



Course Name: YAPISAL PROGRAMLAMAYA GİRİŞ

Course Group: Group 2

Instructor Name: Dr. Öğr.Üyesi Hafıza İrem TÜRKMEN ÇİLİNGİR

DÖNEM PROJESİ

Student Id: 16011706

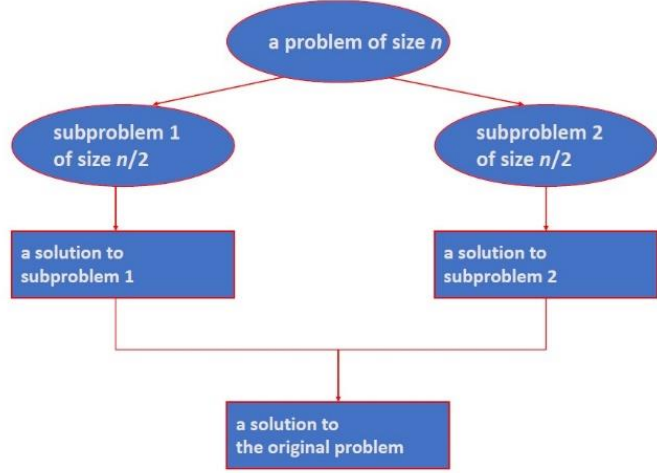
Student Name and Surname: Duygu Erduran

QUICK SORT (HIZLI SIRALAMA) ALGORTİMASI

Hızlı Sıralama 1960 yılında Charles Antony Richard Hoare tarafından geliştirilmiştir. Dünyada yapılmış en iyi kullanılan 10 algoritmalarından biri olarak kabul edilir. Böl ve yönet Algoritma mantığıyla çalışan Sıralama algoritmalarındandır. Diziyi iki alt diziye bölerek(divide) çalışır. Özyineli(recursive) bir algoritma olmasından dolayı kendi kendini tekrar ederek seçtiği elemana göre sıralama işlevini yürütmektedir. Daha sonra ise her alt diziyi birbirinden bağımsız olarak sıralar. Sıralanan diziler birleştirilir.(Conquer).

Böl (Divide) ve Yönet (Conquer):

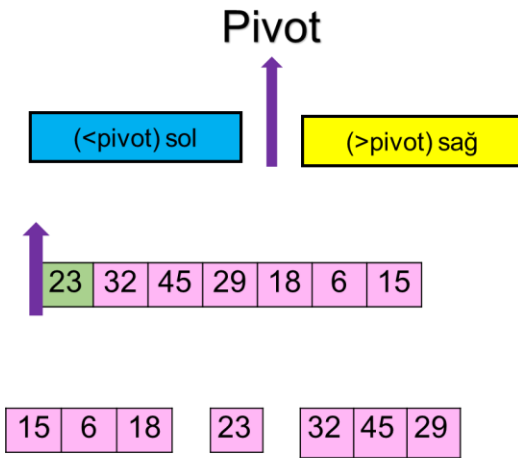
Eski Roma zamanında Romalılar düşmanlarını ilk önce birbirlerine düşürürler yani bu bölmek (divide) olur. Daha sonra mücadele ederler. Bütün olarak mücadele etmek daha zor bölerseniz yönetmesi daha kolaydır. Günümüzde de bu yönetim idare şeklidir.



Design and Analysis of Algorithms - Chapter 4

Algoritmanın Tanımı :

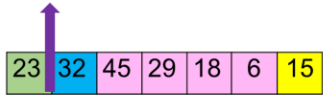
Adım 1: Dizi karıştırılır (shuffle). Daha sonra Pivot seçilir. Pivot ilk eleman, son eleman, rasgele bir değer yada medyan(ortanca değer) olabilir. Biz ilk elemanı seçelim.



Adım 1:

Dizi karıştırılır (shuffle). Daha sonra Pivot seçilir. Pivot ilk eleman, son eleman, rasgele bir değer yada medyan(ortanca değer) olabilir. Biz ilk elemanı seçelim.

pivot



sol

sağ

Adım 2:

Sağ ve sol grupların oluşması için pointer kulan.

