



**YILDIZ TEKNİK ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**BLM2512 Veri Yapıları ve Algoritmalar  
3.Ödev - GR1**

**KONU: Böl ve Yönet Algoritmalar**

**AYŞE HİRAL DOĞAN  
17011907**

## 1.YÖNTEM :

**Problem:** Bir kulede bulunan N odayı açmak için farklı büyüklükte N adet anahtar gerekiyor. Anahtarları diğer anahtarlarla veya kilitlerin büyüklüklerini diğer kilitlerle karşılaştırmadan, sadece anahtarlar kapılara takılarak kontrol yapılabilir. Her seferinde rastgele bir anahtar seçip hangi kilide uyduğunu bularak, anahtarların uydukları bütün kapıları  $O(N \cdot \lg 2N)$  karmaşıklıkla bulan algoritmayı tasarlamak.

Programda main fonksiyon dışında 2 tane fonksiyon kullanılmıştır; ParçalaVeSiral fonksiyonu , sort fonksiyonu.

```
void sort(int *Key , int*Lock , int low , int high,int n)
```

**Sort fonksiyonu** girdi olarak Key ve Lock dizilerini, low ve high indislerini ve eleman sayısını (n) alır. Geri dönüş yapmaz.

Sort fonksiyonu recursive bir fonksiyondur ve fonksiyondaki kontrol şartı low 'un high'dan küçük olmasıdır. Fonksiyon low ve high arasında random bir indis üretir ve bu indisteki sayıya göre bir Lock dizisinde bir sıralama yapmak için bu sayıyı ve Lock dizisini ParçalaVeSiral fonksiyonuna gönderir. Bu dizi **geri dönüş** olarak yollanan sayının yerleştiği indisi gönderir. Daha sonra aynı sayıya göre Key dizisinde sıralama yapmak için ( küçükler soluna büyükler sağına ); Key dizisini ve bu sayıyı (eleman) ParçalaVeSiral fonksiyonuna gönderir. Böylece aynı sayılar iki farklı dizide aynı indise yerleşmiş ve karşı karşıya gelmiş olur. Bu fonksiyon recursive olarak işlediği için; aynı işlemleri bu sefer geri dönüş olarak dönen pivot değerine göre belirlenen dizinin sol tarafı bitene kadar ve daha sonra da aynı şekilde sağ tarafı bitene kadar recursive işlem yapar. Diziler her sıralandığında yani ParçalaVeSiral fonksiyonunda işlem yapıp geri döndüğünde ekrana yazdırılır.

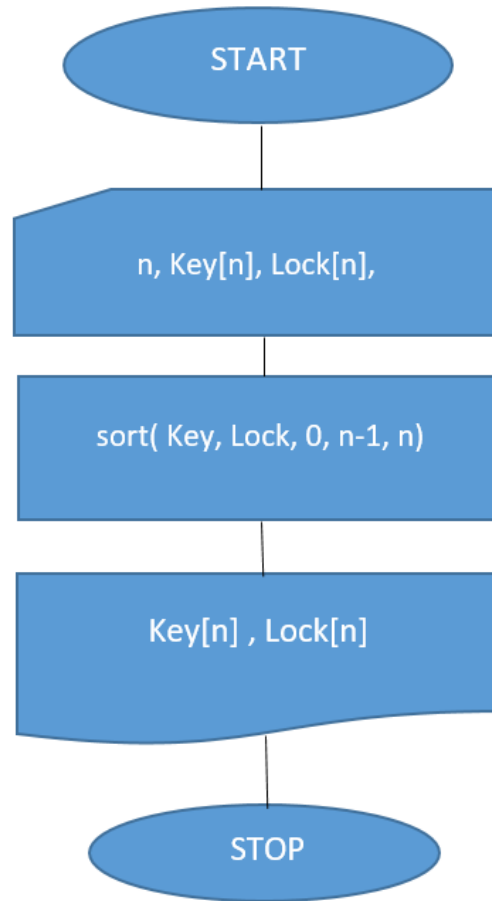
```
int ParçalaVeSiral(int *dizi,int low,int high,int pivot)
```

**ParçalaVeSiral fonksiyonu** girdi olarak bir dizi (Key yada Lock), low ve high indisi ve hangi sayıya göre sıralanacağını (pivot) alır. Geri dönüş olarak ise bu pivot'ın hangi indise yerleştiğini döner.

