YILDIZ TEKNİK ÜNİVERSİTESİ

ELEKTRİK-ELEKTRONİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BLM3021 – ALGORİTMA ANALİZİ

1.ÖDEV 1.SORU RAPORU

Konu : N elemanlı bir dizide birbirine en yakın değere sahip iki elemanın bulunması

Ad Soyad : Eren TUTUŞ

Okul No : 17011702

E-posta : l1117702@std.yildiz.edu.tr

Soru 1-A İçin Programın Çalışma Süreci

1-) *enYakinDegerlerBul (int dizi[ ] , int n) Fonksiyonu :

Brute Force ile en yakın 2 değeri bulurken bu fonksiyona öncelikle dizimizi ve dizimizin eleman sayısını parametre olarak yolluyoruz. Daha sonra gelen eleman değeri 2 den küçük ise program sonlandırılacak bunun nedeni en yakın 2 değer bulmamız istenirse dizi en az 2 elemana sahip olmalıdır. Eğer parametre olarak gelen dizinin eleman sayısı 2 veya daha fazla ise en yakın iki değeri bulmaya başlayabiliriz.

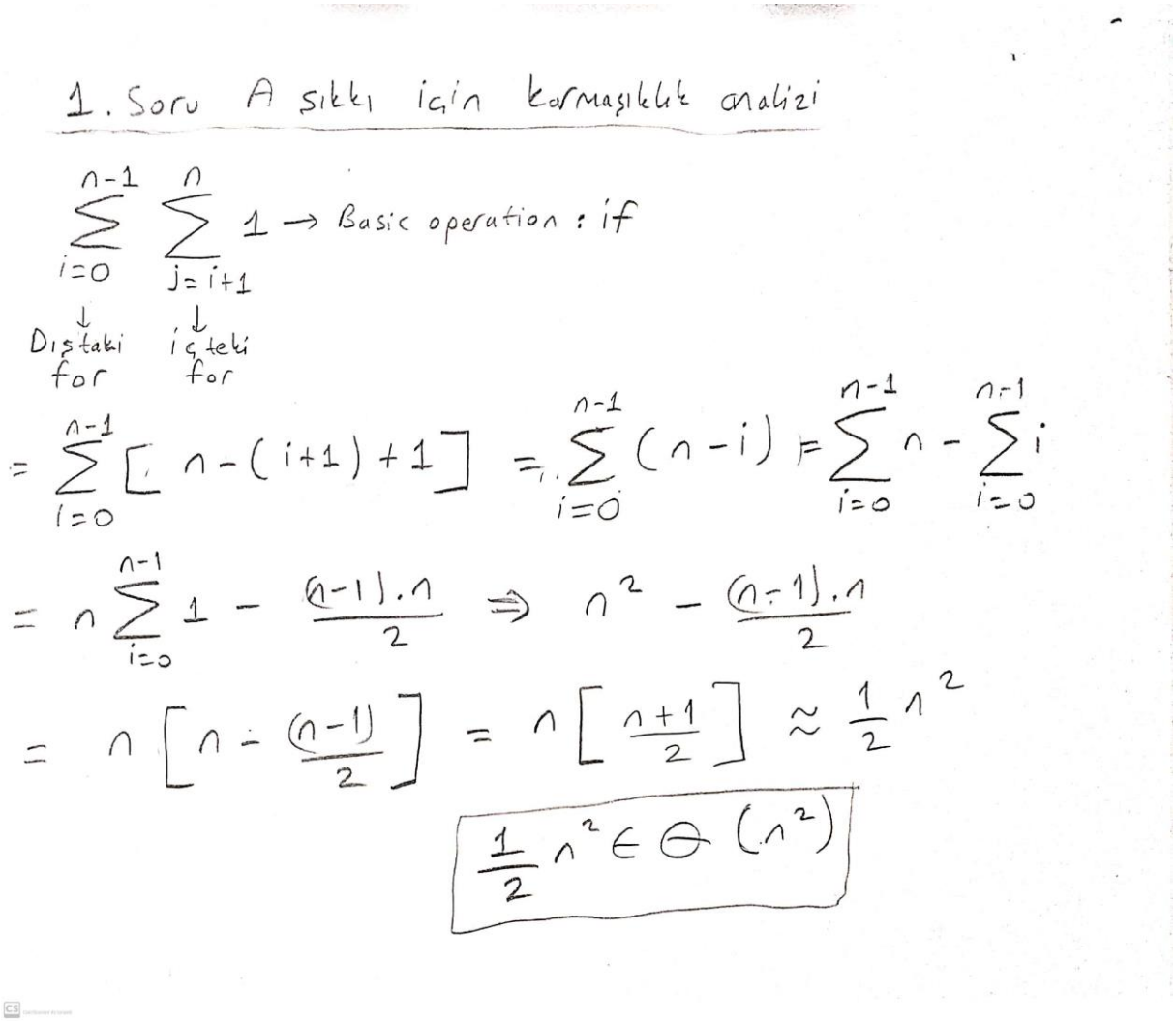
Öncelikle oluşturduğumuz enYakinFark değişkenine başlangıç olarak dizinin ilk 2 elemanı arasındaki mutlak değerce farkı atıyoruz. Ayrıca, *enYakinDegerDizi ismiyle oluşturduğumuz diziye bellekte 2 değerlik yer açıyoruz daha sonra buraya en yakın 2 değeri aktarmış olacağız.

