

2020-2021 Güz Yarıyılı
Algoritma Analizi
Ödev – 1

Ders Yürütücüleri
Doç. Dr. M. Elif KARSLIGİL
Dr. Öğr. Üyesi M. Amaç GÜVENSAN

Burak Boz

18011706

Soru 1 – a ve b

Problem: N elemanlı bir dizide birbirine en yakın değere sahip iki elemanın bulunması isteniyor.

- Problemi brute-force* yaklaşımı ile çözen algoritmanın C dilinde kodunu yazıp, verimliliğini analiz ediniz.

Çözüm: Öncelikle kullanıcıdan dizinin kaç elemanlı olduğu istenmektedir. Kullanıcının girdiği değer boyutunda, 1 ile 100 arası sayılardan oluşan bir dizi oluşturulmuştur. Sonra dizideki tüm elemanlar brute force yaklaşımıyla tüm elemanlarla tek tek karşılaştırılmış ve farkları alınmıştır. Tespit edilen en küçük fark ekrana yazdırılmıştır.

Karmaşıklık: İç içe n elemanlı iki for döngüsü = n^2

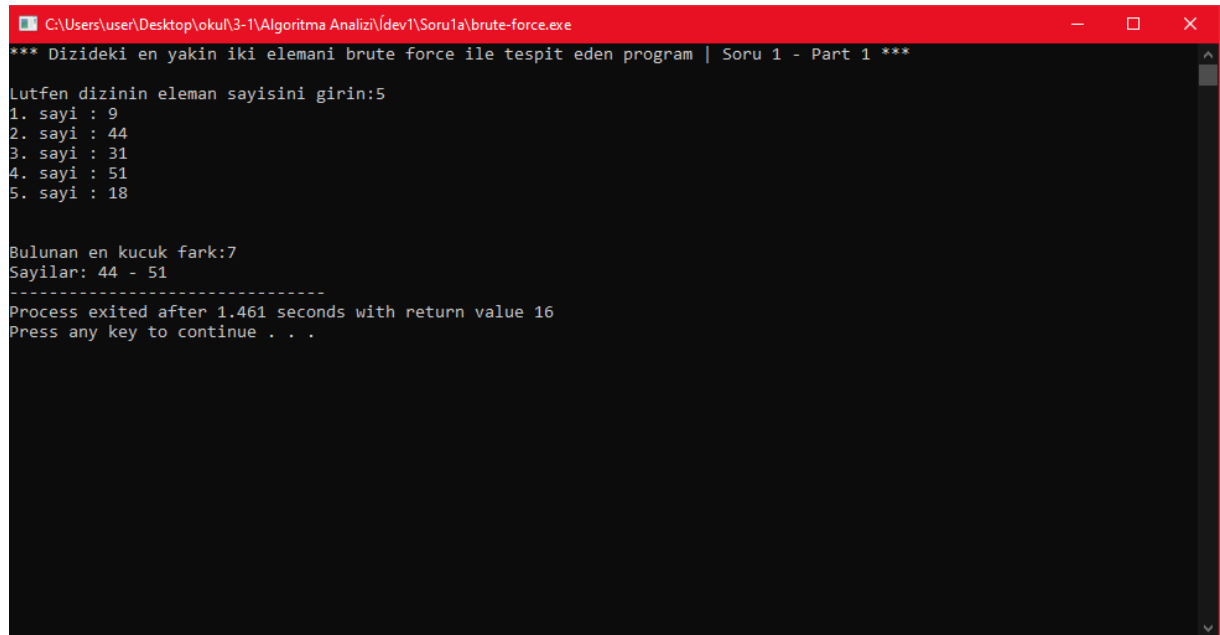
Ekran Çıktıları:

Ekran çıktısı 1:

Girdi: İstenilen eleman sayısı=5

Çıktı: En küçük fark=7

Sayılar : 44 – 51



```
C:\Users\user\Desktop\okul\3-1\Algoritma Analizi\dev1\Soru1a\brute-force.exe
*** Dizideki en yakın iki elemanı brute force ile tespit eden program | Soru 1 - Part 1 ***
Lutfen dizinin eleman sayisini girin:5
1. sayi : 9
2. sayi : 44
3. sayi : 31
4. sayi : 51
5. sayi : 18

Bulunan en kucuk fark:7
Sayilar: 44 - 51
-----
Process exited after 1.461 seconds with return value 16
Press any key to continue . . .
```