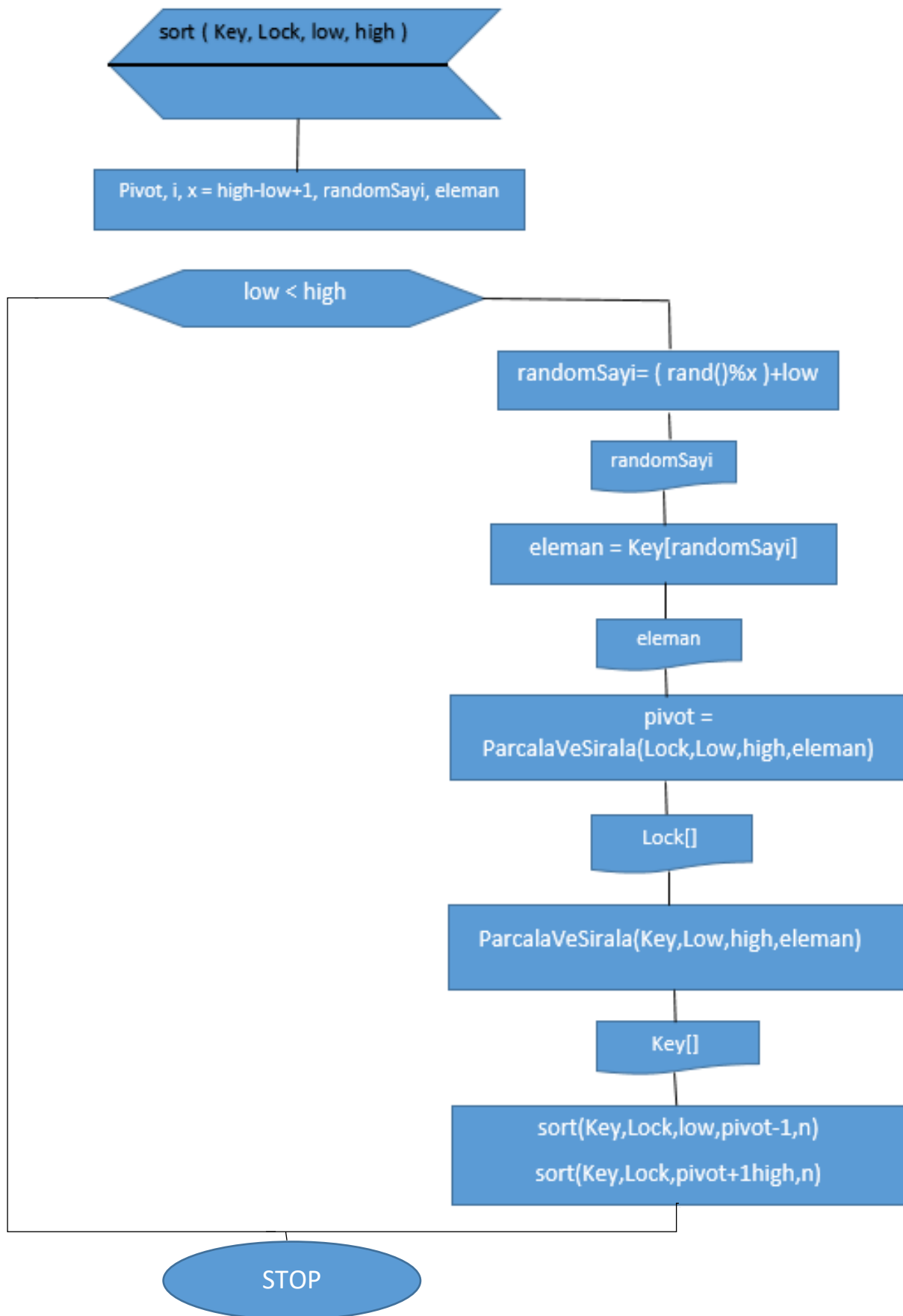
ParçalaVeSiral fonksiyonu önce diziye low olarak gelen indise pivot'ı koymak için pivot'ın yerini bulur ve low'daki elemanla yer değiştirir. Böylece pivot işlem yaptığımız kısmın (low ve high arası) en başında olur ve olması gereken indis bulunduğu oradaki elemanla yer değiştirilir. Daha sonra low'dan başlayan i indisini ve high+1'den başlayan j indisini dizi üzerinde ilerleterek pivot'tan küçük mü büyük mü diye sorgular. i indisini arttırarak ilerlerken pivot'tan büyük veya eşit bir sayı bulunursa döngüden çıkar ve j'yi eksilterek dizi üzerinde gerilemeye başlar. Önüne pivot'tan daha küçük bir sayı çıkarsa döngüden çıkar ve eğer  $i < j$  ise i'nin bulunduğu yerdeki elemanla bu elemanın yerini değiştirir. Böylece küçük sayı sol tarafa, büyük sayı sağ tarafa yerleştirilmiş olur. Bu işlem  $i < j$  olduğu sürece devam eder. İşlem bittiğinde dizinin low'unda pivot vardır. Ve j'sinin olduğu yer pivot'un yerleşmesi gereken yerdir. Buna göre dizi[low] 'a dizi[j] konur ve dizi[j] ye pivot konur. Şimdi sıra pivot'la aynı olan sayıların pivot'un yanına yerleştirilmesindedir. Bunun için for döngüsünde  $i = \text{low}$ 'dan başlayıp  $i < j$  olduğu sürece dizi[i] pivot'a eşit mi ve eşit mi diye bakar ve eşitse pivot'ın yerleştiği adresin soluna yerleştirir. (pivot'ın solundaki eleman ile yer değiştirir) Burada pivot'u-ın solunu k ( $k = j - 1$ ) indisi tutar ve her yerleştirme sonucu bir azalır. Bu şekilde aynı elemanları da pivot'ın yanına toplamış olur.