Çift döngü içinde yaptığımız if sorgusunda fonksiyona parametre olarak gelen dizimizin i. Ve j(i+1). elemanlarının mutlak değerce farkı enYakinFark değişkenindeki değerden küçük ise enYakinFark değişkeni güncellenecek ve yeni indislerdeki sayıların mutlak değerce farkı olacak. Ayrıca, en yakın iki değeri enYakinDegerDizi dizimize atıp, dizimizdeki en yakın iki değeride güncellemiş olduk. Döngüler bittikten sonra ise enYakinDegerDizi dizimize gelen en yakın iki değeride return ile döndürmüş olduk.

2-) main( ) Fonksiyonu :

Main fonksiyonunda öncelikle kullanıcıdan dizinin eleman sayısını istedik ve gireceği eleman sayısı kadarda diziye eleman eklemesini istedik. Daha sonra elemanlar girildikten sonra diziyi ekrana yazdırdık ve bunun hemen altında ise enYakinDegerlerBul fonksiyonundan gelen diziyi çekip ekrana yazdırdık. Böylelikle kullanıcı oluşturduğu diziyi ve bu dizideki birbirine değerce en yakın olan 2 değeri görmüş oldu.

Karmaşıklık Analizi

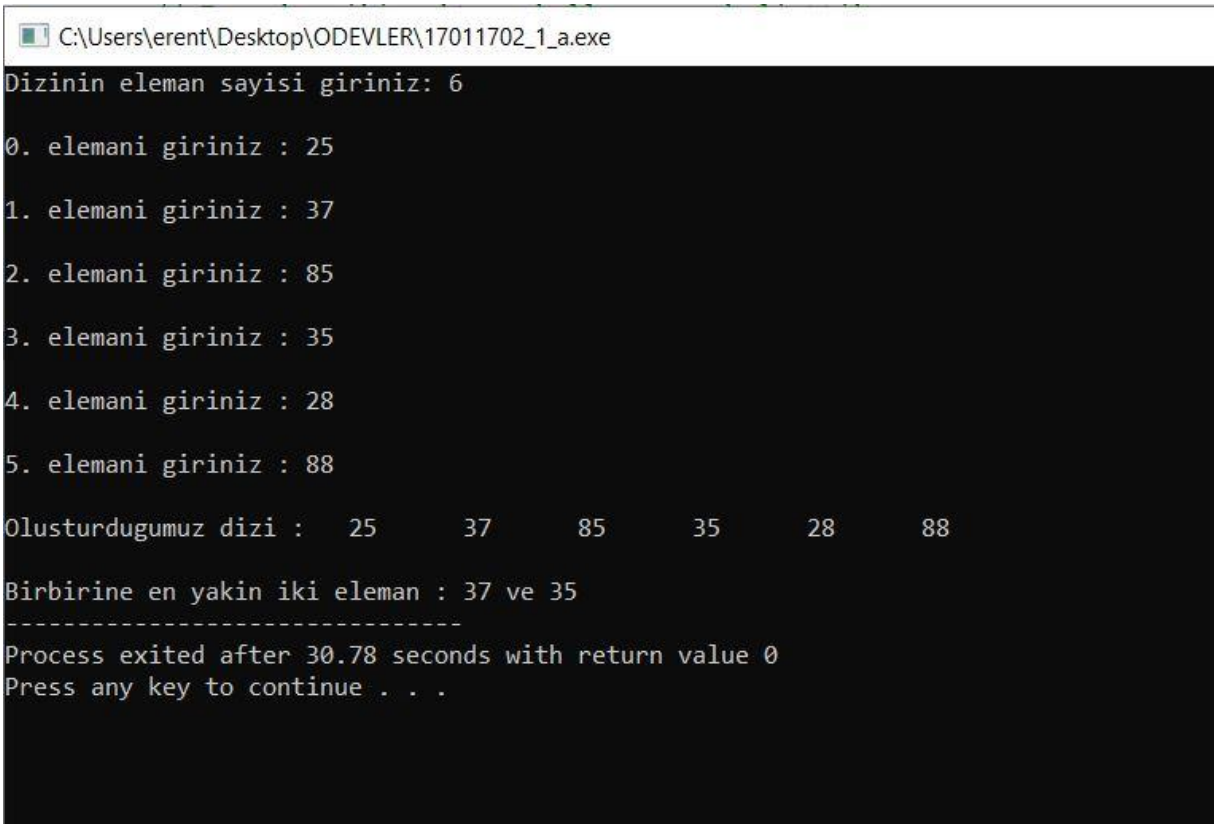


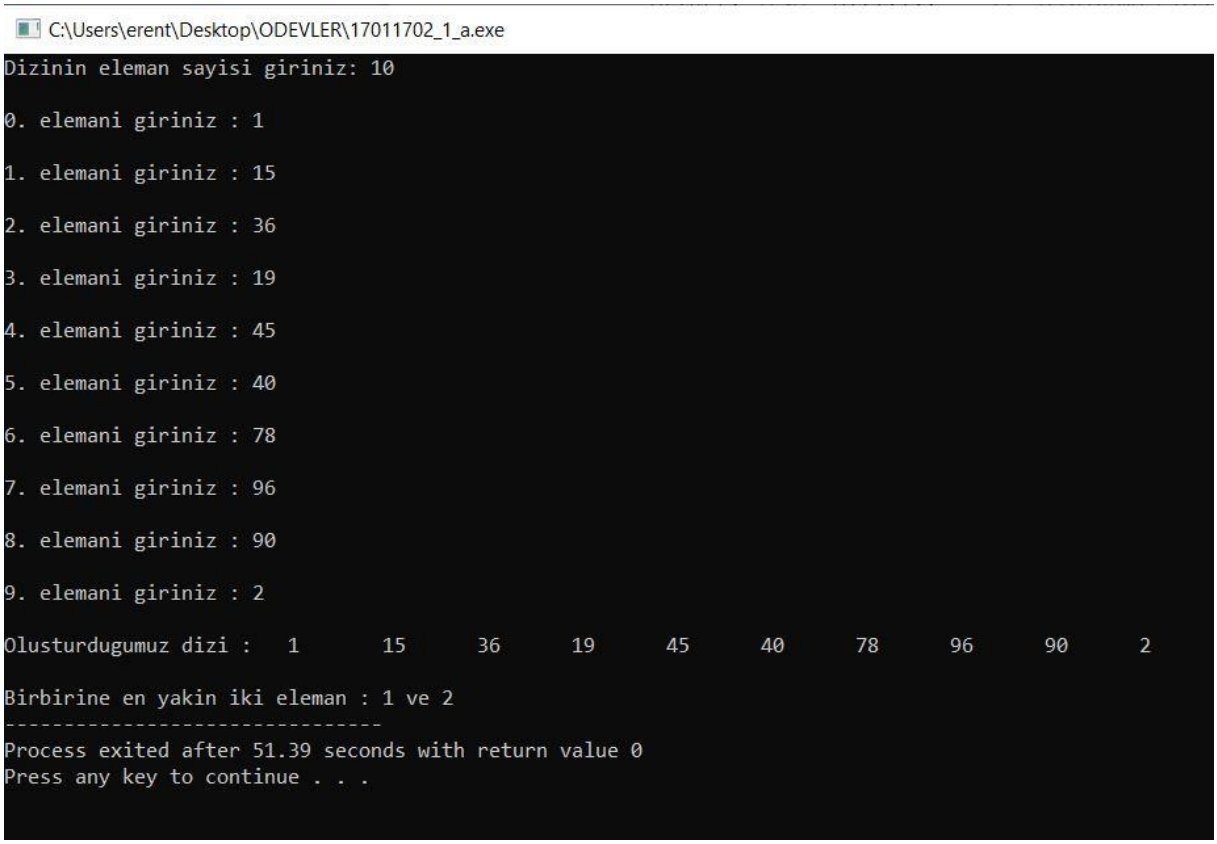
C KODU

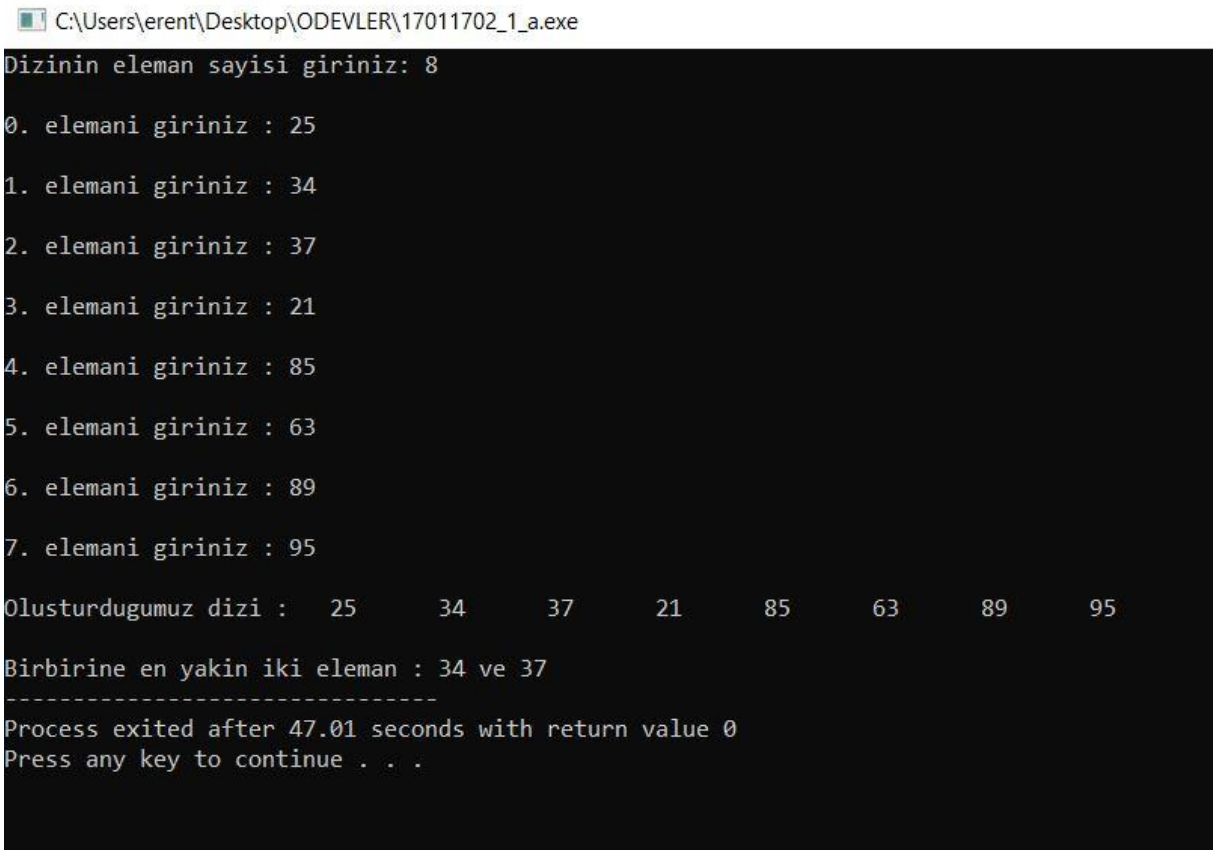
```
#include <stdio.h>
#include <stdlib.h>

int *enYakinDegerlerBul (int dizi[],int n){  // Olusturdugumuz dizi ve onun
eleman sayisini parametre olarak aliyoruz.
    if (n<2){          //  En yakin 2 nokta bulmamız gerektigi icin minimum
2 elemanli bir dizi olmali !
        printf("\nEn yakin 2 deger icin bu dizi en az 2 degere sahip olmali
!");
        return;
    }
    else {           // Kod, else bloguna girdiginde en az 2 elemanli bir
diziye sahip oldugumuzu anliyoruz.