Ekran çıktısı 2:

Girdi: İstenilen eleman sayısı=10

Çıktı: En küçük fark=0

Sayılar : 10 – 10

```
C:\Users\user\Desktop\oku\3-1\Algoritma Analizi\dev1\Soru1a\brute-force.exe
*** Dizideki en yakın iki elemanı brute force ile tespit eden program | Soru 1 - Part 1 ***

Lutfen dizinin eleman sayisini girin:10
1. sayi : 90
2. sayi : 10
3. sayi : 83
4. sayi : 10
5. sayi : 38
6. sayi : 77
7. sayi : 30
8. sayi : 51
9. sayi : 55
10. sayi : 62

Bulunan en kucuk fark:0
Sayilar: 10 - 10
-----
Process exited after 1.333 seconds with return value 16
Press any key to continue . . .
```

Ekran çıktısı 3:

Girdi: İstenilen eleman sayısı=30

Çıktı: En küçük fark=0

Sayılar : 61 – 61

```
C:\Users\user\Desktop\okul\3-1\Algoritma Analizi\dev1\Soru1a\brute-force.exe
*** Dizideki en yakın iki elemanı brute force ile tespit eden program | Soru 1 - Part 1 ***

Lutfen dizinin eleman sayısını girin:30
1. sayi : 52
2. sayi : 22
3. sayi : 23
4. sayi : 61
5. sayi : 7
6. sayi : 81
7. sayi : 31
8. sayi : 69
9. sayi : 96
10. sayi : 21
11. sayi : 49
12. sayi : 70
13. sayi : 4
14. sayi : 61
15. sayi : 81
16. sayi : 91
17. sayi : 73
18. sayi : 79
19. sayi : 86
20. sayi : 38
21. sayi : 25
22. sayi : 89
23. sayi : 24
24. sayi : 24
25. sayi : 85
26. sayi : 44
27. sayi : 0
28. sayi : 96
29. sayi : 67
30. sayi : 45

Bulunan en küçük fark:0
Sayılar: 61 - 61
-----
Process exited after 1.565 seconds with return value 16
Press any key to continue . . .
```

Kodlar ve Tanımlar:

Kullanılan değişkenler:

int elemanSayisi;//Dizinin kaç elemanlı olacağı kullanıcı tarafından girilecektir.

int indexler [2];//Bulunan en yakın iki elemanın index numaralarını tutacak olan dizi

int enKucukFark=100;//Bulunan en küçük farkı tutmak için kullanılacak değişken

int geciciFark=0;//Sayılar arasındaki farkları geçici olarak saklamak için kullanılacak değişken

int dizi[elemanSayisi];//Elemanların tutulduğu dizi, Elemanlar 1 ile 100 arası olacaktır

