

**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:

Öğrenci Numarası:

OCAK 2021

**İçindekiler Tablosu**

[1.GİRİŞ 3](#_Toc61530440)

[2. ARAMA ALGORİTMALARI 3](#_Toc61530441)

[**2.1 Programın Detaylı Çalışma Mantığı** 3](#_Toc61530442)

[**2.2 Algoritmaların Detaylı İncelenmesi** 3](#_Toc61530443)

[3. SIRALAMA ALGORİTMALARI 4](#_Toc61530444)

[**3.1 Programın Detaylı Çalışma Mantığı** 4](#_Toc61530445)

[**3.2 Algoritmaların Detaylı İncelenmesi** 4](#_Toc61530446)

[4. PROGRAMA AIT FOTOĞRAFLAR 5](#_Toc61530444)

[5. SONUÇ 12](#_Toc61530447)

# **1.GİRİŞ**

Sıralama ve arama algoritmaları birçok yerde kullanılıyor. Örneğin telefonun rehberindeki bir kişinin telefon numarasının bulunması bir arama işlemidir. Aynı zamanda alfabeye göre sıralaması da bir sıralama işlemidir. Biz bu programda sayılar ile ilgileneceğiz. Bu program sayıların sıralanması ve aranması işlemlerini içerir.

# **2. ARAMA ALGORİTMALARI**

## **2.1 Programın Detaylı Çalışma Mantığı**

* **Arama algoritmanın seçilmesi:** Bu bölümde Combobox ile 2 adet arama algoritmanın seçimi gerçekleştiriliyor. Ardından rastgele sayı üreteceğimiz bölüm açılıyor. Algoritma seçmeden bu bölüme geçiş yapamazsınız.
* **Rastgele Sayı Üretimi:** Bu bölümde 3 adet textbox ile ilk olarak kaç tane üretileceği yazılır. Eğer üretmek istenilen sayı 0 olursa hiçbir şey olmuyor. Tekrar denenmesini istiyor. Ayrıca minimum maksimum aralıktaki değerler eğer yanlış girilirse hata veriyor. Doğru girmeden arama alanına geçilemiyor. Rastgele sayı üret butonuna tıklandığı zaman ise her şey doğru girilmişse arama alanı açılıyor ardından listBox1’e girilen değerlerde bir dizi oluşturuluyor.
* **Aramanın yapılması:** Aramanın yapılması için bir aranan değere ihtiyacımız var. Bunu da textbox ile alıyor. Textbox’a herhangi bir yazı karakteri girilemiyor. Yalnızca sayı karakterleri girilebiliyor. Ara butonuna tıklandığında listBox1’den seçilen algoritmaya göre arama gerçekleşiyor. Eğer bulunduysa algoritmanın bilgileri veriliyor. Bulunmadıysa “Aradığınız sayı bulunamadı.” diye bir hata çıkarıyor. Program tekrar arama fırsatı sağlıyor. Aranan sayı bulunduğunda ise listBox’da bulunduğu index seçili hale geliyor.
* **Not: Hiçbir textbox’a harf girilemez. Min, max sayı değerleri -9999 ile +9999 arası sınırlıdır.**

## **2.2 Algoritmaların Detaylı İncelenmesi**

* **Linear Search(Lineer Arama):** Verilen veri seti üzerinden her bir elemana tek tek bakıp istenen değer ile karşılaştırır. Eğer bulamazsa sayaç 20 döndürür ve bulunamadığı anlaşılır.
* **Binary Search(İkili Arama):** Öncelikle Binary Search kullanmak için dizinin sıralı gelmesi gereklidir. Çünkü büyük küçük şeklinde bir algoritma izler. Algoritma başlangıcında dizinin ortasındaki değeri aranan değer ile karşılaştırma yapar. Eğer aranan değer ortanca değerden küçük ise dizinin küçük olan yarısına bakmayarak zaman kazanmış olur. Sonrasında diğer yarının ortanca değeri ile karşılaştırır. Aranan değer bu değerden küçükse soldaki yarı ile büyükse sağdaki yarı ile devam eder. Bu şekilde bula bula yarıya bölerek giden bir algoritma elde etmiş olur. Ortanca değer ona eşit olana kadar devam eder. Eğer bulamazsa -1 değer döndürür.