        int i;  // Dış döngü için sayaç değişkeni
```

```
        int j;  // İç döngü için sayaç değişkeni
        int enYakinFark = abs(dizi[0]-dizi[1]);  // Başlangıç olarak
fonksiyona parametre olarak gelen dizinin ilk 2 elemanı arasındaki mutlak
değerce farkı enYakinFark değişkenine atıyoruz.
        int *enYakinDegerDizi = (int*)calloc(2,sizeof(int));  //
Bulacağımız en yakın 2 değerin tutulacağı dizi
        for (i=0;i<n-1;i++){
            for(j=i+1;j<n;j++){
                if(abs(dizi[i] - dizi[j]) <= enYakinFark){   // Dizinin i.
ve j(i+1). elemanlarının mutlak değerce farkı enYakinFark'tan küçükse;
                    enYakinFark = abs(dizi[i] - dizi[j]);   //  enYakinFark
değişkeni güncellenecek ve yeni indislerdeki sayıların mutlak değerce farkı
olacak.
                    enYakinDegerDizi [0] = dizi[i];  // Ayrıca, en yakın
iki değeri enYakinDegerDizi dizimize atıp,
                    enYakinDegerDizi [1] = dizi[j];  // dizimizdeki en
yakın iki değeri güncellemiş olduk.
                }
            }
        }
        return enYakinDegerDizi;        // Son olarak en yakın iki noktayı
döndürdük.
    }
}

int main(){

    int n;      // Dizinin eleman sayısı
    int i;      // Diziye eleman atmak için kullanacağımız sayaç değişkeni
    printf("Dizinin eleman sayisi giriniz: ");
    scanf("%d",&n);
    int dizi[n]; // Kullanıcıdan n elemanlı bir dizi oluşturmasını istedik.

    for(i=0;i<n;i++){
        printf("\n%d. elemani giriniz : ",i);
        scanf("%d",&dizi[i]);          // Dizimize değerleri girdik.
    }

    printf("\nOlusturdugumuz dizi : ");
