

Proje Raporu

Projenin Konusu: Quick Sort ve Bubble Sort Algoritmalarını Kullanarak Sıralama. (Kodların çalıştırılması ve çalışma zamanlarının hesaplanıp karşılaştırılması, zaman karmaşıklığının hesaplanması.)

Programlama dili olarak C dili, Editör olarak da dev c++ yazılımı kullanılmıştır. Projemizi çalıştırdığımızda bizi bir menü karşılıyor bu menüde işlemlerimiz için 4 farklı seçenek vardır. Bu seçeneklerden ilk 3'ü varsayılan olarak sırasıyla 1000, 2000, 3000 elemanlı bir dizi ve içerisine sırasıyla 1-1000, 1-2000, 1-3000 sayı aralığında random sayılar üretilerek dizilere yazılmıştır. Daha sonra 4.seçeneğimizde ise elle sırasıyla dizi boyutu belirleme ve random sayı aralığı belirtebileceğimiz klavyeden girdi alınmıştır.

Bubble sort: Kabarcık algoritması sıralanacak dizinin üzerinde sürekli ilerlerken her defasında iki ögenin birbiriyle karşılaştırılıp, karşılaştırılan öğelerin yanlış sırada olmaları durumunda yerlerinin değiştirilmesi mantığı ile çalışır

Quick sort: Hızlı sıralama algoritması sıralanacak bir diziyi sürekli iki parçaya bölerek oluşan bu küçük parçaların kendi aralarında sıralanması mantığı ile çalışır.

Projenin Çalıştırılması:

```
#####
BUBBLE SORT & QUICK SORT
#####
1) 1000 elemanli dizi
2) 2000 elemanli dizi
3) 3000 elemanli dizi
4) Elle belirle
q) cikis
#####
seciminiz:
>
```

4. Seçeneği Seçtiğimizde;

```
#####
seciminiz:
> 4
Dizi boyutu belirtmek için bir sayi giriniz:
>40000
Uretilcek dizi icin rastgele sayilarin arali-ini sirayla belirtin..
>(min)
>2500
(max)
>30000
40000 boyutlu dizi ile calisilacaktir...

Bekleme durumunda bir tusa basarak devam edebilirsiniz...
```

```
3715 15393 24977 13670 5356 12024 6638 17640 3947 6581 12261 10475 3927 28489 16067 18324 14285 13775
17334 26224 11367 21952 5865 20229 23132 27141 5250 8621 15218 13531 26981 19108 18115 23940 3174 198
24257 13884 7271 23071 24696 7791 22391 25031 14355 21801 24292 7146 7842 20430 7235 13097 25882 26
22233 3004 28829 4255 7540 22606 19452 16368 6301 28942 29307 3085 25757 8451 14292 18697 10081 23
29054 21194 22049 7320 17710 20268 4978 15347 16543 23956 10683 12386 7823 17235 26134 28111 23818
14549 13046 15929 7549 5914 18092 3922 25247 3581 3656 23290 16428 10053 22751 20112 20594 23194 8876
5382 9137 13881 13307 7273 7500 5404 10213 19138 13004 2833 12006 22119 7208 19142 24893 4373 22852 9
539 4245 27102 12471 14014 23780 10733 3025 13530 Rastgele uretilen diziyi 2 algoritma icin kullanilacagindan y
eni bir diziyi kopyalıyoruz.
```

