Sort) Algoritması .....	5
2.5 Sayarak Sıralama (Counting Sort) Algoritması .....	6
2.6 Kova Sıralama (Bucket Sort) Algoritması .....	6
2.7 Seçmeli Sıralama (Selection Sort) Algoritması .....	7
3. ARAMA (SEARCH) ALGORİTMALARI .....	8
3.1 Doğrusal Arama (Linear Search) Algoritması .....	8
3.2 İkili Arama (Binary Search) Algoritması .....	8
4. ARAYÜZ OLUŞTURMA .....	9
4.1 PYQT .....	9
5. KODLARIN ARAYÜZE AKTARIMI .....	9
6. KAYNAKÇA.....	13

KISACA- Bu rapor incelemesinde, Arama ve Sıralama Algoritmalarının PYQT5 ile tasarlanan bir arayüz ile kullanıcıya aktarılması amaçlanmıştır.

Anahtar Kelimeler: Arama Algoritması, : Sıralama Algoritması, arayüz, PYQT5

### 1. GİRİŞ

Bir sorunu çözmek veya belirlenmiş bir amaca ulaşmak için tasarlanan yola, takip edilen işlem basamaklarına ve bu basamakları bilgisayar ortamında herhangi bir dilde kodlamaya “**algoritma**” denir. Başka bir deyişle herhangi bir işi yapmak veya bir problemi çözmek için uygulanan kurallar dizisi bütününe denir. Algoritmalar açıkça belirtilmiş bir başlangıcı ve sonu olan işlemler kümesidir. İlk algoritma kavramı, El-Harezmi’nin ‘Hisab-el Cebir ve El Mukabala’ kitabında ortaya çıkmıştır. Alimin ismini telaffuz edemeyen Avrupalılar “**algorizm**” sözcüğünü “Arap sayıları kullanarak aritmetik

problemler çözme kuralları” anlamında kullanılır. Bu sözcük daha sonra algoritmaya dönüşür ve genel kapsamda kullanılır.

## 1.1 Algoritma Analizi ve Tasarımı

**Algoritma analizi** veya diğer adıyla **algoritma çözümlemesi**, bilgisayar biliminde bir algoritmayı çalıştırabilmek için gereken kaynakların (zaman, yer gibi) miktarının tespiti. Algoritmaların çoğunluğu, rastgele seçilmiş uzunluktaki girdiler ile çalışmak için tasarlanmıştır. Genellikle, bir algoritmanın verimlilik veya çalışma zamanı, adımların sayısı (zaman karmaşıklığı) veya depolama yerlerinin (alan karmaşıklığı) girdi uzunluğuyla ilişkili olan işlev olarak ifade edilir.

Algoritma çözümlemesi, verilmiş olan hesaplamalı bir bilinmezi çözen herhangi bir algoritma aracılığıyla gereksinim duyulan kaynaklar için kuramsal tahminleri sağlayan hesaplamalı karmaşıklık kuramının önemli bir parçasıdır. O, verimli algoritmaları bulabilmek veya kıyaslayabilmek için bir anlayış geliştirmemizi sağlar.

Algoritma analizi yapılırken göz önünde bulundurulması gereken noktalar vardır:

- 🚦 Algoritmanın Doğruluğu (Correctness)
- 🚦 Zaman Verimliliği (Time Efficiency)
- 🚦 Bellek Alanı Verimliliği (Space Efficiency)
- 🚦 Uygunluk, en iyilik (Optimality)

## 2. SIRALAMA (SORT) ALGORİTMALARI

Sıralama algoritmalarının hesaplama verimliliği açısından; veri yapıları olarak dizi (array), yığın (stack), kuyruk (queue) veya ağaç (tree) kullanılabilir. Literatürde çok farklı sıralama algoritmaları mevcuttur.

Bu rapor incelemesinde 7 tür sıralama algoritması ele alınmıştır.

### 2.1 Sokuşturma Sıralama (Insertion Sort) Algoritması

Türkçe ismi Sokma Sıralaması olan Insertion Sort, diğer sıralama algoritmalarına nazaran kodlaması oldukça kolay bir algoritmadır. Kodlanması kolay olduğu gibi uygulanması da kolaydır. Genelde kart

oyunlarında insanlar elindeki kartları sıralamak için farkında olmadan bu algoritmayı uygulamaktadır. Ancak karmaşıklığı  $N^2$  olduğu için verimli bir algoritma olarak kabul edilmez.