    for(i=0;i<n;i++){
        printf("\t%d",dizi[i]);    // Oluşturan dizimizi ekrana yazdırdık.
    }

    // ve son olarak kullanıcıya en yakın 2 noktayı belirttik.
    int *enYakinDegerDizi = enYakinDegerlerBul (dizi,n);
    printf("\n\nBirbirine en yakin iki eleman : %d ve %d",enYakinDegerDizi
[0],enYakinDegerDizi [1]);

    return 0;
}
```

EKRAN ÇIKTILARI







Soru 1-B İçin Programın Çalışma Süreci

1-) mergeSort (int dizi [ ] , int firstIndex , int lastIndex) Fonksiyonu :

Main fonksiyonunda oluşturduğumuz diziyi, o dizinin ilk ve son elemanlarını parametre olarak aldık. MergeSort'da diziyi sürekli ortadan ikiye rekursif olarak ayırdığımız için her seferinde dizinin orta noktasını alıyoruz. İlk eleman ile orta noktaya kadar olan bölümü rekursif olarak kendi içerisinde her defasında ortadan ikiye ayırıp tek parça haline getiriyoruz.

Orta nokta + 1 ' den başlayarak son elemana kadar olan bölümü rekursif olarak kendi içerisinde her defasında ortadan ikiye ayırıp tek parça haline getiriyoruz. Daha sonra tek parça halinde kalanları birleştir fonksiyonu ile yani merge mantığıyla birleştiririz.

2-) void birlestir(int dizi [ ],int firstIndex, int middleIndex, int lastIndex) Fonksiyonu:

Birlestir fonksiyonuna gelen parametreleri biz mergeSort'dan aldık ve burada birleştirme parçalara ayrılmış değerleri yeniden sıralı olarak birleştireceğiz. İlk alt dizimizi oluşturup buna ilk baştan orta noktaya kadar olan kısmın eleman sayısını aktardık ve aynı işlemi ikinci alt dizi için orta nokta + 1 'den son elemana kadar olan kısmın eleman sayısını aktardık.