Bu girdileri yaptıktan sonra bize kısa bir bilgi aktarılıyor bilgi aktarıldıktan sonra bir tuşa basana dek konsol ekranı bekletiliyor bu esnada herhangi bir tuşa basarak devam edebilirsiniz. Bu aşamadan sonra random üretilen sayılar ekrana bastırılıyor ve yine bir bilgi aktarılıyor. Üretilen dizi iki algoritma içinde kullanılacağından üretilen bu diziyi başka bir diziyi kopyalıyoruz. Kopyalama işleminden sonra üretilen dizi hakkında bilgi veriyor.

```
7476 8905 16537 21039 14411 24974 3693 7049 20476 6787 20924 15447 34
3715 15393 24977 13670 5356 12024 6638 17640 3947 6581 12261 10475 39
17334 26224 11367 21952 5865 20229 23132 27141 5250 8621 15218 13531
07 24257 13884 7271 23071 24696 7791 22391 25031 14355 21801 24292 7146
891 22233 3004 28829 4255 7540 22606 19452 16368 6301 28942 29307 3085
967 29054 21194 22049 7320 17710 20268 4978 15347 16543 23956 10683 123
14549 13046 15929 7549 5914 18092 3922 25247 3581 3656 23290 16428 100
5382 9137 13881 13307 7273 7500 5404 10213 19138 13004 2833 12006 221
539 4245 27102 12471 14014 23780 10733 3025 13530
40000 adet 2500 ile 30000 arasında rastgele sayılar uretildi ve ekrana basildi...
```

Devam etmek için herhangi bir tuşa basıyoruz. Algoritmanın çalıştırılacağı hakkında bilgi veriyor.

```
5382 9137 13881 13307 7273 7500 5404 10213 19138 13004 2833 12006 22119
539 4245 27102 12471 14014 23780 10733 3025 13530
40000 adet 2500 ile 30000 arasında rastgele sayılar uretildi ve ekrana basildi...
Uretilen dizi Bubble sort siralama algoritmasi ile calistiriliyor..
```

Devam ettiğimizde kısa bir süre ekranda bizi bekletebilir. Daha sonra çalışmaya başlayacaktır. Sıralama işlemi bittiğinde bize kaç saniyede bittiği bilgisini veriyor ve ardından diğer algoritma çalışması bekleniyor yapmamız gereken devam etmek.

```
29926 29927 29927 29929 29929 29930 29930 29930 29931 29931 29931 29932 29932 29933 29936 29938 29940 2
9941 29943 29944 29945 29949 29950 29953 29954 29954 29956 29956 29959 29960 29960 29961 29961 29961 29
961 29962 29962 29962 29963 29963 29964 29965 29967 29967 29968 29970 29971 29972 29974 29974 29975 299
76 29977 29977 29978 29979 29980 29982 29983 29983 29984 29985 29986 29986 29987 29988 29989 29991 2999
1 29992 29992 29993 29995 29997 29999 30000 30000 Siralama islemi tamamlandi..
Siralama islemi 10.495000 saniye de bitti...
Uretilen dizi Quick sort siralama algoritmasi ile calistiriliyor..
```

Quick Sort algoritması çalıştırıldığında çalışma saniyesi 0.000000 gibi bir sonuç karşımıza çıkabilir ama 40 bin elemanlı bir dizi ile çalıştığımızda aşağıdaki gibi bir sonuç ekrana gelecektir.

```
29926 29927 29927 29929 29929 29930 29930 29930 29931 29931 29931 29932 29932 29933 29936 29938 29940 2
9941 29943 29944 29945 29949 29950 29953 29954 29954 29956 29956 29959 29960 29960 29961 29961 29961 29
961 29962 29962 29962 29963 29963 29964 29965 29967 29967 29968 29970 29971 29972 29974 29974 29975 299
76 29977 29977 29978 29979 29980 29982 29983 29983 29984 29985 29986 29986 29987 29988 29989 29991 2999
1 29992 29992 29993 29995 29997 29999 30000 30000 Siralama islemi tamamlandi..
Siralama islemi 0.016000 saniye de bitti...
```

İşlemlerimiz tamamen bittiğinde istatistikler ekranı bizi karşılıyor.