# **3. SIRALAMA ALGORİTMALARI**

## **3.1 Programın Detaylı Çalışma Mantığı**

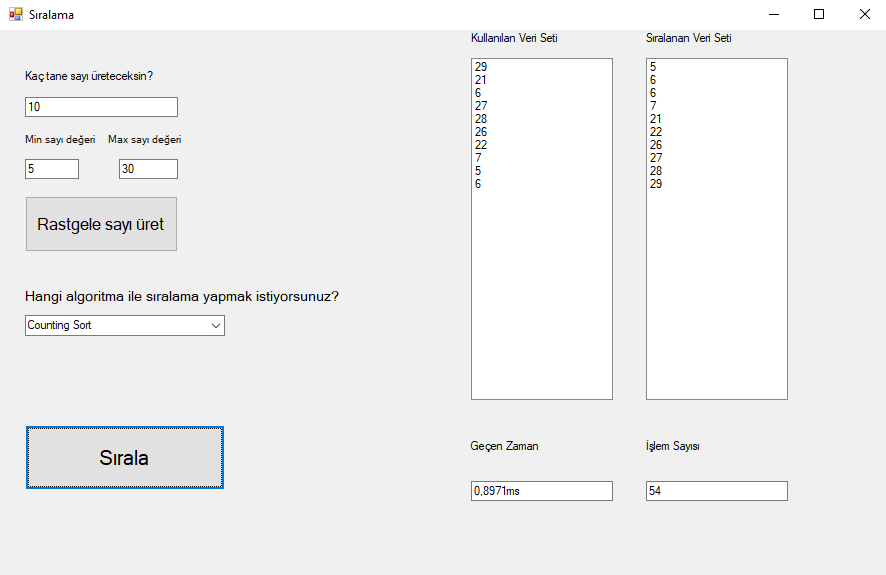
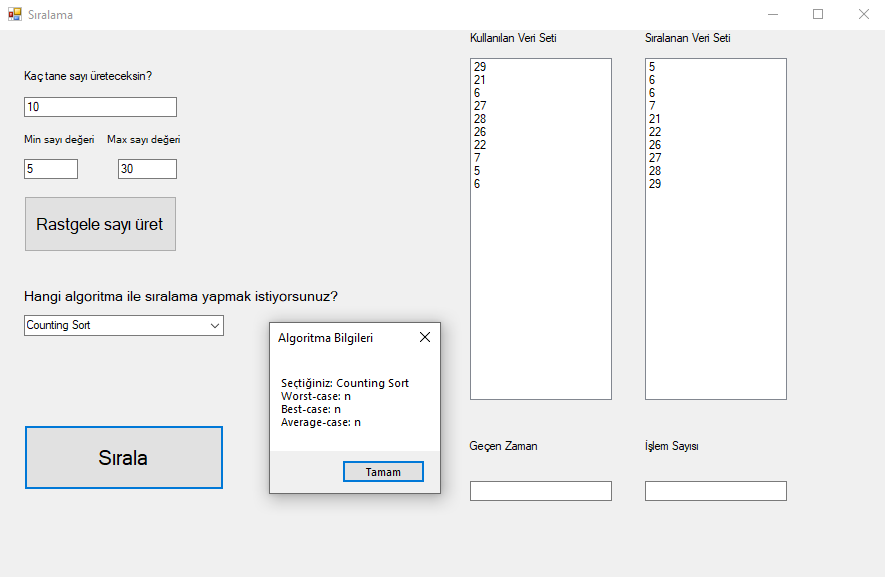
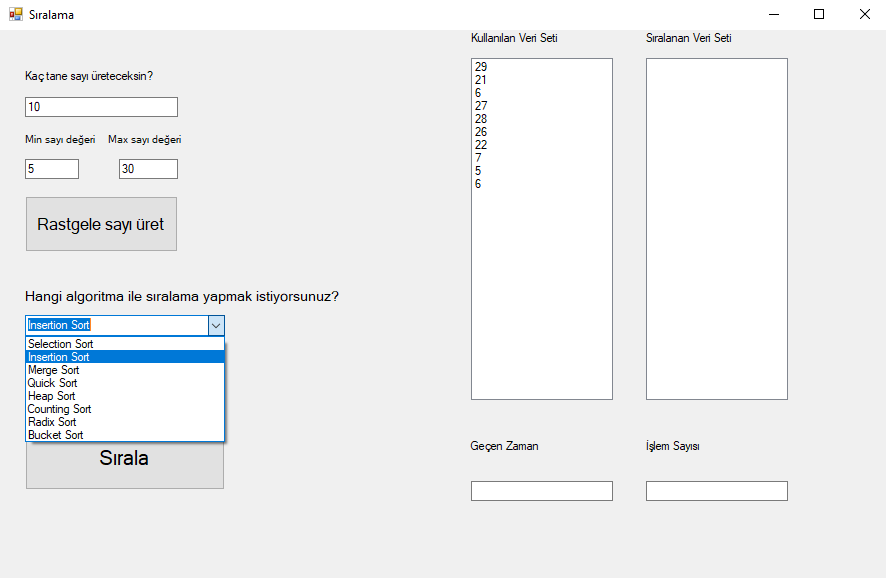
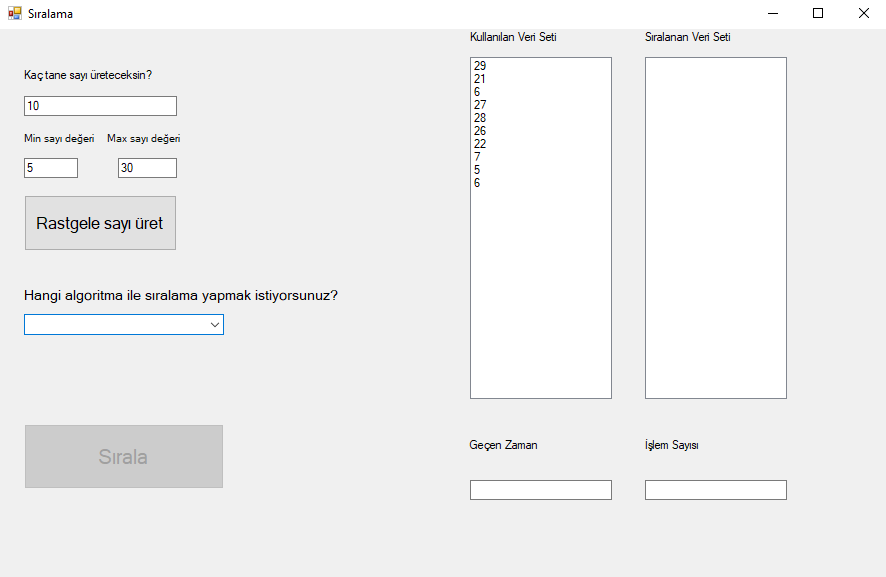