Daha sonra orta noktaya kadar olan tüm değerleri firstSubArray dizisine aktardık ve aynı işlemi secondSubArray dizisinede gerçekledik. Yani orta nokta + 1 'den son elemana kadar olan değerleri aktardık. Daha sonra her iki alt dizinin başlangıç index'inin kendi eleman sayılarından küçük olduğu sorguyu yaparız. Eğer ilk alt dizinin başlangıç değeri ikinci alt dizinin başlangıç değerinden küçük ise dizinin ilk elemanı olur. Aksi halde ise ikinci alt dizinin ilk elemanı, büyük dizinin elemanı olur. Bundan sonraki işlemlerde ise eğer varsa kalan firstSubArray elemanlarını aktarırız. Aynı işlemi secondSubArray içinde gerçekliyoruz.

3-) enYakinDegerlerBul (int dizi [ ] , int n) Fonksiyonu :

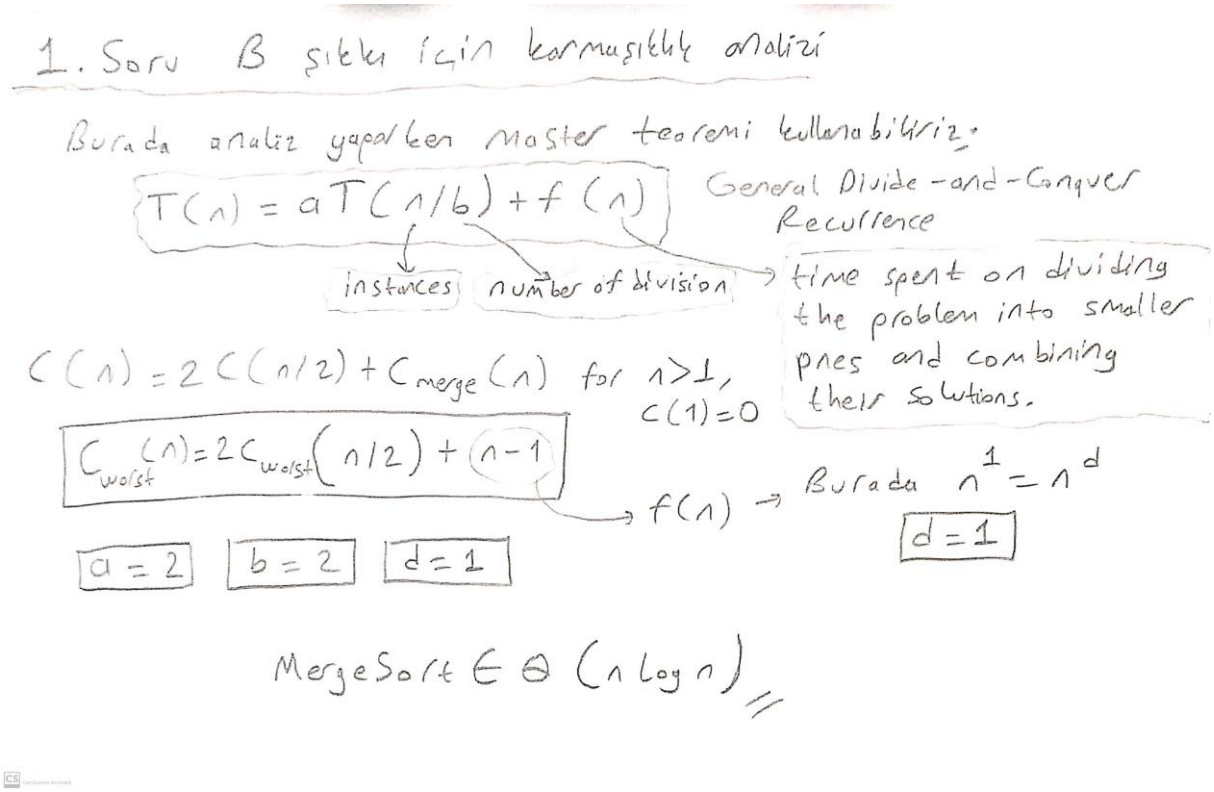
Bu fonksiyona gelen parametreler main fonksiyonunda mergeSort ile sıralamış olduğumuz dizi ve dizinin eleman sayısıdır. Bu fonksiyondaki enYakinFark değişkeni en yakın iki değer arasındaki mutlak değerce farkı tutacağımız değişkendir. Oluşturduğumuz *enYakinDegerDizi dizisi ise en yakın iki noktayı içinde tutacağımız dizidir. Başlangıç olarak dizinin ilk ve ikinci indislerinin mutlak değerce farkını alıp bu değeri enYakinFark değişkenine aktardık. Ayrıca, *enYakinDegerDizi dizisi için hafızada 2 değerlik yer açıyoruz ve buraya ileri ki işlemlerimizde en yakın iki değeri atıyor olacağız.

Eğer bir önceki yakın 2 değerin mutlak değerce farkı yeni sorguladığımız sıralı iki değerin mutlak değerce farkından büyük ise; enYakinFark değişkenimizi yeni gelecek iki değerin mutlak değerce farkıyla güncelleştiriyoruz. Ayrıca, en yakın iki değeride enYakinDegerDizi dizisine aktarıyoruz böylelikle enYakinDegerDizi dizisinide güncellemiş oluyoruz. Tüm işlemler bittikten sonra en yakın iki değerin olduğu enYakinDegerDizi dizisini döndürüyoruz.