Algoritmada döngüsü her bir tur döndüğünde sıradaki elemanı sondan başa doğru karşılaştırarak yerine yerleştirme esaslı çalışmaktadır.

### **Bu algoritma basit bir örnek ile açıklanabilir:**

Masada bir deste sırasız, sırtları yukarıya doğru oyun kâğıdı bulunduğu varsayılırsa

- 1) Desteden en üstteki kartı alınarak masaya yüzü görünür biçimde koyulur. Tek kart olduğu için sıralı bir kümedir.
- 2) Sırasız destenin üstünden bir kart daha çekilir. Masadaki ilk çekilen kart ile karşılaştırılır. Gerekirse yerleri değiştirerek, çekilen iki kart, küçükten büyüğe doğru sıralanır.
- 3) Sırasız destenin üstünden bir kart daha çekilir. Masada sıralı dizilen iki kart ile karşılaştırılır. Gerekirse yerleri değiştirerek, çekilen üç kart, küçükten büyüğe doğru sıralanır.
- 4) Bu işlem sırasız deste bitene kadar devam edilir. Sonunda, oyun kağıtları sıralanmış olur.

Herhangi bir adımda, kâğıt destesi sıralı alt deste ve sırasız alt deste olmak üzere iki alt kümeye ayrılmış durumdadır. Her adımda, sırasız desteden bir kart çekilip, sıralı destedeki doğru yerine yerleştirilir ve bu iş sırasız deste bitene kadar devam eder.

## **2.2 Birleşen Sıralama (Merge Sort) Algoritması**

Birleşen Sıralama; diziyi ardışık olarak en küçük alt dizilerine kadar yarılayan, sonra da onları sıraya koyarak birleştiren özyinelemeli bir algoritmadır. Yarılama işlemi en büyük alt dizi en çok iki öğeli olana kadar sürer. Sonra merge (birleşim) işlemiyle alt diziler ikiye ikiye bölünüş sırasıyla sıralı olarak bir üst dizide birleştirilir. Süreç sonunda en üstte sıralı diziyi ulaşılr. Bu algoritmanın karmaşıklığı  $O(n \log n)$  dir.

Algoritmanın çalışması kavramsal olarak şu şekildedir:

- 1) Sıralı olmayan dizi, ortadan eşit olarak iki alt diziyeye ayırır.
- 2) Bu ayırma işlemi, alt diziler en çok iki elemanlı olana kadar devam eder.
- 3) Alt dizileri kendi içinde sıralar.
- 4) Sıralı iki alt diziyi tek bir sıralı dizi olacak şekilde birleştirir.

### 2.3 Yığın Sıralama (Heap Sort) Algoritması

Bu sıralama algoritması iyi denilebilecek bir hızda çalışan sıralama algoritmalarından birisidir. En önemli özelliklerinden biri de en büyük ve en küçük sayıya çok hızlı bir şekilde ulaşmaya olanak sağlamaktadır.

Yığın sıralamasında: bir ata root bulunur. Root'un ise sol ve sağ olmak üzere çocukları bulunur. Sıralama işlemi yaparken bu alanlar üzerlerinde değişiklikler yapılır. Detaylı incelendiğinde yığın sıralamasının bir ağaç yapısına benzediği farkedilir. İkili ağaç yapısı gibi görünür, fakat o şekilde çalışmaz., Yığın sıralaması hem dizi mantığı ile hem de düğüm (node) mantığı ile kod olarak yazıp çalıştırılabilir.