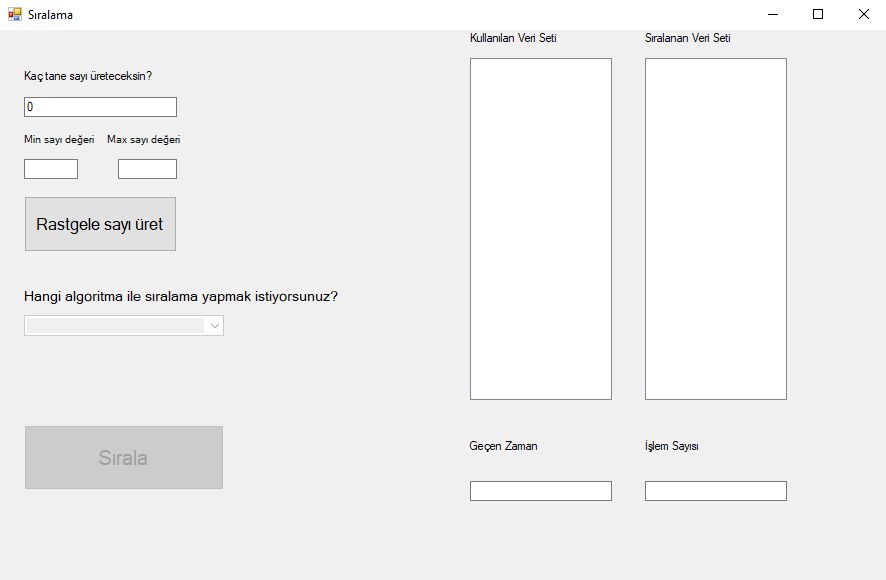
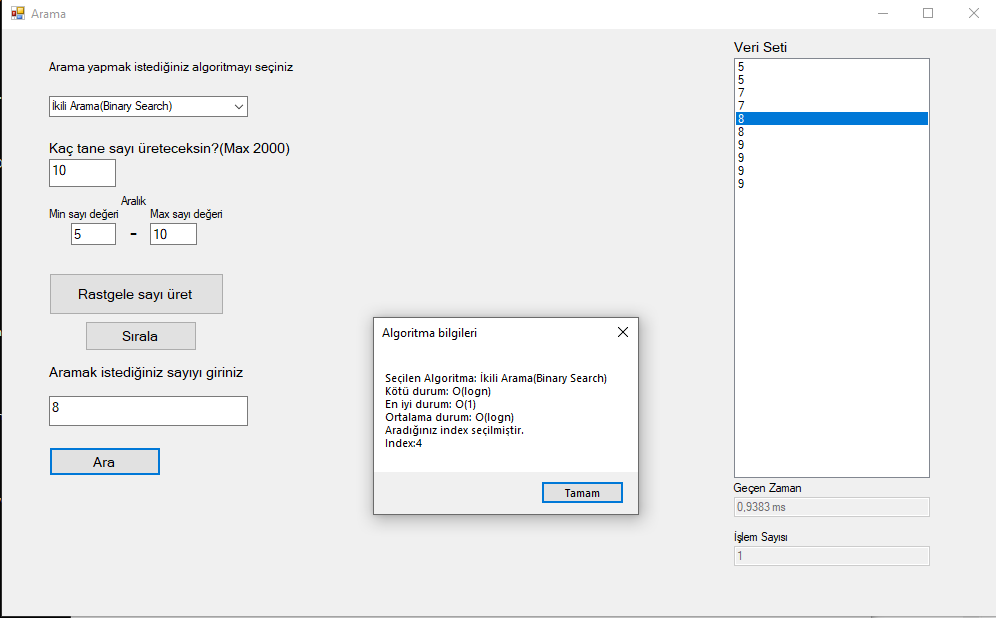
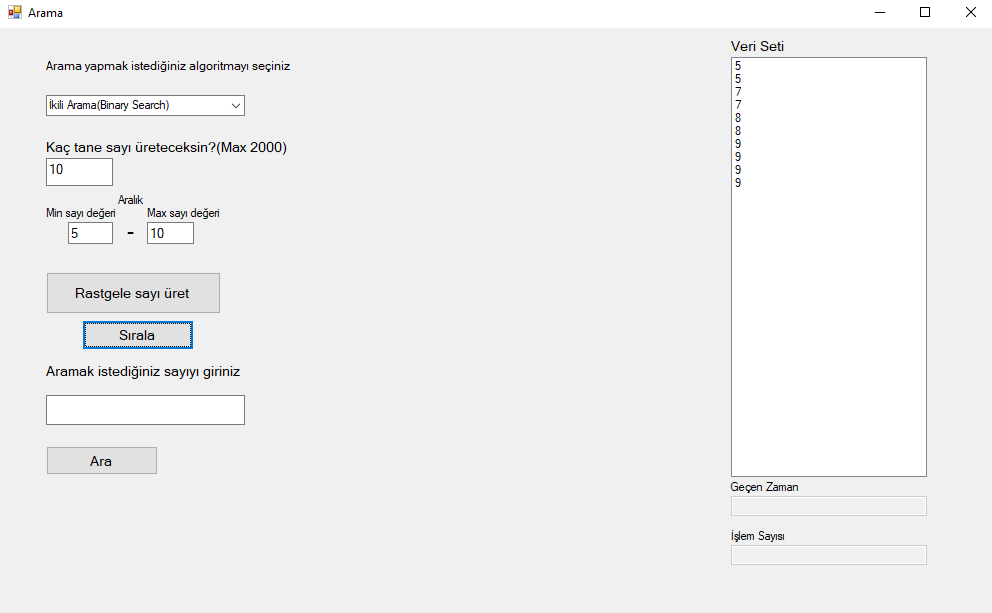
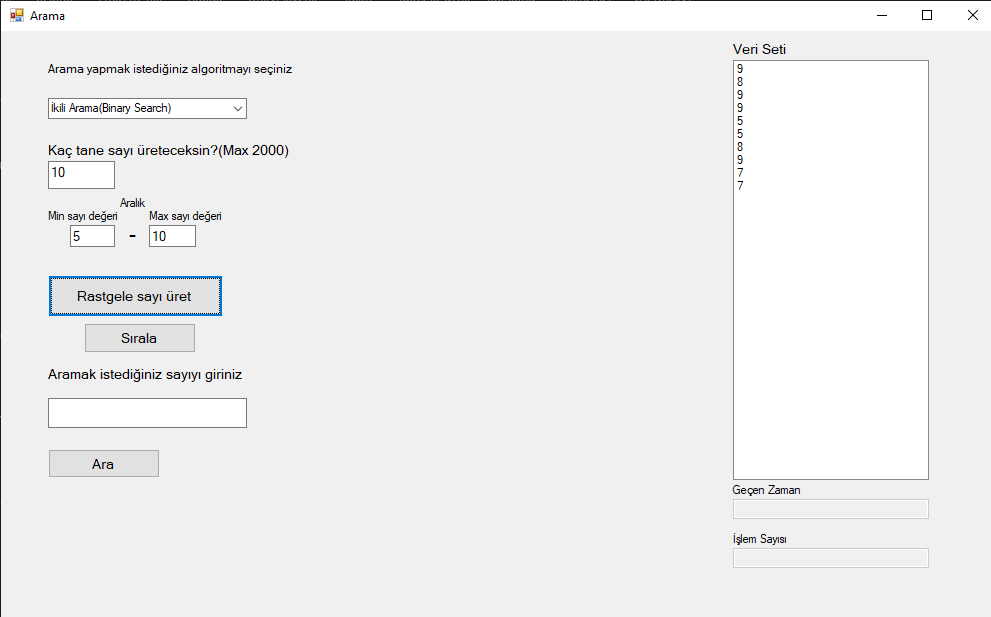
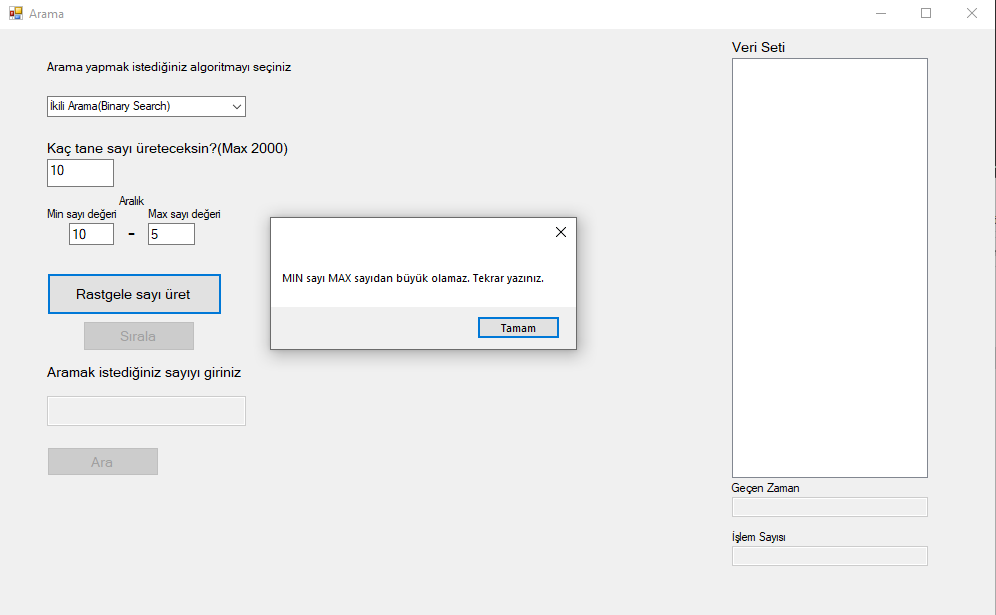