**Main fonksiyonu** Key ve Lock dizilerini kullanıcıdan alır ve bu iki diziyi ve eleman sayısını sort fonksiyonuna gönderir.

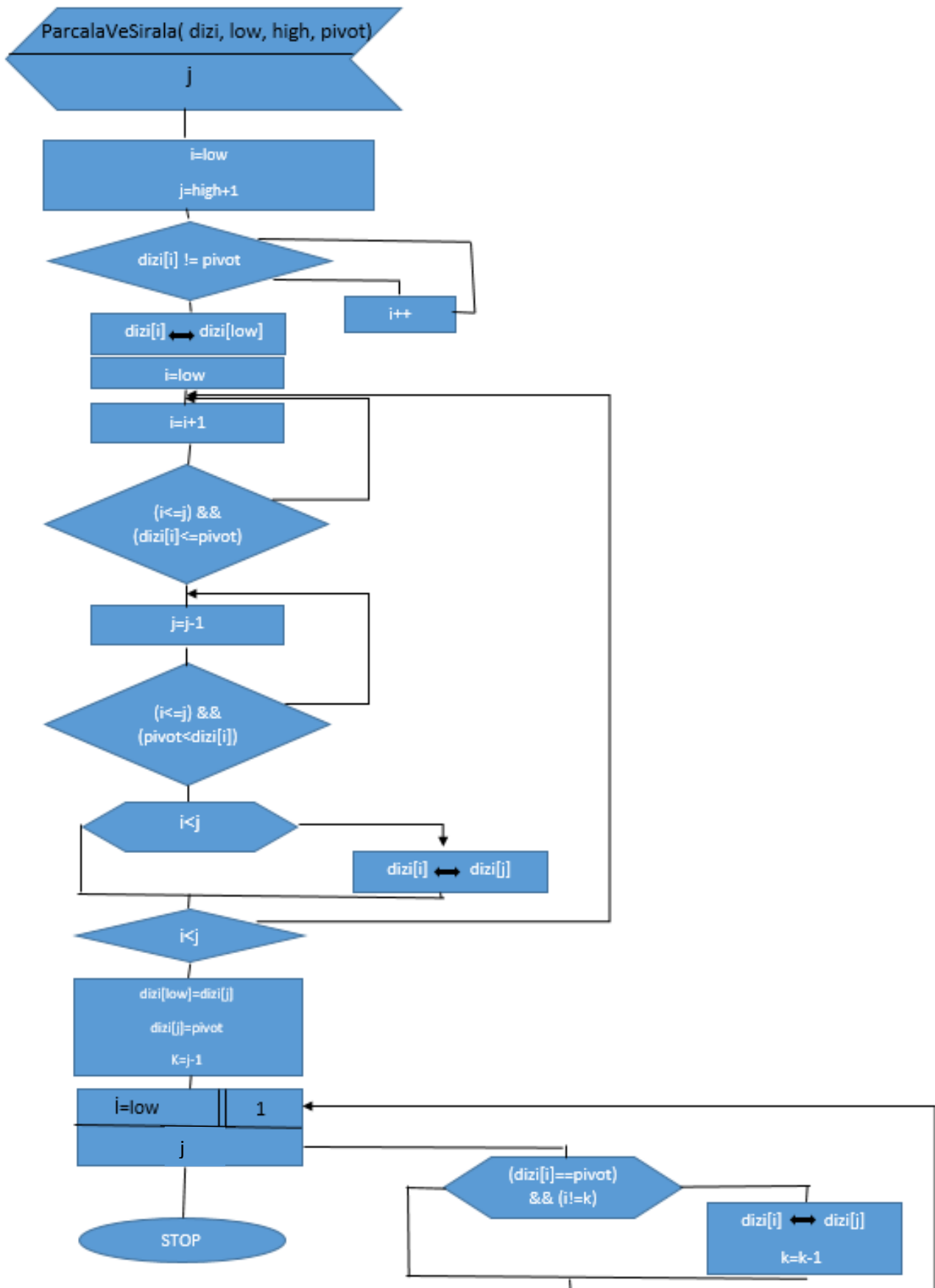
## MAIN FONKSİYONU:



## sort FONKSİYONU:



## ParcalaVeSiral FONKSİYONU:



## 2.UYGULAMA:

### Dizilerin rastgele sayılardan oluştuğu örnek ve ekran çıktıları:

Bu program KEY dizisinde tutulan anahtar değerlerini ve LOCK dizisinde tutulan kilit büyüklüklerini birbirlerine eşleştirir..  
Acılması gereken toplam oda sayısını giriniz:6

Key Dizisi İçin Anahtar Değerlerini Girin:

6  
3  
2  
5  
1  
4

Lock Dizisi İçin Kilit Büyüklüklerini Girin:

2  
4  
1  
3  
6  
5

DIZI LOCK:

2 4 1 3 6 5

DIZI KEY:

6 3 2 5 1 4

Random indis 1 secildi!

\* 3 \* için sıralama yapılıyor..

low: 0 high: 5

Lock arrayi:

1 2 3 4 6 5

low: 0 high: 5

Key arrayi:

2 1 3 5 6 4

Random indis 1 secildi!

\* 1 \* için sıralama yapılıyor..

low: 0 high: 1

Lock arrayi:

1 2 3 4 6 5

low: 0 high: 1

Key arrayi:

1 2 3 5 6 4

Random indis 4 secildi!

\* 6 \* için sıralama yapılıyor..

low: 3 high: 5

Lock arrayi:

1 2 3 5 4 6

low: 3 high: 5

Key arrayi:

1 2 3 4 5 6

Random indis 4 secildi!

\* 5 \* için sıralama yapılıyor..

low: 3 high: 4

Lock arrayi:

1 2 3 4 5 6

low: 3 high: 4

Key arrayi:

1 2 3 4 5 6

Sıralama ve eşleştirme yapıldıktan sonra Key dizisi :

1 2 3 4 5 6

Sıralama ve eşleştirme yapıldıktan sonra Lock dizisi :

1 2 3 4 5 6

-----

Process exited after 36.49 seconds with return value 0

Press any key to continue . . .

## Dizilerin küçükten büyüğe sıralı sayılardan oluştuğu örnek ve ekran çıktıları:

Bu program KEY dizisinde tutulan anahtar degerlerini ve LOCK dizisinde tutulan kilit buyukluklerini birbirlerine eslestirir..  
Acilmasi gereken toplam oda sayisini giriniz:6

Key Dizisi Icin Anahtar Degerlerini Girin:

1  
3  
5  
7  
9  
11

Lock Dizisi Icin Kilit Buyukluklerini Girin:

1  
3  
5  
7  
9  
11

DIZI LOCK:

1 3 5 7 9 11

DIZI KEY:

1 3 5 7 9 11

Random indis 4 secildi!