**Yığın sıralamasının çalışma süresi (karmaşıklığı) şu şekildedir:**

En iyi Durum:  $\Omega(n \log(n))$

Ortalama Durum:  $O(n \log(n))$

En Kötü Durum:  $O(n \log(n))$

Maksimumum & Minimum Sayıya Ulaşma:  **$O(1)$  Sabit**

Dizideki En Büyük Değer Max-Heap'a Göre Dizinin İlk Elemanıdır = **Array[0]**

Dizideki En Küçük Değer İse Max-Heap'a Göre Dizinin Son Elemanıdır = **Array[Array.Length - 1]**

### 2.4 Hızlı Sıralama (Quick Sort) Algoritması

Hızlı Sıralama algoritması C.A.R.Hoare tarafından bulunan etkin bir sıralama yöntemidir. Siyaset biliminde çok kullanılan “böl ve yönet” stratejisine dayanan basit ve hızlı bir sıralama yöntemi kullanır. Zaman karmaşası, ortalama  $O(n \log n)$  ve en kötü durumda ise  $O(n^2)$  dir. Bu algoritma, başlarken dizinin terimleri arasından bir terimi mihenk (pivot) olarak seçer. Sonra verilen diziyi üç alt diziyeye ayırır. Mihenk'ten küçük olan terimlerin hepsini (soldaki) birinci altdiziyeye taşır. İkinci alt dizi biricik ögesi mihenk olan tek terimli {mihenk} alt dizisidir. Mihenk'ten büyük olan terimlerin hepsini (sağdaki) ikinci alt diziyeye taşır. Sonra sol ve sağ alt dizilere aynı ayrıştırma yöntemini, alt diziler tek terimli birer diziyeye indirgenene kadar uygular ve sıralama işlemi biter. Algoritma özyinelemelidir. (recursive).

**NOT:** Merge Sort algoritması ile Heap Sort algoritması aynı zaman aralığına sahip olmalarına rağmen Heap Sort algoritması bellekte Merge Sort algoritmasına göre daha az yer tutar ve bu algoritmalar

gerçekleştirildiğinde Heap Sort algoritması daha hızlı çalışır. Quick Sort algoritması da bellek tabanlı dizilerde Merge Sort'a göre daha hızlı çalışmaktadır. Ancak bağlı liste sıralamasında seçilebilecek en performanslı algoritma Merge Sort algoritmasıdır. Çünkü bağlı listelerin yapısı gereği Merge Sort bellekte fazladan sadece 1 birim yer tutar ve bağlı listelerin yavaş ve rastgele erişim performansı nedeniyle Quick Sort gibi diğer algoritmaların çalışma performansı düşer, Heap Sort gibi algoritmalar için ise imkansızdır.

## 2.5 Sayarak Sıralama (Counting Sort) Algoritması

Sayarak sıralama bilgisayar bilimlerinde kullanılan ve kova sıralaması gibi sıralanacak dizinin içindeki değerlerin aralığının bilinmesi durumunda kullanılabilen bir sıralama algoritmasıdır.

Bu algoritmada en iyi durum diye bir şey söz konusu değildir. Ortalama ve n kötü durum vardır ve karmaşıklığı  $O(n+r)$  dir. Potansiyel değer aralığı büyük ise sıralama sayımı çok fazla alan gerektirir.

### Çalışma prensibi şu şekilde özetlenebilir:

Sayarak sıralama algoritması dizideki değerlerin aralık bilgilerini yeni bir dizi oluşturmak için kullanır. Oluşturulan yeni dizinin her bir satırı ana dizide o satır numarasının değerine sahip öğelerin sayısını gösterir. Yeni dizideki öğe değeri sayıları daha sonra ana dizideki tüm değerlerin doğru konuma konulması için kullanılır.

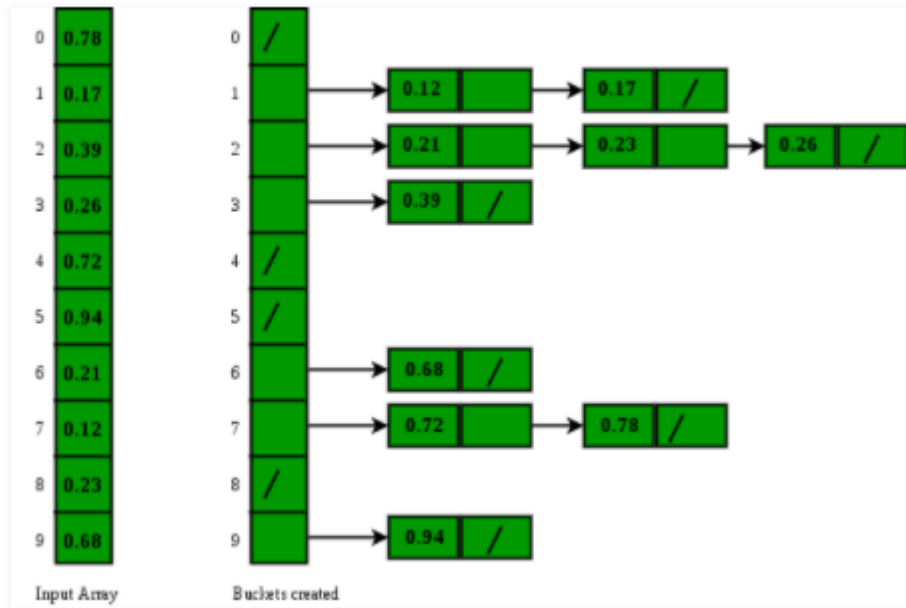
## 2.6 Kova Sıralama (Bucket Sort) Algoritması