40000 elemanlı dizi manuel olarak girildiğinde istatistikler

```
##### İSTATİSTİKLER #####
40000 Elemanlı 2500 minimum 30000 maximum sayı aralı-ındaki random sayılardan oluşan bir dizi ile işlemler yapıldı..
Bubble sort algoritması işlemi 10.495000 saniye de bitirdi...Karsilastırma sayısı: 399089045
Quick sort algoritması işlemi 0.016000 saniye de bitirdi...Karsilastırma sayısı: 390792
#####
```

1000 elemanlı dizi seçildiğinde istatistikler

```
##### İSTATİSTİKLER #####
1000 Elemanlı 1 minimum 1000 maximum sayı aralığındaki random sayılardan oluşan bir dizi ile işlemler yapıldı..
Bubble sort algoritması işlemi 0.109000 saniye de bitirdi...Karsilastırma sayısı: 256912
Quick sort algoritması işlemi 0.000000 saniye de bitirdi...Karsilastırma sayısı: 6464
```

2000 elemanlı dizi seçildiğinde istatistikler

```
##### İSTATİSTİKLER #####
2000 Elemanlı 1 minimum 2000 maximum sayı aralığındaki random sayılardan oluşan bir dizi ile işlemler yapıldı..
Bubble sort algoritması işlemi 0.241000 saniye de bitirdi...Karsilastırma sayısı: 1009481
Quick sort algoritması işlemi 0.000000 saniye de bitirdi...Karsilastırma sayısı: 12793
```

3000 elemanlı dizi seçildiğinde istatistikler

```
##### İSTATİSTİKLER #####
3000 Elemanlı 1 minimum 3000 maximum sayı aralığındaki random sayılardan oluşan bir dizi ile işlemler yapıldı..
Bubble sort algoritması işlemi 0.365000 saniye de bitirdi...Karsilastırma sayısı: 2236758
Quick sort algoritması işlemi 0.000000 saniye de bitirdi...Karsilastırma sayısı: 19719
```

Aşağıdaki tabloda projede kullandığımız Quick Sort ve Bubble Sort algoritmalarının Karşılaştırma Sayıları, Çalışma Süreleri ve Karmaşıklık Analizi verilmiştir.

		1000 Elemanlı Random Dizi	2000 Elemanlı Random Dizi	3000 Elemanlı Random Dizi
Karmaşıklık Analizi	Bubble Sort	n^2	n^2	n^2
	Quick Sort	n^2	n^2	n^2
Çalışma Süresi	Bubble Sort	0.109000	0.241000	0.365000
	Quick Sort	0.000000	0.000000	0.000000
Karşılaştırma Sayısı	Bubble Sort	256912	1009481	2236758
	Quick Sort	6464	12793	19719

Projenin Sözde kodu:

Swap fonksiyonu

```
Algoritma swap(a, b)
    GeçiciDeğer = a'nın değeri
    a'nın değerini b'nin değeri yap
    b'nin değerini GeçiciDeğer yap
SonAlgoritma
```

Bubble sort algoritması

```
Algoritma bubbleSort(n, dizi)
    karsilastirmaB = 0
    s = n
    x = 0

    Döngü x < (s - 1) olduğu sürece
        y = 0

        Döngü y < s - x - 1 olduğu sürece
            Eğer dizi[y] > dizi[y+1] ise
                karsilastirmaB += 1
                swap(&dizi[y], &dizi[y+1])
            Sonuç Eğer
                y = y + 1
        SonDöngü

        x = x + 1
    SonDöngü

    x = 0
    Döngü x < s olduğu sürece
        Yazdır dizi[x]
```

x = x + 1
SonDöngü

Sonuç karsilastirmaB
SonAlgoritma

Quick sort algoritması

Algoritma partition(arr, low, high)
pivot = arr[high]
i = low - 1
karsilastirmaQ = 0

Döngü j = low'dan başlayarak high-1'e kadar olduğu sürece
Eğer arr[j] <= pivot ise
karsilastirmaQ += 1
i = i + 1
swap(&arr[i], &arr[j])
Sonuç Eğer
SonDöngü

swap(&arr[i + 1], &arr[high])