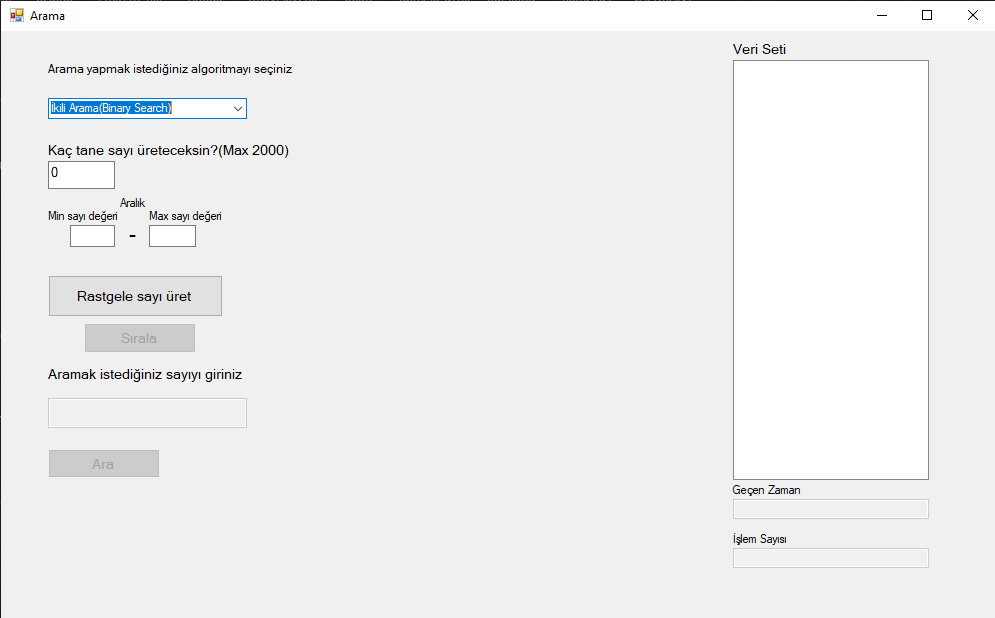
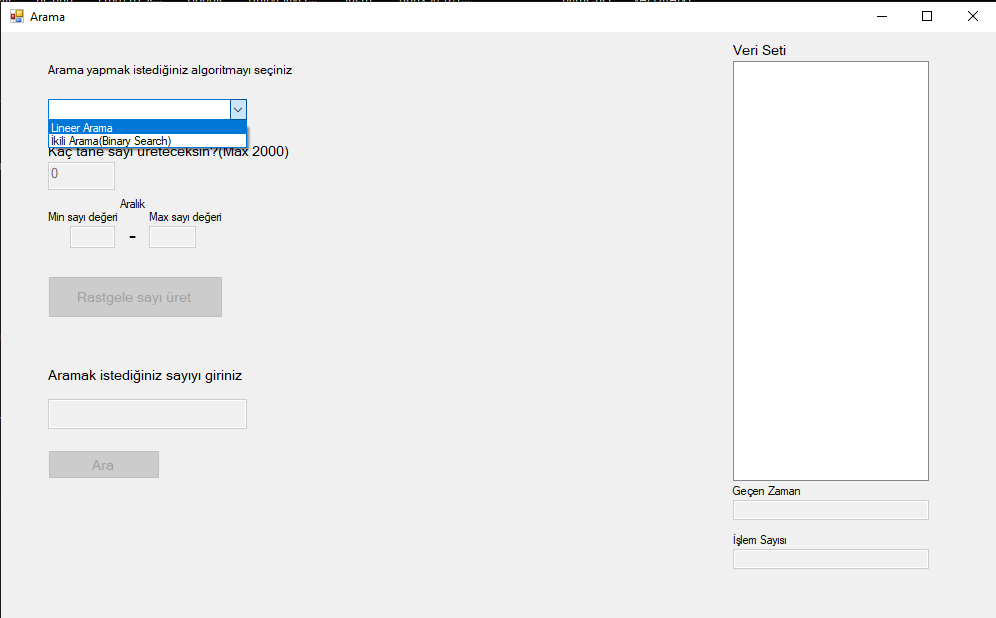
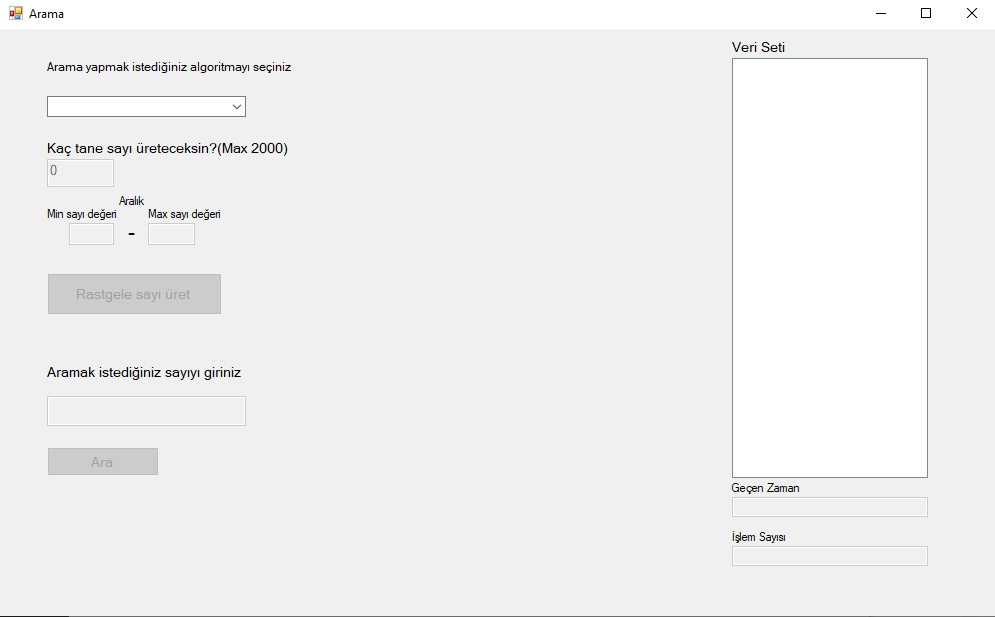
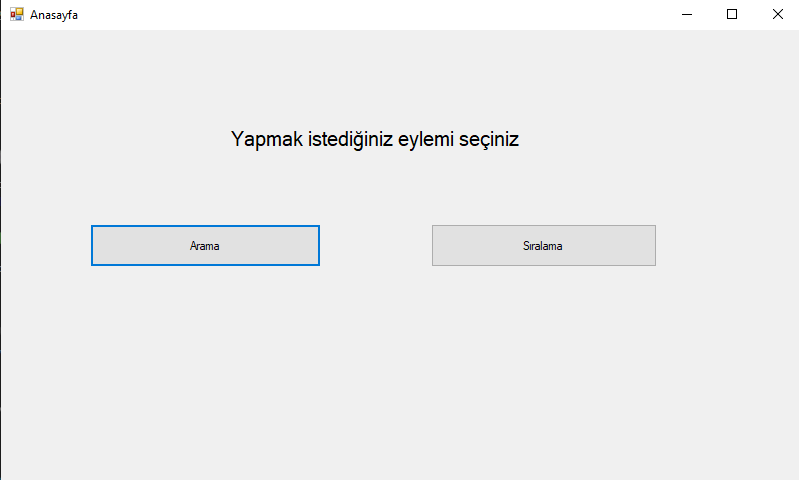
* **Rastgele Sayı Üretimi:** Minimum ve maksimum değerlikte iki sayı arasındaki sayılardan kaç tane üretilecekse onun değeri istenen bir bölüm. RastgeleSayi metodu ile sağlanıyor. Aynı zamanda programın sağlığı için yalnızca 9999 kadar sayı üretilebilir. Rastgele sayı üret butonuna tıklandığında listBox1’e rastgele dizilmiş veri seti gelir.
* **Algoritma Seçme:** ComboBox aracı ile 8 adet sıralama algoritmasının seçimi sağlanır. Arka planda bu algoritma ile rastgele üretilmiş veri seti kullanılarak sıralanması sağlanır.
* **Sıralama:** Bir buton ile sıralamanın gerçekleşmesi için emir verilir. listBox2 adlı veri setine sıralanmış verilerin girişi sağlanır. Aynı zamanda geçen zaman ve işlem sayısı parametrelerini de ekranda görebilirsiniz.