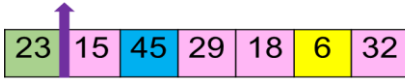
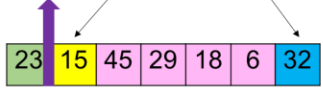
pivottan küçük olanlar sol grubu, pivottan büyük olanlar sağ grubu oluşturacaktır.

Adım 3:

Eğer sayı, sol gruba aitse sol sayacı artır

Eğer sağdaki sayı sağ gruba aitse sağ sayacı azalt.

Diğer gruba ait eleman görürsen elemanların yerini değiştir



sol

sağ

Adım 4:

Yer değiştirdikten sonra sol ve sağ sayacı ilerlet.

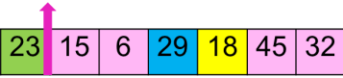


sol

sağ

Adım 5:

Yer değiştir.

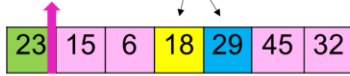


sol

sağ

Adım 6:

Yer değiştir.



sağ

sol

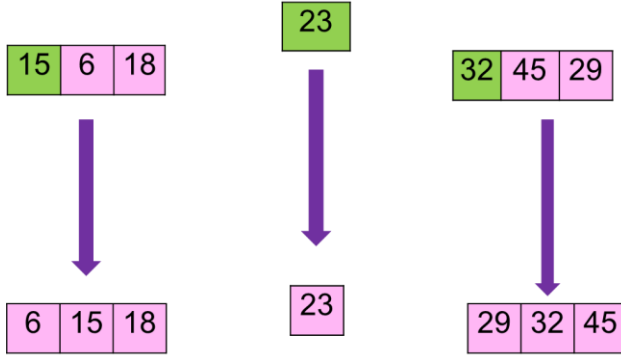


sol

sağ

Adım 7:

Sol ve sağ pointer yer değiştirirse, sağ pointer ile pivotun yerini değiştir.



6 15 18 23 29 32 45

Adım 8:

Quick sort algoritması iki alt gruba
Özyineli(recursive) olarak uygulanır.

Alt gruplar bitince dizi birleşir.

Algoritmanın Karmaşıklığı:

En İyi Durum (Best-Case): $O(n \log n)$ (ortaya yakın veya ortanca eleman)

Ortalama Durum (Average-Case): $O(n \log n)$ (rasgele sayı)

En Kötü Durum (Worst-Case): $O(n^2)$ (küçükten büyüğe sıralı ise)

Algoritmanın Kısıtları:

Dizinin sıralı olma ihtimalidir.

İyileştirme:

1. Diziyi karıştırıp sıralayabiliriz
2. Diziden 3 sayı seçilir ve 2.ci büyük sayı pivot yapılarak sıralanır.

Uygulama Alanları:

Daha çok büyük problemler için kullanılır. Java ve C programla dilleri sıralama için Quick Sort kullanırlar.

Algoritmanın Avantajları:

Quick Sort daha geniş bir kullanım alanına sahiptir. Basit bir yapıdadır bu yüzden gerçekleştirilmesi zor değildir. Yedek dizi kullanmadan diziyi shuffle eder. Diziyi karıştırmasının sebebi de karmaşıklığı artmaması içindir. Dizi iki alt parçalara bölüp rekürsif olarak sıralar. Daha az karşılaştırma yapmış olur. Birleştirme işlemini yapar. Belirli uygulamalarda diğer sıralama algoritmalarından büyük ölçüde daha hızlı çalışabilir.

Algoritmanın Dezavantajı:

Quick Sort'un olumsuz yönüne ise şunu söyleyebiliriz:

Eğer parçalama işlemi sonucu oluşan alt dizilerin boyutu çok dengesiz olursa oldukça verimsiz bir algoritma haline gelebilmektedir.