int i =0;//Döngülerde kullanılacak olan değişken

int j =0;//Döngülerde kullanılacak olan değişken

Kod:

```
/*
```

```
06.11.2020
```

```
Burak Boz
```

```
18011706
```

Ödev 1, Soru 1, Part 1

Brute Force yaklaşımı ile dizideki en yakın iki elemanı tespit etme

```
*/
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
void main()
```

```
{
```

```
    srand(time(NULL)); //Random methodu kullanıldığında farklı sayılar üretmek için kullanılır
```

```
    printf("*** Dizideki en yakın iki elemanı brute force ile tespit eden program | Soru 1 - Part 1 ***\n\n");
```

```
    int elemanSayisi;//Dizinin kaç elemanlı olacağı kullanıcı tarafından girilecektir
```

```
    int indexler [2];//Bulunan en yakın iki elemanın index numaralarını tutacak olan dizi
```

```
    int enKucukFark=100;//Bulunan en küçük farkı tutmak için kullanılacak değişken
```

```
    int geciciFark=0;//Sayılar arasındaki farkları geçici olarak saklamak için kullanılacak değişken
```

```
    printf("Lutfen dizinin eleman sayisini girin:");
```

```
    scanf("%d",&elemanSayisi);
```

```
    int dizi[elemanSayisi];//Elemanların tutulduğu dizi, Elemanlar 1 ile 100 arası olacaktır
```

```
    int i =0;//Döngülerde kullanılacak olan değişken
```

```
    int j =0;//Döngülerde kullanılacak olan değişken
```

```
    for(i = 0;i<elemanSayisi;i++)//Diziyi rastgele sayılarla doldurduk
```

```
    {
```

```
        dizi[i]=rand()%100;
```

```
    }
```

```
    for(i = 0;i<elemanSayisi;i++)//Diziyi yazdırma
```

```
    {
```

```
        printf("%d. sayi : %d\n",i+1,dizi[i]);
```

```
}

printf("\n\n");
for(i=0;i<elemanSayisi;i++)
{
    for(j=0;j<elemanSayisi;j++)
    {
        if(i!=j)
        {
            geciciFark=abs(dizi[i]-dizi[j]);
            if(geciciFark<enKucukFark)
            {
                enKucukFark=geciciFark;
                indexler[0]=dizi[i];
                indexler[1]=dizi[j];
            }
        }
    }
}

printf("Bulunan en kucuk fark:%d\n",enKucukFark);
printf("Sayilar: %d - %d",indexler[0],indexler[1]);
}
```

Problem: b. Problemi brute-force yaklaşımdan daha az karmaşıklığa sahip başka bir algoritmik yöntem ile çözüp analiz ederek verimliliğini ortaya koyunuz. Yeni yaklaşımınızı C dilinde kodlayınız.

Çözüm: Dizi ile ilgili işlemler yaparken kullandığımız dizinin sıralı olduğunu bilmek her zaman bize fayda sağlayacaktır. A Şıkında dizi sırasız olduğu için brute force yönelimiyle tüm elemanları teker teker karşılaştırmak zorunda kaldık ve karmaşıklığımız n^2 oldu. Eğer dizimizi daha kolay şekilde sıralayabilirsek bu durum algoritmamızın daha düzenli çalışmasını sağlayacaktır.

Dizimizi $n(\log n)$ karmaşıklığına sahip olan merge sort sıralama yöntemiyle sıraladıktan sonra sayılar arasındaki farkları karşılaştırmak artık bizim için daha kolay olacak. Tüm sayıları teker teker diğer sayılarla kontrol etmektense, her sayıyı bir sonraki sayı ile kontrol etmek bizim için daha kolay olacak ve tek döngü ile en küçük farkı bulabileceğiz.

Karmaşıklık: Merge Sort + Tek döngü ile farkın bulunması

$$\begin{aligned} n(\log(n)) + n &= n(1 + \log(n)) \\ &= n(\log(n)) \end{aligned}$$

Ekran Çıktıları:

Ekran çıktısı 1:

Girdi: İstenilen eleman sayısı=5

Çıktı: En küçük fark=4

Sayılar : 86 – 90

```
C:\Users\user\Desktop\oku\3-1\Algoritma Analiz\dev1\Soru1b\EnKucukFark.exe
*** Dizideki en yakın iki elemanı tespit eden program | Soru 1 - Part 2 ***

Lutfen dizinin eleman sayisini girin. Elemanlar 1 ile 100 arasında rastgele sayılardan olusturulacaktır:5
1. sayi : 35
2. sayi : 60
3. sayi : 86
4. sayi : 90
5. sayi : 2

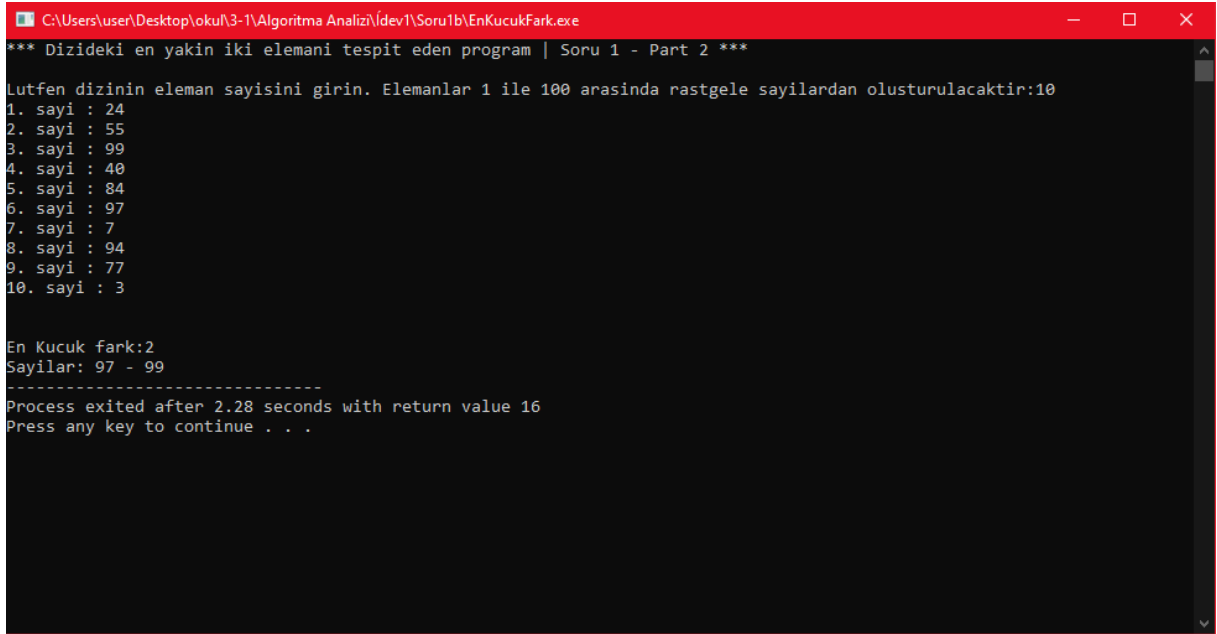
En Kucuk fark:4
Sayilar: 86 - 90
-----
Process exited after 1.572 seconds with return value 16
Press any key to continue . . .
```

Ekran ıktısı 2:

Girdi: İstenilen eleman sayısı=10

Çıktı: En küçük fark=2

Sayılar : 97 – 99



```
C:\Users\user\Desktop\oku\3-1\Algoritma Analizi\dev1\Soru1b\EnKucukFark.exe
*** Dizideki en yakın iki elemanı tespit eden program | Soru 1 - Part 2 ***

Lutfen dizinin eleman sayisini girin. Elemanlar 1 ile 100 arasinda rastgele sayilardan olusturulacaktır:10
1. sayi : 24
2. sayi : 55
3. sayi : 99
4. sayi : 40
5. sayi : 84
6. sayi : 97
7. sayi : 7
8. sayi : 94
9. sayi : 77
10. sayi : 3

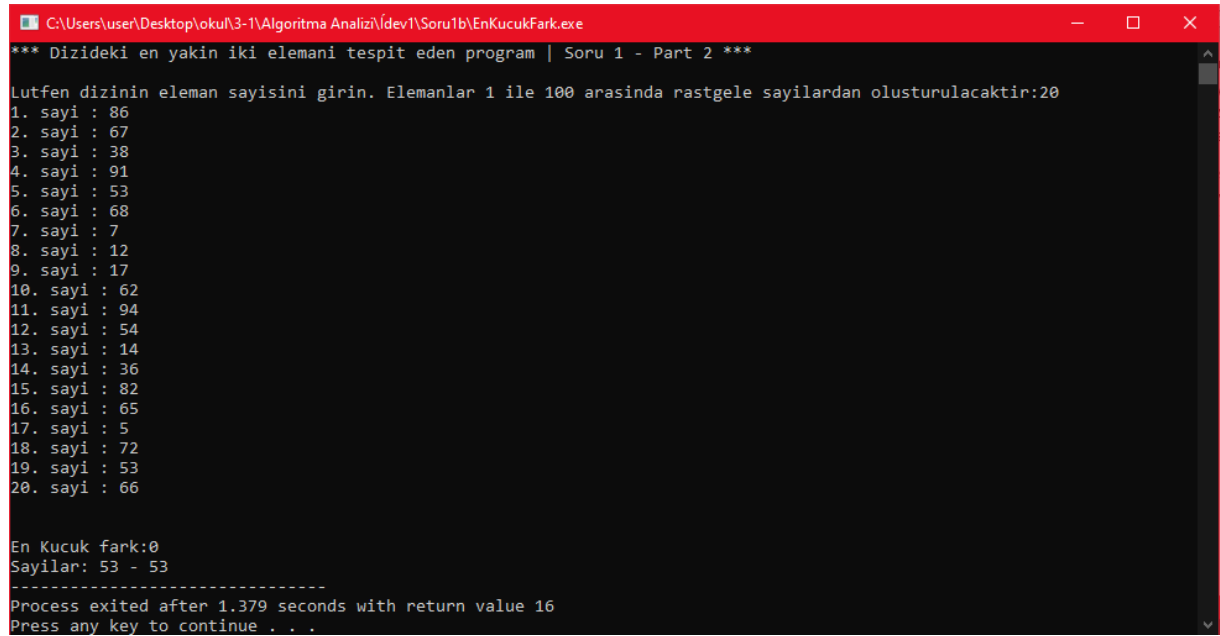
En Kucuk fark:2
Sayilar: 97 - 99
-----
Process exited after 2.28 seconds with return value 16
Press any key to continue . . .
```