Kova Sıralaması (ya da sepet sıralaması), sıralanacak bir diziyi parçalara ayırarak sınırlı sayıdaki kovalara (ya da sepetlere) atan bir sıralama algoritmasıdır. Ayrışma işleminin ardından her kova kendi içinde ya farklı bir algoritma kullanılarak ya da kova sıralamasını özyinelemeli olarak çağırarak sıralanır. Kova sıralaması aşağıdaki biçimde çalışır:

- 1) Başlangıçta boş olan bir "kovalar" dizisi oluştur.
- 2) Asıl dizinin üzerinden geçerek her öğe, ilgili aralığa denk gelen kovaya atılır.
- 3) Boş olmayan bütün kovalar sıralanır.
- 4) Boş olmayan kovalardaki bütün öğeler, yeniden diziyeye alınır.

Kova sıralaması doğrusal zamanda çalışır. Girişin düzgün dağılımlı olduğu kabul edilir. Rastgele olarak  $[0,1)$  aralığında oluşturulmuş giriş bilgileri olduğu kabul edilir. Temel olarak  $[0, 1)$  aralığını n eşit alt aralığa böler ve girişi bu aralıklara dağıtır. Aralıklardaki değerleri insert sort ile sıralar. Aralıkları birbiri ardına ekleyerek sıralanmış diziyi elde eder.

Bucket sort genelde dinamik veri yapıları kullanıldığı zaman kullanılır. Yaptığı iş her bir digit değerlendirmesi için liste oluşturmaktır. Örnek olarak, 3 ile bitenler için L3 linked list (bağlı liste)'i, 4 ile bitenler için L4 linked list'i ve her listin kendi arasında sıralanması verilebilir.



## 2.7 Seçmeli Sıralama (Selection Sort) Algoritması

Seçmeli sıralama algoritması ile sıralanmamış bir listede her iterasyondaki en küçük eleman bulunur. İterasyon sonunda belirlenen en küçük eleman listenin başında bulunan eleman ile yer değiştirir. Bir sonraki iterasyona dahil edilmez. İkinci iterasyonda yine dizi elemanlarından en küçük olanı bulunur ve bu sefer listedeki 2. eleman ile yer değiştirir ve bu şekilde liste sıralanana kadar devam eder.

Algoritmanın çalışması kavramsal olarak şu şekildedir:

[4, 3, 5, 2] listesini ele alırsak,

Seçmeli sıralama kullanılarak bu listeyi sıralanacak olursa eğer, minimum değer olarak ilk elemanı minimum değer olarak belirlenir. Daha sonra ilk eleman olan 4 rakamı diğer dizi elemanlarının her biri ile karşılaştırılmaya başlanır. 4, 3'ten büyük olduğu için artık yeni minimum değeri 3 olarak kabul edilir ve diğer elemanlar ile karşılaştırma işlemine yeni minimum değer ile devam edilir. İlk iterasyonun sonunda minimum değer olarak 2 belirlenir ve belirlenen bu 2 değeri, dizinin en başında bulunan değer ile yer değiştirir.

1. İterasyon: [4, 3, 5, 2] → [4, 3, 5, 2] → [4, 3, 5, 2] → [4, 3, 5, 2] → [2, 3, 5, 4]

2. İterasyon: [2, 3, 5, 4] → [2, 3, 5, 4] → [2, 3, 5, 4]

3. Iterasyon: [2, 3, **5**, **4**] → [2, 3, 4, 5]

Bu örnekte, 2 iterasyon gerçekleştirdikten sonra dizi elemanları sıralanmış olur.

