

# Algoritma Analizi ve Tasarımı

## Proje

**Burak AYTAŞ**

### 1. Giriş

Bu proje de 7 adet Sıralama uygulaması ve 2 adet Arama uygulamasından bahsedeceğiz. Bunları c# üzerinde kodlanmış bir görsel uygulamada inceleyip göreceğiz.

### 2. Arama Algoritmaları

Arama algoritmaları, bilgisayar bilimleri için önemli konulardan biridir. Arama algoritmaları adından da anlaşılacağı gibi, bir veri yapısı(data structure) üzerinde belirli bir verinin aranması için kullanılan algoritmalar bütünüdür. Bu veri yapısı bir liste, array ya da graph olabilir. Belirli durumlara özgü farklı arama algoritmaları geliştirilmiştir ve hepsinin birbirine göre farklı avantajları ve farklı çalışma süreleri vardır.

#### 2.1 Linear Search

Bilgisayar biliminde, doğrusal arama veya sıralı arama, bir liste içindeki bir öğeyi bulmak için bir yöntemdir. Bir eşleşme bulunana veya tüm liste aranana kadar listenin her bir öğesini sırayla kontrol eder. Linear Search en kötü zamana sahiptir liste uzunluğu (n) kadar arama yapar.

#### 2.2 Binary Search

İkili arama, sıralanmış öğeler listesinden bir öğeyi bulmak için etkili bir algoritmadır.

Olası konumları tek bir yere daraltıncaya kadar, listenin öğeyi içerebilecek bölümünü tekrar tekrar ikiye bölerek çalışır.

### 3. Sıralama Algoritmaları

Sıralama algoritmalarını kullanmamızdaki amaç, -algoritmanın isminden de anlaşılacağı üzere- sahip olduğumuz veriyi en hızlı şekilde büyükten küçüğe (ya da küçükten büyüğe) bir sıraya sokmak. Bunun için kullanılan birçok sıralama algoritması mevcut. Bazısı çok hızlı ama yazımı zor, bazısı az sayıda veri için çok hızlı, bazısının da yazması kolay.

#### 3.1 Insertion sort

Ekleme sıralaması, her seferinde bir öğe olmak üzere son sıralanan diziyi (veya listeyi) oluşturan basit bir sıralama algoritmasıdır. Büyük listelerde hızlı sıralama, yığın sıralaması veya birleştirme sıralaması gibi daha gelişmiş algoritmalarından çok daha az etkilidir.

**Bununla birlikte, ekleme sıralaması birkaç avantaj sağlar:**

- Diğer ikinci dereceden sıralama algoritmaları gibi (oldukça) küçük veri kümeleri için verimli
- Yerinde; yani, yalnızca sabit miktarda  $O(1)$  ek bellek alanı gerektirir
- Pratikte, seçim sıralama veya kabarcık sıralama gibi diğer basit ikinci

dereceden (yani  $O(n^2)$ )  
algoritmalarından daha verimli

### 3.2 Merge sort

Merge sort , bilgisayar bilimlerinde  $O(n \log(n))$  derecesinde karmaşıklığa sahip bir sıralama algoritmasıdır. Girdi olarak aldığı diziyi en küçük hale gelene kadar ikili gruplara böler ve karşılaştırma yöntemi kullanarak diziyi sıralar.

### 3.3 Heap sort

Bilgisayar biliminde, yığın sıralaması, karşılaştırmaya dayalı bir sıralama algoritmasıdır. Yığın sıralaması, gelişmiş bir seçim sıralaması olarak düşünülebilir: seçim sıralaması gibi, yığın, girdisini sıralı ve sıralanmamış bir bölgeye böler ve sıralanmamış bölgeyi ondan en büyük elemanı çıkararak ve onu sıralanmış bölgeye ekleyerek yinelemeli olarak küçültür. Seçim sıralamanın aksine, yığın sıralaması, sıralanmamış bölgenin doğrusal zaman taramasıyla zaman kaybetmez; bunun yerine, yığın sıralama, her adımda en büyük öğeyi daha hızlı bulmak için bir yığın veri yapısındaki sıralanmamış bölgeyi korur.

### 3.4 Quick sort

Quicksort, verimli bir sıralama algoritmasıdır. İngiliz bilgisayar bilimcisi Tony Hoare tarafından 1959'da geliştirilen ve 1961'de yayınlanan bu algoritma, sıralama için hala yaygın olarak kullanılan bir algoritmadır. İyi uygulandığında, ana rakiplerinden iki veya üç kat daha hızlı olabilir, birleştirme sıralaması ve yığın sıralaması. Quicksort, bir böl ve yönet algoritmasıdır. Diziden bir 'pivot' öğesi seçerek ve diğer öğeleri pivottan küçük veya büyük olmalarına göre iki alt diziyi bölerek çalışır. Alt diziler daha sonra özyinelemeli olarak sıralanır. Bu, sıralamayı gerçekleştirmek için küçük miktarlarda ek bellek gerektiren yerinde yapılabilir.