\* 9 \* icin siralama yapiliyor..

low: 0 high: 5

Lock arrayi:

1 3 5 7 9 11

low: 0 high: 5

Key arrayi:

1 3 5 7 9 11

Random indis 2 secildi!

\* 5 \* icin siralama yapiliyor..

low: 0 high: 3

Lock arrayi:

1 3 5 7 9 11

low: 0 high: 3

Key arrayi:

1 3 5 7 9 11

Random indis 0 secildi!

\* 1 \* icin siralama yapiliyor..

low: 0 high: 1

Lock arrayi:

1 3 5 7 9 11

low: 0 high: 1

Key arrayi:

1 3 5 7 9 11

Siralama ve eslestirme yapildiktan sonra Key dizisi :

1 3 5 7 9 11

Siralama ve eslestirme yapildiktan sonra Lock dizisi :

1 3 5 7 9 11

## Dizilerin büyükten küçüğe sıralı sayılardan oluştuğu örnek ve ekran çıktıları:

Bu program KEY dizisinde tutulan anahtar degerlerini ve LOCK dizisinde tutulan kilit buyukluklerini birbirlerine eslestirir..  
Acilmasi gereken toplam oda sayisini giriniz:7

Key Dizisi Icin Anahtar Degerlerini Girin:

130  
120  
110  
100  
90  
80  
70

Lock Dizisi Icin Kilit Buyukluklerini Girin:

130  
120  
110  
100  
90  
80  
70

DIZI LOCK:

130 120 110 100 90 80 70

DIZI KEY:

130 120 110 100 90 80 70

Random indis 6 secildi!  
\* 70 \* icin siralama yapiliyor..

low: 0 high: 6

Lock arrayi:  
70 120 110 100 90 80 130

low: 0 high: 6

Key arrayi:  
70 120 110 100 90 80 130

Random indis 6 secildi!  
\* 130 \* icin siralama yapiliyor..

low: 1 high: 6

Lock arrayi:  
70 120 110 100 90 80 130

low: 1 high: 6

Key arrayi:  
70 120 110 100 90 80 130

Random indis 5 secildi!  
\* 80 \* icin siralama yapiliyor..

low: 1 high: 5

Lock arrayi:  
70 80 110 100 90 120 130

low: 1 high: 5

Key arrayi:  
70 80 110 100 90 120 130

Random indis 5 secildi!  
\* 120 \* icin siralama yapiliyor..

low: 2 high: 5

Lock arrayi:  
70 80 110 100 90 120 130

low: 2 high: 5

Key arrayi:  
70 80 110 100 90 120 130



```

Random indis 4 secildi!
* 90 * icin siralama yapiliyor..

low: 2  high: 4

Lock arrayi:
70 80 90 100 110 120 130

low: 2  high: 4

Key arrayi:
70 80 90 100 110 120 130

Random indis 4 secildi!
* 110 * icin siralama yapiliyor..

low: 3  high: 4

Lock arrayi:
70 80 90 100 110 120 130

low: 3  high: 4

Key arrayi:
70 80 90 100 110 120 130

```

```

Siralama ve eslestirme yapildiktan sonra Key dizisi :
70 80 90 100 110 120 130
Siralama ve eslestirme yapildiktan sonra Lock dizisi :
70 80 90 100 110 120 130
-----
Process exited after 39.61 seconds with return value 0
Press any key to continue . . . █

```

### 3.SONUÇ :

#### ALGORİTMANIN İŞLEM KARMAŞIKLIĞI:

$$T(n) = 2T(n/2) + n$$

$$T(n) = 2[ 2T(n/4) + n/2 ] + n$$

$$T(n) = 4T(n/4) + 2n$$

$$T(n) = 4[ 2T(n/8) + n/4 ] + 2n$$

$$T(n) = 8T(n/8) + 3n$$

$$T(n) = 2^i T(n/2^i) + i*n$$

$$2^i \rightarrow \text{if}(n=2^i) \rightarrow i=\log_2 n$$

$$T(n) = n*T(1) + \log_2 n * n = n + n * \log_2 n = \underline{n * \log_2 n}$$