### 3. ARAMA (SEARCH) ALGORİTMALARI

Arama algoritmaları, bilgisayar biliminde seçili özelliklere göre istenilen bilgileri bulan algoritmalarlardır. Listeler, metinler ve şekiller üzerinde çalışırlar. Arama algoritmaları adından da anlaşılacağı gibi, bir veri yapısı üzerinde belirli bir verinin aranması için kullanılan algoritmalar bütünüdür.

Bu rapor incelemesinde 2 tür arama algoritması ele alınmıştır.

#### 3.1 Doğrusal Arama (Linear Search) Algoritması

Sıralı Arama olarak da bilinen belirli bir değer için bir veri listesini aramaya uygun ve en temeli olan bir arama algoritmasıdır. Eşleşme bulana kadar listenin bütün elemanlarını bir kere kontrol ederek çalışır. Algoritmanın mantığı, bir kümeye, diziye, link liste veya başka bir veri yapısında bulunan elemanları, baştan başlayarak sırası ile kontrol etmektir.

Doğrusal arama  $O(n)$  karmaşıklığı ile çalışır. En iyi durum ilk karşılaştırmada aranan değer bulunmasıdır. En kötü durum ise tüm listenin tüm elemanlarını karşılaştırma işlemine sokmaktır. Örneğin 5 elemanlı bir dizi içerisindeki bir sayının arama sırasında en kötü son elemanda veya dizide hiç bulunmaması durumudur. Bu yüzden dizide 5 kez kontrol yapılacaktır.

#### 3.2 İkili Arama (Binary Search) Algoritması

İkili Arama Algoritması, sıralı (sorted) bir veri yapısı için kullanılır. Yani algoritmaya aranan veri ve sıralı bir veri yapısı verilir. Algoritma da size önceki eğer bulunursa aranan verinin indeksini döndürür. Bunun için önce eldeki verinin, bir sıralama (sorting) algoritması ile sıralanması gerekir.

Örnek olarak; 2,3,4,5,6,7,8,9,22,33,45 elemanlarından oluşan sıralı bir dizi içerisinde, 4 elemanının ikili arama ile bulunması için sırasıyla şu adımlar izlenecektir.

Dizinin ortadaki elemanı 7 olarak seçilir ve aranan eleman olan 4 ile karşılaştırılır. Aranan eleman (4) ortadaki elemana (7) eşit olmadığı için, dizinin ortasındaki eleman olan 7'den küçük olan kısmına bakılır. Yeni arama dizisi: 2,3,4,5,6 olur.

Yeni arama dizimizin ortadaki elemanı 4 olacaktır ve arama işlemi tamamlanır.

İkili arama algoritmasında, sıralı bir dizide en fazla  $\log_2 N$  adet karşılaştırma yapılarak sonuca ulaşılır.



## 4. ARAYÜZ OLUŞTURMA

Hesaplama bir arayüz, bir bilgisayar sisteminin iki veya daha fazla ayrı bileşeninin bilgi alışverişi içinde bulunduğu paylaşılan bir sınırdır. Değişim, yazılım, bilgisayar donanımı, çevresel cihazlar, insanlar ve bunların kombinasyonları arasında olabilir. Bu rapor incelemesinde istenilen arayüz PYQT kullanılarak tasarlanmıştır.

### 4.1 PYQT

**PyQt**, çapraz platform uygulama geliştirmeye yarayan ve C++ ile yazılmış olan Qt kütüphanesinin Python bağlamasıdır. **Bir programlama dili değildir.** Python ile grafiksel kullanıcı arayüzlü programlar oluşturmamızı sağlar.

**PyQt**, *grafiksel kullanıcı arayüzleri, XML işlemeyi, ağ iletişimini, SQL veritabanlarını* ve Qt' de bulunan diğer teknolojileri de kapsayan 620'den fazla sınıfı içerisinde barındırır. PyQt, İngiliz şirket Riverbank Computing Limited tarafından geliştirilmiştir.

PyQt'nin iki sürümü vardır. Bunlar; Qt 4.x ve Qt 5.x için oluşturulan PyQt4 ve yalnızca 5.x için kullanılacak olan **PyQt5**'dir. Bu rapor incelemesinde **PyQt5** sürümünü kullanılmıştır.