4-) main ( ) Fonksiyonu :

Kullanıcıdan oluşturacağımız dizi için eleman sayısı istiyoruz ve bu eleman sayısı kadarda değer girmesini istiyoruz. Oluşturulan dizi ekrana yazdırılır ve daha sonra hemen altında mergeSort'a gönderdiğimiz parametrelere göre yapılan işlemler sonucunda sıralanmış olan diziyi ekrana yazdırırız. Bu sıralanmış diziyi enYakinDegerlerBul fonksiyonumuza yollayarak buradaki işlemler sonucunda bize dönen en yakın 2 değeri kullanıcının görmesi için ekrana yazdırıyoruz.

Karmaşıklık Analizi



C KODU

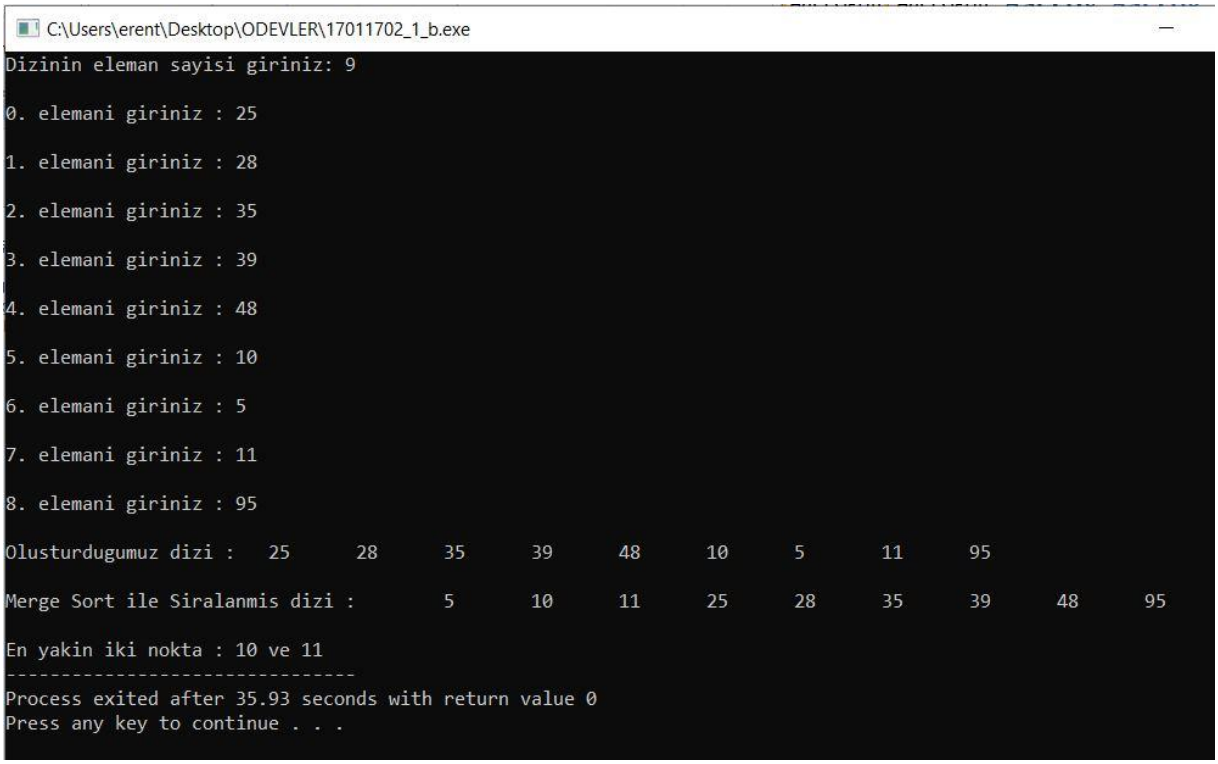
```c
#include <stdio.h>
#include <stdlib.h>


void birlestir(int dizi[],int firstIndex,int middleIndex,int lastIndex){
// mergeSort fonksiyonundan gelen dizi,firstIndex,middleIndex ve lastIndex
parametreleri

    int firstSubArrayCount;  // İlk alt dizinin eleman sayısı
    firstSubArrayCount = middleIndex - firstIndex + 1;
    int secondSubArrayCount;     // İkinci alt dizinin eleman sayısı
    secondSubArrayCount = lastIndex - middleIndex;
    int i;  // Döngü için sayaç değişkeni
    int firstSubArray[firstSubArrayCount];   // İlk alt dizimizi
oluşturduk. --> firstSubArray [firstIndex...middleIndex]
    int secondSubArray[secondSubArrayCount]; // İkinci alt dizimizi
oluşturduk. --> secondSubArray [firstIndex...middleIndex]

    for(i=0;i<firstSubArrayCount;i++){
        firstSubArray[i] = dizi[firstIndex+i];   // Orta noktaya kadar olan
değerleri firstSubArray dizisine aktarıyoruz.
    }

    for(i=0;i<secondSubArrayCount;i++){
        secondSubArray[i] = dizi[middleIndex+i+1];   // Orta nokta + 1 'den
son elemana kadar olan değerleri ise secondSubArray dizisine aktarıyoruz.
    }


    int first;  // Birleştirilmiş alt dizinin başlangıç indeksi.
    first = firstIndex;
    int a = 0;  // İlk alt dizinin başlangıç indeksi.
    int b = 0;  // İkinci alt dizinin başlangıç indeksi.

    while(a<firstSubArrayCount && b<secondSubArrayCount){   // Her iki alt
dizinin başlangıç index'inin kendi eleman sayılarından küçük olduğu sorguyu
yaparız.
        if(firstSubArray[a]<=secondSubArray[b]){   // Eğer ilk alt dizinin
başlangıç değeri, ikinci alt dizinin başlangıç değerinden küçük ise dizinin
ilk elemanı olur.
            dizi[first]=firstSubArray[a];
            a++;
        }
        else{
            dizi[first]=secondSubArray[b];   // Aksi halde ise ikinci alt
dizinin ilk elemanı, büyük dizinin ilk elemanı olur.
            b++;
        }
        first++;
    }

    while (a < firstSubArrayCount)   // Eğer varsa kalan firstSubArray
elemanlarını aktarırız.
    {
        dizi[first] = firstSubArray[a];
        a++;
        first++;
    }
```