**Not: Hiçbir textbox’a harf girilemez. Min, max sayı değerleri -9999 +9999 arası sınırlıdır.**

## **3.2 Algoritmaların Detaylı İncelenmesi**

* **Selection Sort:** İlk başta dizideki birinci en küçük elemanı tüm elemanları gezerek bulur ve en başa yazar. Daha sonra ikinci en küçük elemanı arayışa çıkar. Onu da bulduktan sonra bir önceki küçük elemanın yanındaki index ile yer değiştirir. Bu işlemi tüm elemanlar sıralanıncaya kadar sıradaki elemanlarıyla tekrar eder.
* **Insertion Sort:** Algoritmada dönülen her bir tur yeni gelen değerler sonda başa doğru karşılaştırarak yerini bulur. Sıra sıra gelen sayılar yerlerini buldukça sıralama oluşur. Tüm dizi sıralanmış olur.
* **Merge Sort:** Rastgele olan verinin sıralı tutulması için geliştirilmiş sıralama algoritmalarından olan merge sort, sıralanacak olan diziyi ikişer elemanlarını birleştirerek aralarında sıralayarak birleştirir. En son elde edilen dizi sıralı dizinin kendisidir. Merge Sort böl ve fethet algoritmalarına örnek verilebilir. İçlerinde sıralı iki veri grubunu birleştirerek başka bir sıralı veri grubu elde eder.
* **Quick Sort:** Diziden bir adet pivot eleman alınmalı. Bu pivot elemandan küçük olanları bir diziye, büyükleri başka bir diziye toplanır. Bu alt dizilerden aynı şekilde içlerinden pivot eleman seçip aynı işlem uygulanır. İç içe küçük parçalara ulaşana kadar bu yöntem sürdürülür. Sonrasında oluşan küçük parça dizilerin birleştirilmesiyle sıralama oluşur.
* **Heap Sort:** Sıralama yapısı ; dizinin ilk elemanı her zaman büyük olacaktır. Dizinin üzerinde bulunan i. Eleman ile çocukları 2\*i ve 2\*i+1 karşılaştırılıp büyük olan elemanlar yer değiştirilecektir. Dizinin son elemanları dizinin tam ortasındaki elemanların çocuk düğümü olacağından bu işlem dizinin yarısına kadar yapılır. Bu şekilde zamandan da kazanılır. Elde edilen diziyi sıralamak için ise dizinin ilk elemanı en büyük olduğundan sıralamaya son eleman ile ilk elemanı yer değiştirilir. Baştan aşağı bozulan dizeye yukarıda yazılan işlemler tekrar tekrar uygulanır. Dizin boyutu 1 oluncaya kadar işleme devam edilir.
* **Counting Sort:** Basitçe sıralanacak olan dizideki her sayının kaç tane olduğunu farklı bir dizide sayar. Daha sonra bu sayıların bulunduğu dizinin üzerinde bir işlemler sıralanmış olan diziyi elde eder.
* **Radix Sort:** Sıralanacak olan verilerin tamsayı olduğu durumlarda kullanılan bu algoritma ilk olarak sıralanacak olan veri kümesindeki elemanların en büyük elemanının kaç basamaklı olduğunu tespit ettikten sonra sayıların en değersiz olan basamağından itibaren incelenmeye başlanır en değerli haneye kadar devam eder ve yeni bir diziye yerleştirilir. Bu işlem dizinin en büyük elemanının basamak sayısı kadar tekrar edilir. Her basamak işlemi için yeni bir bellek alanı gerektirdiği için bellekten dezavantaj sağlar.
* **Bucket Sort**: Sıralanacak olan veriyi parçalara ayırarak sınırlı sayıdaki kovalara atan bir sıralama algoritmasıdır. Ayrıştırılma aşaması adında her kova kendi farklı bir sıralama algoritması kullanılarak ya da yine kova sıralamasının özyinelemeli olarak çağırarak sıralar.

# **4. PROGRAMA AIT FOTOĞRAFLAR**



# **5. SONUÇ**

Bazı sıralama ve arama algoritmalarının nasıl çalıştığını, ne kadar sürede çalıştığını, arama ve sıralama algoritmalarının içerisinde ne kadar işlem yaptığının bilgilerinin paylaşıldığı bir program.