## 5. KODLARIN ARAYÜZE AKTARIMI

Arayüzde bir giriş iki de arama ve sıralama algoritmaları olmak üzere 3 sayfa oluşturulmuş daha sonra bu 3 sayfa birleştirilmiştir.

Algoritmanın ekran görüntüleri aşağıda verilmiştir.

Sıralama

Kaç tane sayı üreteceksiniz

Min Sayı DeğeriMax Sayı Değeri

Rastgele Sayı Üret

Hangi algoritma ile sıralama yapmak istediğinizi seçiniz.

Üretilen Sayılar

Geçen Süre

Sıralanmış Sayılar

Yapılan İşlem Sayısı

Arama

Kullanılacak Algoritmayı Seçiniz.

Aramak istediğiniz sayıyı giriniz.

Üretilen Sayılar

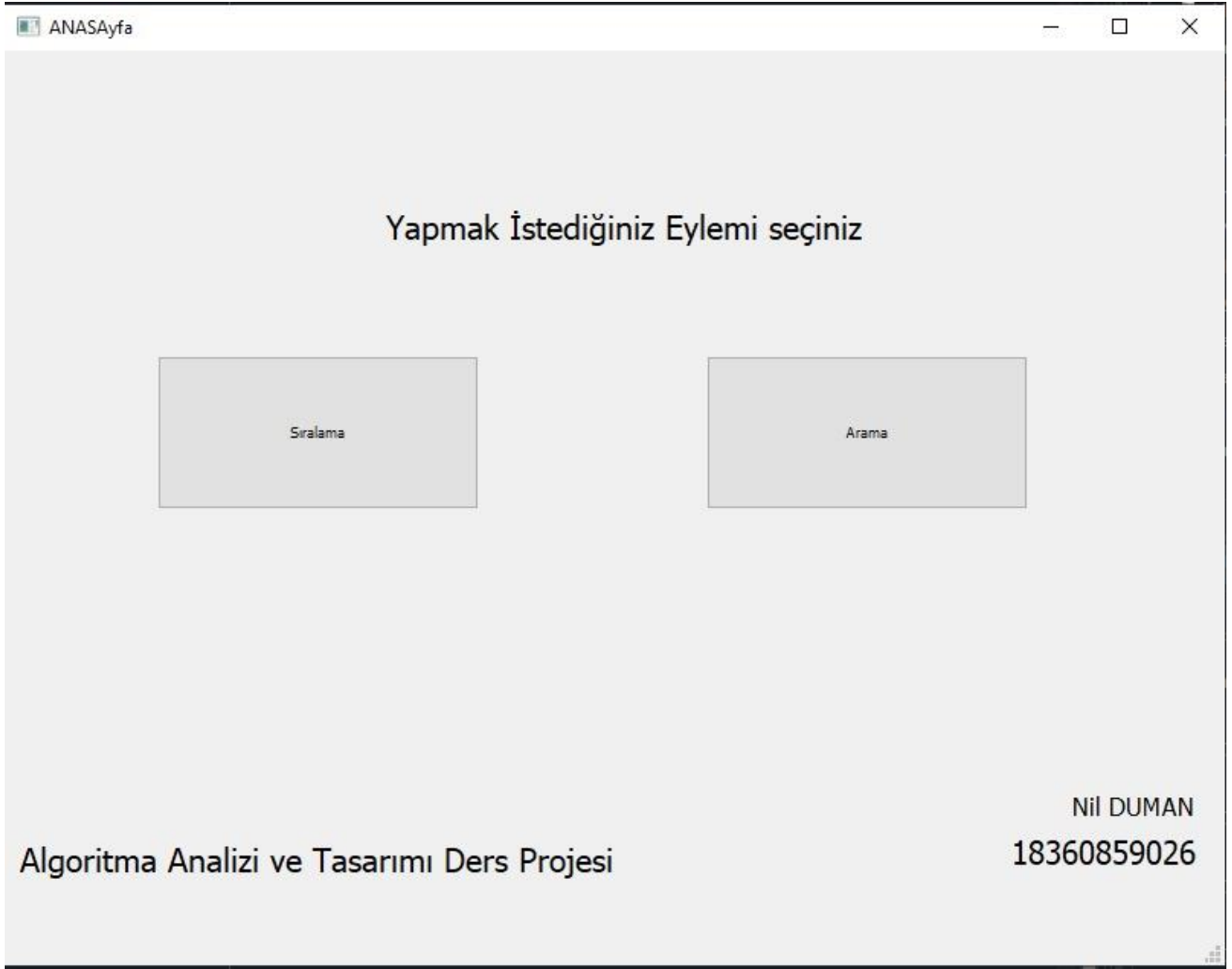
Kaç tane sayı üreteceksiniz

Min Sayı Değeri Max Sayı Değeri

Rastgele Sayı Üret

Geçen Süre

Yapılan İşlem Sayısı



## 6. KAYNAKÇA

- Medium. 2017. *Selection Sort Algoritması Nedir?* . “<https://medium.com/kodcular/selection-sort-algoritmas%C4%B1-nedir-484c40c9473c> 08.01.2021”
- Cagataykiziltan.net. 2019. 8. *Taban Sıralama (Radix Sort)* | *Mustafa Çağatay KIZILTAN*. “<http://cagataykiziltan.net/algoritmalar/1-siralama-algoritmaları/9-taban-siralamaradix-sort/> 08.01.2021”
- HackerEarth. 2020. *Selection Sort Tutorials & Notes | Algorithms | Hackerearth*. “<https://www.hackerearth.com/practice/algorithms/sorting/selection-sort/tutorial/> 08.01.2021”
- Maker | Türkiye | Robotistan.com. 2021. *Algoritma Nedir? Niçin Kullanılır? Algoritma Örnekleri*. “<https://maker.robotistan.com/algoritma/> 09.01.2021”
- Hakan Torun. 2021. *İkili Arama Algoritması (Binary Search Algorithm)- C Örnek Kod*. “<https://hakan.io/ikili-arama-algoritmasi-binary-search-algorithm-c-ornek-kod/> 09.01.2021”