Sonuç (i + 1)
SonAlgoritma

Algoritma quickSort(arr, low, high)
Eğer low < high ise
pi = partition(arr, low, high)

quickSort(arr, low, pi - 1)
quickSort(arr, pi + 1, high)
Sonuç Eğer
SonAlgoritma

Metin renklendirme fonksiyonları

Algoritma hConsole()
Sonuç GetStdHandle(STD_OUTPUT_HANDLE)
SonAlgoritma

Algoritma savedAttrHandleFonk()
GetConsoleScreenBufferInfo(hConsole(), &consoleInfo)
Sonuç consoleInfo.wAttributes
SonAlgoritma

Menu fonksiyonu

Algoritma printMenu()
Yazdır "#####"
Yazdır " BUBBLE SORT & QUICK SORT "
Yazdır "#####"

Yazdır "1) 1000 elemanli dizi"
Yazdır "2) 2000 elemanli dizi"
Yazdır "3) 3000 elemanli dizi"
Yazdır "4) Elle belirle"
Yazdır "q) cikis"
Yazdır "#####"

SonAlgoritma

Dizi kopyalama fonksiyonu

Algoritma diziKopyala(ilkDizi, ikinciDizi, boyut)

i = 0

Döngü i < boyut olduğu sürece

ikinciDizi[i] = ilkDizi[i]

i = i + 1

SonDöngü

Sonuç ikinciDizi

SonAlgoritma

Dizi ekrana yazdırma fonksiyonu

Algoritma diziPrint(dizi, boyut)

i = 0

Döngü i < boyut olduğu sürece

Yazdır dizi[i]

Yazdır " "

i = i + 1

SonDöngü

SonAlgoritma

Random sayı üretme fonksiyonu

Algoritma randNum(min, max)

Sonuç (rastgele() mod (((max) + 1) - (min))) + (min)

SonAlgoritma

Random dizi üretme fonksiyonu

Algoritma randomDiziUret(dizi, boyut, min, max)

i = 0

Döngü i < boyut olduğu sürece

dizi[i] = randNum(min, max)

i = i + 1

SonDöngü

Sonuç dizi

SonAlgoritma

Algoritma işlemleri fonksiyonu

```
Algoritma algoritmaIslem(boyut, min, max)
    basla, bit, basla2, bit2
    saved_attributes = savedAttrHandleFonk()
```

```
    dizi[boyut]
    diziK[boyut]
    a
```

```
    SetConsoleTextAttribute(hConsole(), BACKGROUND_GREEN)
    Yazdır boyut "boyutlu dizi ile çalışılacaktır..."
    Yazdır "\n\nBekleme durumunda bir tusa basarak devam edebilirsiniz..."
    SetConsoleTextAttribute(hConsole(), saved_attributes)
    getch()
```

```
    randomDiziUret(dizi, boyut, min, max)
    diziPrint(dizi, boyut)
```

```
    SetConsoleTextAttribute(hConsole(), BACKGROUND_RED)
    Yazdır "Rastgele üretilen diziyi 2 algoritma için kullanılacağından yeni bir diziye kopyalıyoruz."
    SetConsoleTextAttribute(hConsole(), saved_attributes)
    getch()
```

```
    diziKopyala(dizi, diziK, boyut)
    diziPrint(diziK, boyut)
```

```
    SetConsoleTextAttribute(hConsole(), BACKGROUND_GREEN)
    Yazdır boyut "adet " min " ile " max " arasında rastgele sayılar üretildi ve ekrana basıldı..."
    SetConsoleTextAttribute(hConsole(), saved_attributes)
    getch()
```

```
    SetConsoleTextAttribute(hConsole(), BACKGROUND_RED)
    Yazdır "Üretilen dizi Bubble sort sıralama algoritması ile çalıştırılıyor.."
    SetConsoleTextAttribute(hConsole(), saved_attributes)
    getch()
```

```
    basla = clock()
    karsilastirmaBubble = bubbleSort(boyut, dizi)
    bit = clock()
```

```
    SetConsoleTextAttribute(hConsole(), BACKGROUND_GREEN)
    Yazdır "Sıralama işlemi tamamlandı.."
    SetConsoleTextAttribute(hConsole(), saved_attributes)
```

```
    fark = (bit - basla) / CLOCKS_PER_SEC
```

```
    SetConsoleTextAttribute(hConsole(), BACKGROUND_GREEN |
BACKGROUND_INTENSITY)
    Yazdır "Sıralama işlemi " fark " saniye de bitti..."
    SetConsoleTextAttribute(hConsole(), saved_attributes)
```

```
SetConsoleTextAttribute(hConsole(), BACKGROUND_RED)
Yazdır "Uretilen dizi Quick sort sıralama algoritması ile çalıştırılıyor.."
SetConsoleTextAttribute(hConsole(), saved_attributes)
getch()
```