Örneğin; ilk parçalama işlemi en küçük elemana göre, ikinci parçalama bir sonraki en küçük elemana göre yani elimizdeki verilerin küçükten büyüğe doğru sıralı olduğunu düşünüyoruz. Diğer parçalamalarda benzer şekilde yapılmaya devam ederse, her çağrımda program, sadece bir elemanı diziden ayırır ve büyük boyutlu alt dizileri parçalamak için çok sayıda parçalama işleminin yapılmasına neden olmaktadır. Hızlı sıralamanın parçalamadan önce diziyi shuffle yani karıştırmasının da nedeni de bu olumsuz durumu önlemektir. Shuffle sonucunda dizinin art arda dengesiz parçalanmaların oluşma ihtimali çok düşük olduğundan, bu ihtimal göz ardı edilebilmektedir.

Diğer Sıralama Algoritmaları ile Karşılaştırma:

Algoritma	İngilizcesi	Algoritma Analizi			Kararlılık	Yöntem	Açıklama	inplace	stable
		En İyi	Ortalama	En Kötü					
Seçerek Sıralama	Selection Sort	n^2	n^2	n^2	Kararsız	Seçerek		Var	Yok
Hızlı Sıralama	Quick Sort	$n \log(n)$	$n \log(n)$	n^2	Kararsız	Parçala Fethet		Var	Yok
Birleştirme Sıralaması	Merge Sort	$n \log(n)$	$n \log(n)$	$n \log(n)$	Kararlı	Parçala Fethet		Yok	Var
Sokma Sıralaması	Insertion Sort	n	$d+n$	n^2	Kararlı	Sokma	d yer değiştirme sayısıdır ve n^2 cinsindendir	Var	Var

Inplace: Aynı yerde sıralama yapılabilmesidir.

Stable: Sıralama yaparken bir adım önceki sırayı bozmamasıdır.

Quick Sort merge sort'un tümleyenidir. **Merge sort'ta** sıralanacak dizi iki alt diziye bölünür ve daha sonra sıralanmış alt diziler birleştirilerek, tüm dizi sıralı hale getirilir. **Hızlı sıralama da ise;** dizi yeniden düzenlenir. Öyle ki iki alt dizi sıralı olduğunda tüm dizi sıralanmış olur.

Merge Sort'ta tüm dizi üzerinden işlem yapmadan önce iki kez özyineli çağrı yapılır.

Quick Sort'ta ise; tüm dizi üzerinden işlem yapıldıktan sonra iki kez özyineli çağrı yapılır.

Birleştirmeli sıralama da dizi orta noktadan iki parçaya bölünürken, **Quick Sort'ta** ise parçaların ayırma noktası dizinin içeriğine bağlıdır. Aslında bakarsak sıralama algoritmalarından selection sort tüm diziye bakarak sıralama yapıyordu. **Merge Sort** bunu çözmek için kontrolsüz parçalara bölüyor, merge ederken sıralamayı sağlıyordu. Quick Sort ise önce diziyi karıştırıp kontrollü bir şekilde parçalara böler.

Küçük diziler için **Insertion Sort** kullanmalıyız. Recursion overread'ten kaçınmak için daha uygun seçenektir. milyar elemanı olan büyük dizimiz var ve yer problem yaşıyorsak **quick sort** kullanmamız gerekir. Dizinin arada bir sıralı olmasından endişemiz varsa ve saniyeler bizim için çok değerliyse; iyileştirme yapıp (ilk önce diziyi sıralayıp) **quick sort** kullanabiliriz. Yada diziyi **merge sort** yapmakta bir seçenektir.

KAYNAKÇA:

URL:

- <https://en.wikipedia.org/wiki/Quicksort>
- <http://bilgisayarkavramlari.sadievrenseker.com/2008/08/09/hizli-siralama-algoritmasi-quick-sort-algorithm/>
- 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

KİTAP:

- Algoritmalar Dördüncü Basımdan Çeviri, Robert Sedgewick | Kevin Wayne, Çeviri Editörü: Şadi Evren Şeker

Youtube Video Linki: <https://youtu.be/wByfS3Wlq7M>