- Jain, S., 2016. *Linear Search Vs Binary Search | Tutorialhorizon*. Algorithms. “<https://algorithms.tutorialhorizon.com/linear-search-vs-binary-search/> 09.01.2021”
- Medium. 2021. *Sıralama Algoritmalarının Karşılaştırılması*. [https://medium.com/@halisak\\_/s%C4%B1ralama-algoritmalar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9F%C4%B1r%C4%B1lmas%C4%B1-dae2a88de6b](https://medium.com/@halisak_/s%C4%B1ralama-algoritmalar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9F%C4%B1r%C4%B1lmas%C4%B1-dae2a88de6b) 10.01.2021”
- Alikorayyavic.gitbooks.io. 2021. *Sıralama Algoritmaları · Algoritma Ve Programlamaya Giriş*. <https://alikorayyavic.gitbooks.io/algoritma-ve-programlamaya-giris/content/algoritma-cesitleri.html> 10.01.2021
- DUMAN, N., 2021. *Yazılıma Girişte Algoritmanın Önemi- Teknikafa*. “<https://teknikafa.com/yazilima-giriste-algoritmanin-onemi/> 12.01.2021”
- Tutorialspoint.com. 2021. *Design And Analysis Of Algorithms Tutorial- Tutorialspoint*. “[https://www.tutorialspoint.com/design\\_and\\_analysis\\_of\\_algorithms/index.htm](https://www.tutorialspoint.com/design_and_analysis_of_algorithms/index.htm) 11.01.2021”
- Aytugonan.cbu.edu.tr. 2021. “[http://aytugonan.cbu.edu.tr/YZM3207/Lecturtes/YZM3107\\_ders01.pdf](http://aytugonan.cbu.edu.tr/YZM3207/Lecturtes/YZM3107_ders01.pdf) 11.01.2021”
- ÇELİK, D., 2019. *Sıralama Algoritmaları*. Ceng.harran.edu.tr. <http://ceng.harran.edu.tr/demo-day/2019/ParallelSortingAlgorithms/dokumanlar/RAPOR1.pdf> 07.01.2021
- Karahasan, B., 2021. *Pyqt Nedir? Qt Designer Nedir? Python Arayüz Tasarımı*. Birhankarahasan.com. “<https://birhankarahasan.com/pyqt-nedir-qt-designer-nedir-python-arayuz-olusturma> 15.01.2021”