Ekran çıktısı 3:

Girdi: İstenilen eleman sayısı=20

Çıktı: En küçük fark=0

Sayılar : 53 - 53



```
C:\Users\user\Desktop\oku\3-1\Algoritma Analizi\dev1\Soru1b\EnKucukFark.exe
*** Dizideki en yakın iki elemanı tespit eden program | Soru 1 - Part 2 ***

Lutfen dizinin eleman sayisini girin. Elemanlar 1 ile 100 arasında rastgele sayılardan olusturulacaktır:20
1. sayi : 86
2. sayi : 67
3. sayi : 38
4. sayi : 91
5. sayi : 53
6. sayi : 68
7. sayi : 7
8. sayi : 12
9. sayi : 17
10. sayi : 62
11. sayi : 94
12. sayi : 54
13. sayi : 14
14. sayi : 36
15. sayi : 82
16. sayi : 65
17. sayi : 5
18. sayi : 72
19. sayi : 53
20. sayi : 66

En Kucuk fark:0
Sayilar: 53 - 53
-----
Process exited after 1.379 seconds with return value 16
Press any key to continue . . .
```

Kodlar ve Tanımlar:

Kullanılan değişkenler:

```
int elemanSayisi;//Dizinin kaç elemanlı olacağı kullanıcı tarafından girilecektir
int indexler [2];//Bulunan en yakın iki elemanın index numaralarını tutacak olan dizi
int enKucukFark=100;//Bulunan en küçük farkı tutmak için kullanılacak değişken
int geciciFark=0;//Sayılar arasındaki farkları geçici olarak saklamak için kullanılacak değişken
int dizi[elemanSayisi];//Elemanların tutulduğu dizi, Elemanlar 1 ile 100 arası olacaktır
int i =0;//Döngülerde kullanılacak olan değişken
int j =0;//Döngülerde kullanılacak olan değişken
```

Kullanılan fonksiyonlar:

```
void merge(int arr[], int l, int m, int r)
```

```
/*
```

Fonksiyonun aldığı değerler(Sırası ile): Dizinin kendisi,sol değer,orta değer,sağ değer

Fonksiyon diziyi bölmeye ve tekrar birleştirmeye yarar

```
*/
```

```
void mergeSort(int arr[], int l, int r) //Dizinin sıralanması için kullanılan fonksiyon
```

```
/*
```

Fonksiyonun aldığı değerler(Sırası ile):Dizinin kendisi, sol değer, sağ değer

Dizinin sıralanmasını sağlayan özyinelemeli fonksiyondur. Bu fonksiyon tamamlandıktan sonra dizi sıralanmış olacaktır.

```
*/
```