### 3.5 Counting Sort

Bilgisayar biliminde, sayma sıralaması, küçük tamsayılar olan anahtarlara göre bir nesne koleksiyonunu sıralamak için kullanılan bir algoritmadır; yani tamsayı sıralama algoritmasıdır. Her farklı anahtar değerine sahip nesnelerin sayısını sayarak ve çıktı dizisindeki her bir anahtar değerinin konumlarını belirlemek için bu sayılarda aritmetik kullanarak çalışır. Çalışma süresi, öğe sayısında ve maksimum ve minimum anahtar değerleri arasındaki farkta doğrusaldır, bu nedenle yalnızca anahtarlardaki varyasyonun öğe sayısından önemli ölçüde daha fazla olmadığı durumlarda doğrudan kullanım için uygundur. Bununla birlikte, genellikle daha büyük anahtarları daha verimli bir şekilde işleyebilen başka bir sıralama algoritması olan radix sıralamasında bir alt rutin olarak kullanılır.

### 3.6 Bucket Sort

Kova sıralama veya bin sıralama, bir dizinin öğelerini birkaç kümeye dağıtarak çalışan bir sıralama algoritmasıdır. Her bir kova daha sonra ya farklı bir sıralama algoritması kullanılarak veya kova sıralama algoritmasını yinelemeli olarak uygulayarak ayrı ayrı sıralanır. Bu bir dağıtım türüdür, güvercin deliği türünün bir genellemesidir ve en çok ila en az anlamlı rakam çeşidinde radix türünün kuzenidir. Kova sıralaması, karşılaştırmalarla uygulanabilir ve bu nedenle, bir karşılaştırmalı sıralama algoritması olarak da düşünülebilir. Hesaplama karmaşıklığı, her bir grubu sıralamak için kullanılan algoritmaya, kullanılacak grup sayısına ve girdinin tek tip olarak dağıtılıp dağıtılmadığına bağlıdır.

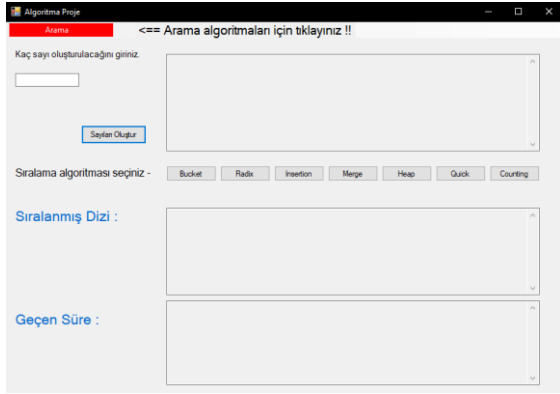
### 3.7 Radix Sort

Bilgisayar biliminde, radix sıralama, karşılaştırmalı olmayan bir sıralama algoritmasıdır. Temellerine göre elemanlar oluşturup kovalara dağıtarak karşılaştırmayı önler. Birden fazla anlamlı basamağı olan

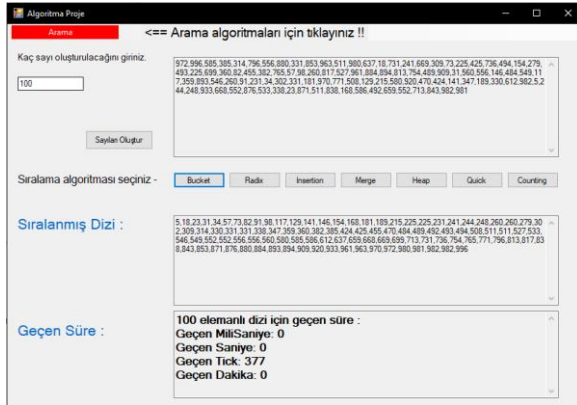
elemanlar için, bu grupta işlemi, tüm basamaklar dikkate alınana kadar önceki adımın sıralaması korunarak her basamak için tekrarlanır. Bu nedenle, radix sıralama, kovalı sıralama ve dijital sıralama olarak da adlandırılmıştır.

#### 4. Yöntem

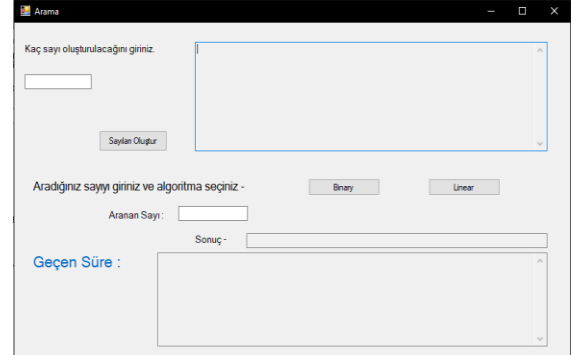
Program çalıştığında alttaki gibi bir pencere sizi karşılar. Bu kısımda Sıralama algoritmaları mevcut ancak bunları çalıştırmak için öncelikle dizinin eleman sayısı verilmeli ve random bir dizi oluşturulmalı.



Dizi oluşturulduktan sonra istediğiniz sıralama algoritmasını seçerek işlemi yaptırabilirsiniz. Dizi eleman sayısı girilmeden çalışmaması için yapılan kontrol dizinin eleman sayısı 0 olup olmadığıdır. Seçilen işlem yapıldıktan sonra ise aşağıdaki örnek gibi bir çıktı alınır.



Arama algoritmaları için ise sol üste köşeye tıklandıktan sonra ulaşılabilir. Bu kısımda önce Kaç sayılıklı dizi olacağı sonra ise Aranan sayıyı girmeniz gerekmektedir.



Program bu şekilde çalışmaktadır. geçen süre her işlem için verilmiştir.