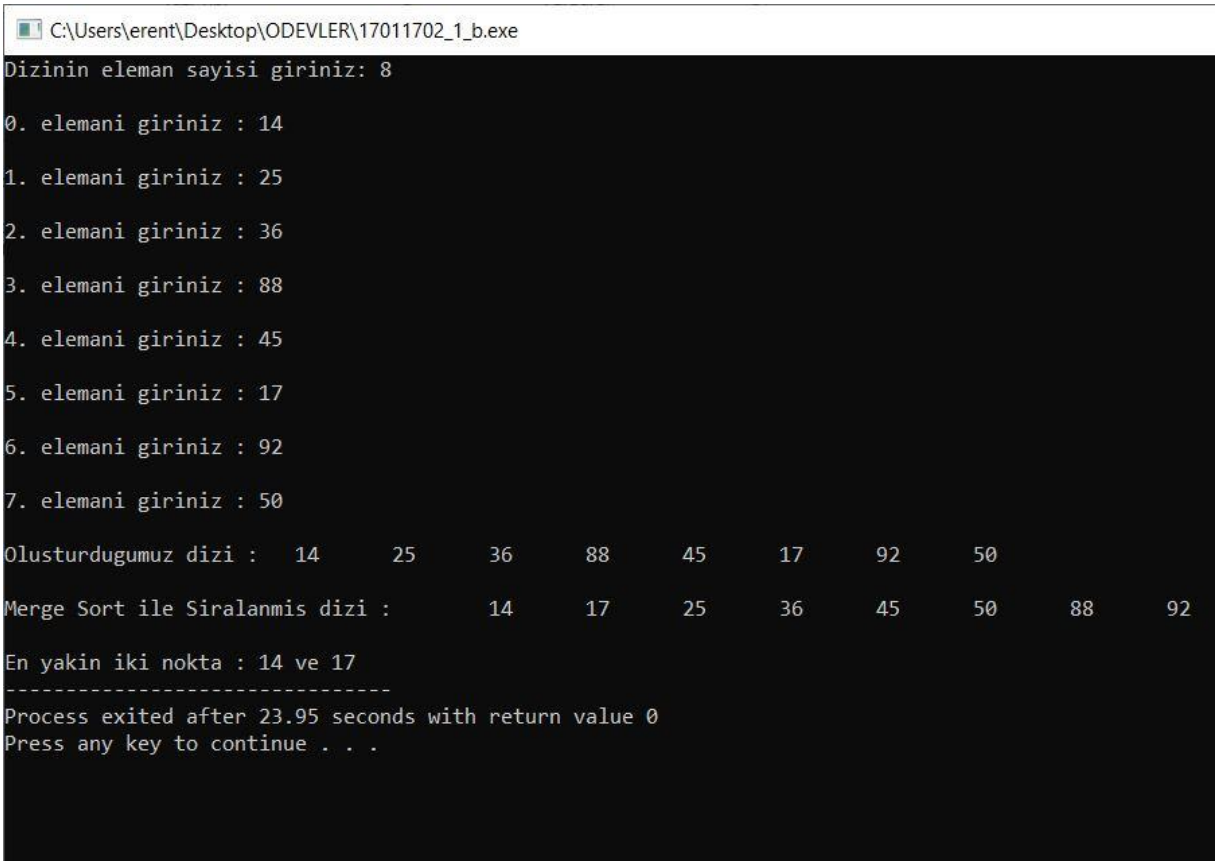
```
    while (b < secondSubArrayCount)  // Eğer varsa kalan secondSubArray
elemanlarını aktarırız.
    {
        dizi[first] = secondSubArray[b];
        b++;
        first++;
    }

}

void mergeSort(int dizi[],int firstIndex,int lastIndex){ // Oluşturduğumuz
diziyi, o dizinin ilk ve son elemanlarını parametre olarak aldık.
    if (lastIndex > firstIndex)
    {
        int middle; // Dizi için orta noktayı belirlediğimiz değişken
        middle = (firstIndex+lastIndex)/2; //  Merge Sort'ta diziyi sürekli
ortadan ikiye ayırdığımız için her seferinde dizinin orta noktasını
alıyoruz.
        mergeSort(dizi,firstIndex,middle); // İlk eleman ile orta noktaya
kadar olan bölümü rekursif olarak kendi içerisinde her defasında ortadan
ikiye ayırıp tek parça haline getiriyoruz.

        mergeSort(dizi,middle+1,lastIndex); // Orta nokta + 1 ' den
başlayarak son elemana kadar olan bölümü rekursif olarak kendi içerisinde
her defasında ortadan ikiye ayırıp tek parça haline getiriyoruz.

        birlestir(dizi,firstIndex,middle,lastIndex);  // Tek parça halinde
kalanları birleştir fonksiyonu ile yani merge mantığıyla birleşiririz.
    }

}


int *enYakinDegerlerBul (int dizi[],int n){   // parametre olarak Merge
Sort ile sıraladığımız diziyi ve dizinin elemanı sayısını yolladık.
    int i;  // Döngü sayaç değişkeni
    int enYakinFark;  // En yakın iki değer arasındaki mutlak değerce farkı
tutacağımız değişken
    int *enYakinDegerDizi = (int*)calloc(sizeof(int),2);  // En yakın iki
değeri tuttuğumuz dizi
    enYakinFark = abs(dizi[0]-dizi[1]);  // Başlangıç olarak dizinin ilk ve
ikinci indislerinin mutlak değerce farkını aldık. Bunu döngüde
kullanacağız.
    for(i=0;i<n;i++){
        if(enYakinFark > (abs(dizi[i]-dizi[i+1]))){    // Eğer bir önceki
en yakın 2 değerin mutlak değerce farkı yeni sorguladığımız sıralı iki
değerin mutlak değerce farkından büyük ise;
            enYakinFark = abs(dizi[i]-dizi[i+1]);   // enYakinFark
değişkenimizi yeni gelecek iki değerin mutlak değerce farkıyla
güncelleştiriyoruz.
            enYakinDegerDizi [0] = dizi[i];
            enYakinDegerDizi [1] = dizi[i+1];  // Ayrıca, En yakın iki
noktayıda enYakinDegerDizi dizisine aktarıyoruz böylelikle enYakinDegerDizi
dizisinide güncellemiş oluyoruz.
        }
    }
    return enYakinDegerDizi;   // Tüm işlemler bittikten sonra en yakın iki
değerin olduğu enYakinDegerDizi dizisini döndürüyoruz.
}
```

```c
int main(){

    int n;  // Dizinin eleman sayısı
    int i;  // Döngü için sayaç değişkeni
    printf("Dizinin eleman sayisi giriniz: ");
    scanf("%d",&n);
    int dizi[n];  // n elemanlı bir dizi oluşturduk.

    for(i=0;i<n;i++){
        printf("\n%d. elemani giriniz : ",i);
        scanf("%d",&dizi[i]);    // Kullanıcı diziye elemanları ekledi.
    }

    printf("\nOlusturdugumuz dizi : ");
    for(i=0;i<n;i++){
        printf("\t%d",dizi[i]);    // Oluşturduğumuz diziyi ekrana
yazdırdık.
    }

    mergeSort(dizi,0,n-1);   // Merge Sort ile diziyi sıraladık.

    printf("\n\nMerge Sort ile Siralanmis dizi : ");
    for(i=0;i<n;i++){
        printf("\t%d",dizi[i]);   // Sıralanmış diziyi ekrana yazdırdık.
    }


    int *yeniDizi = enYakinDegerlerBul (dizi,n);
    printf("\n\nEn yakin iki nokta : %d ve %d",yeniDizi[0],yeniDizi[1]);
// En yakın iki noktayı kullanıcıya belirttik.

    return 0;
}
```