Kod:

```
/*
```

06.11.2020

Burak Boz

18011706

Ödev 1, Soru 1, Part 2

Dizideki en yakın iki elemanı tespit etme

Bu programda dizi önce merge sort ile sıralanacak ve sıralı ikili elemanlar kontrol edilerek en küçük fark bulunacak

```
*/
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <time.h>
```

```
void merge(int arr[], int l, int m, int r)
```

```
{
```

```
    int i, j, k;
```

```
    int n1 = m - l + 1;
```

```
    int n2 = r - m;
```

```
    /* Kullanılacak geçici değişkenler */
```

```
    int L[n1], R[n2];
```

```
    /* Sol ve sağ değerler geçici değişkenlere kopyalanıyor */
```

```

for (i = 0; i < n1; i++)
    L[i] = arr[l + i];
for (j = 0; j < n2; j++)
    R[j] = arr[m + 1+ j];

/* Geçici değişkenler dizi ile birleştiriliyor*/
i = 0; // altdizinin ilk değerinin ayarlanması için
j = 0; // altdizinin ikinci değerinin ayarlanması için
k = l; // Birleştirilecek dizi için
while (i < n1 && j < n2)
{
    if (L[i] <= R[j])
    {
        arr[k] = L[i];
        i++;
    }
    else
    {
        arr[k] = R[j];
        j++;
    }
    k++;
}

//Sol dizide eleman kaldıysa diziye kopyalanması
while (i < n1)
{
    arr[k] = L[i];
    i++;
    k++;
}

/* Sağ dizide eleman kaldıysa diziye kopyalanması*/
while (j < n2)
{
    arr[k] = R[j];
    j++;
    k++;
}

```

```

    }
}

void mergeSort(int arr[], int l, int r) //Dizinin sıralanması için kullanılan fonksiyon
{
    if (l < r)
    {
        int m = l+(r-l)/2;

        mergeSort(arr, l, m);

        mergeSort(arr, m+1, r);

        merge(arr, l, m, r);
    }
}

void main()
{
    srand(time(NULL)); //Random methodu kullanıldığında farklı sayılar üretmek için kullanılır

    printf("*** Dizideki en yakın iki elemanı tespit eden program | Soru 1 - Part 2 ***\n\n");

    int elemanSayisi; //Dizinin kaç elemanlı olacağı kullanıcı tarafından girilecektir

    int indexler [2]; //Bulunan en yakın iki elemanın index numaralarını tutacak olan dizi

    int enKucukFark=100; //Bulunan en küçük farkı tutmak için kullanılacak değişken

    int geciciFark=0; //Sayılar arasındaki farkları geçici olarak saklamak için kullanılacak değişken

    printf("Lutfen dizinin eleman sayisini girin. Elemanlar 1 ile 100 arasinda rastgele sayılardan olusturulacaktır:");

    scanf("%d",&elemanSayisi);

    int dizi[elemanSayisi]; //Elemanların tutulduğu dizi, Elemanlar 1 ile 100 arası olacaktır

    int i =0; //Döngülerde kullanılacak olan değişken

    int j =0; //Döngülerde kullanılacak olan değişken

    for(i = 0; i<elemanSayisi; i++) //Diziyi rastgele sayılarla doldurduk
    {
        dizi[i]=rand()%100;
    }

    for(i = 0; i<elemanSayisi; i++) //Diziyi yazdırma
    {
        printf("%d. sayi : %d\n", i+1, dizi[i]);
    }

    printf("\n\n");

    mergeSort(dizi, 0, elemanSayisi - 1); //Dizi sıralanıyor

```

```
for(i=0;i<elemanSayisi-1;i++)
{
    geciciFark=abs(dizi[i]-dizi[i+1]);
    if(geciciFark<enKucukFark)
    {
        enKucukFark=geciciFark;
        indexler[0]=dizi[i];
        indexler[1]=dizi[i+1];
    }
}

printf("En Kucuk fark:%d\n",enKucukFark);
printf("Sayilar: %d - %d",indexler[0],indexler[1]);
}
```