N = boyut

```
basla2 = clock()
quickSort(diziK, 0, N - 1)
bit2 = clock()
diziPrint(diziK, boyut)
```

```
SetConsoleTextAttribute(hConsole(), BACKGROUND_GREEN)
Yazdır "Sıralama işlemi tamamlandı.."
SetConsoleTextAttribute(hConsole(), saved_attributes)
```

fark2 = (bit2 - basla2) / CLOCKS_PER_SEC

```
SetConsoleTextAttribute(hConsole(), BACKGROUND_GREEN |
BACKGROUND_INTENSITY)
Yazdır "Sıralama işlemi " fark2 " saniye de bitti...\n\n"
SetConsoleTextAttribute(hConsole(), saved_attributes)
getch()
```

```
SetConsoleTextAttribute(hConsole(), BACKGROUND_GREEN | BACKGROUND_RED |
BACKGROUND_INTENSITY)
Yazdır "\n##### İSTATİSTİKLER #####\n"
Yazdır "\n" boyut "Elemanlı " min " minimum " max " maximum sayı aralığındaki random
sayılardan oluşan bir dizi ile işlemler yapıldı..\n\n"
Yazdır "Bubble sort algoritması işlemi " fark " saniye de bitirdi...Karşılaştırma sayısı: "
karşılaştırmalarBubble "\n"
Yazdır "Quick sort algoritması işlemi " fark2 " saniye de bitirdi...Karşılaştırma sayısı: "
karşılaştırmalarQ "\n"
Yazdır "\n#####\n"
SetConsoleTextAttribute(hConsole(), saved_attributes)
SonAlgoritma
```

Main fonksiyonu

```
AnaFonksiyon
saved_attributes = savedAttrHandleFonk()

secim

printMenu()
Yazdır "seciminiz: \n> "
SonsuzDöngü
    srand(time(NULL))

    Oku secim

    Eğer secim 'q' ise
```


Kır

Değilse Eğer secim '1' ise

 algoritmaIslem(BIN, 1, BIN)

Değilse Eğer secim '2' ise

 algoritmaIslem(IKIBIN, 1, IKIBIN)

Değilse Eğer secim '3' ise

 algoritmaIslem(UCBIN, 1, UCBIN)

Değilse Eğer secim '4' ise

 diziBoyutu, min, max

SetConsoleTextAttribute(hConsole(), BACKGROUND_GREEN)

Yazdır "Dizi boyutu belirtmek için bir sayı giriniz: \n>"

SetConsoleTextAttribute(hConsole(), saved_attributes)

Oku diziBoyutu

SetConsoleTextAttribute(hConsole(), BACKGROUND_GREEN)

Yazdır "Uretilcek dizi için rastgele sayıların aralığını sırayla belirtin..\n>"

SetConsoleTextAttribute(hConsole(), saved_attributes)

Yazdır "(min) \n>"

Oku min

Yazdır "(max) \n>"

Oku max

algoritmaIslem(diziBoyutu, min, max)

SonDöngü

SonAnaFonksiyon