EKRAN ÇIKTILARI

```
C:\Users\erent\Desktop\ODEVLER\17011702_1_b.exe
Dizinin eleman sayisi giriniz: 5

0. elemani giriniz : 24

1. elemani giriniz : 36

2. elemani giriniz : 12

3. elemani giriniz : 15

4. elemani giriniz : 26

Olusturdugumuz dizi :   24      36      12      15      26

Merge Sort ile Siralanmis dizi :        12      15      24      26      36

En yakin iki nokta : 24 ve 26
--------------------------------
Process exited after 23.3 seconds with return value 0
Press any key to continue . . .
```


C:\Users\erent\Desktop\ODEVLER\17011702_1_b.exe
Dizinin eleman sayisi giriniz: 9
0. elemani giriniz : 25
1. elemani giriniz : 28
2. elemani giriniz : 35
3. elemani giriniz : 39
4. elemani giriniz : 48
5. elemani giriniz : 10
6. elemani giriniz : 5
7. elemani giriniz : 11
8. elemani giriniz : 95
Olusturdugumuz dizi : 25 28 35 39 48 10 5 11 95
Merge Sort ile Siralanmis dizi : 5 10 11 25 28 35 39 48 95
En yakin iki nokta : 10 ve 11
Process exited after 35.93 seconds with return value 0
Press any key to continue . . .


C:\Users\erent\Desktop\ODEVLER\17011702_1_b.exe
Dizinin eleman sayisi giriniz: 8
0. elemani giriniz : 14
1. elemani giriniz : 25
2. elemani giriniz : 36
3. elemani giriniz : 88
4. elemani giriniz : 45
5. elemani giriniz : 17
6. elemani giriniz : 92
7. elemani giriniz : 50
Olusturdugumuz dizi : 14 25 36 88 45 17 92 50
Merge Sort ile Siralanmis dizi : 14 17 25 36 45 50 88 92
En yakin iki nokta : 14 ve 17
Process exited after 23.95 seconds with return value 0
Press any key to continue . . .