VERİ YAPILARI VE ALGORİTMALAR

ÖDEV 3

Böl ve Yönet Algoritmalar

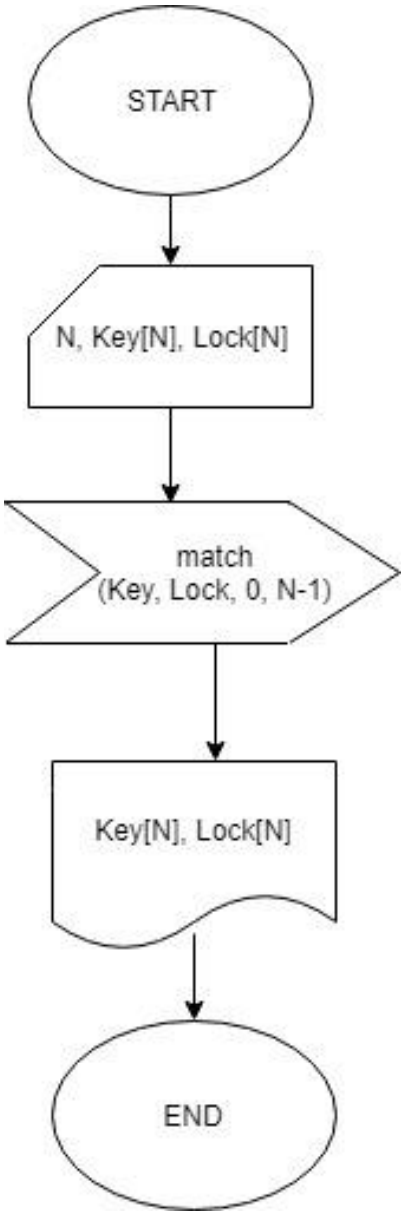
AHMET ONUR AKMAN

16011059

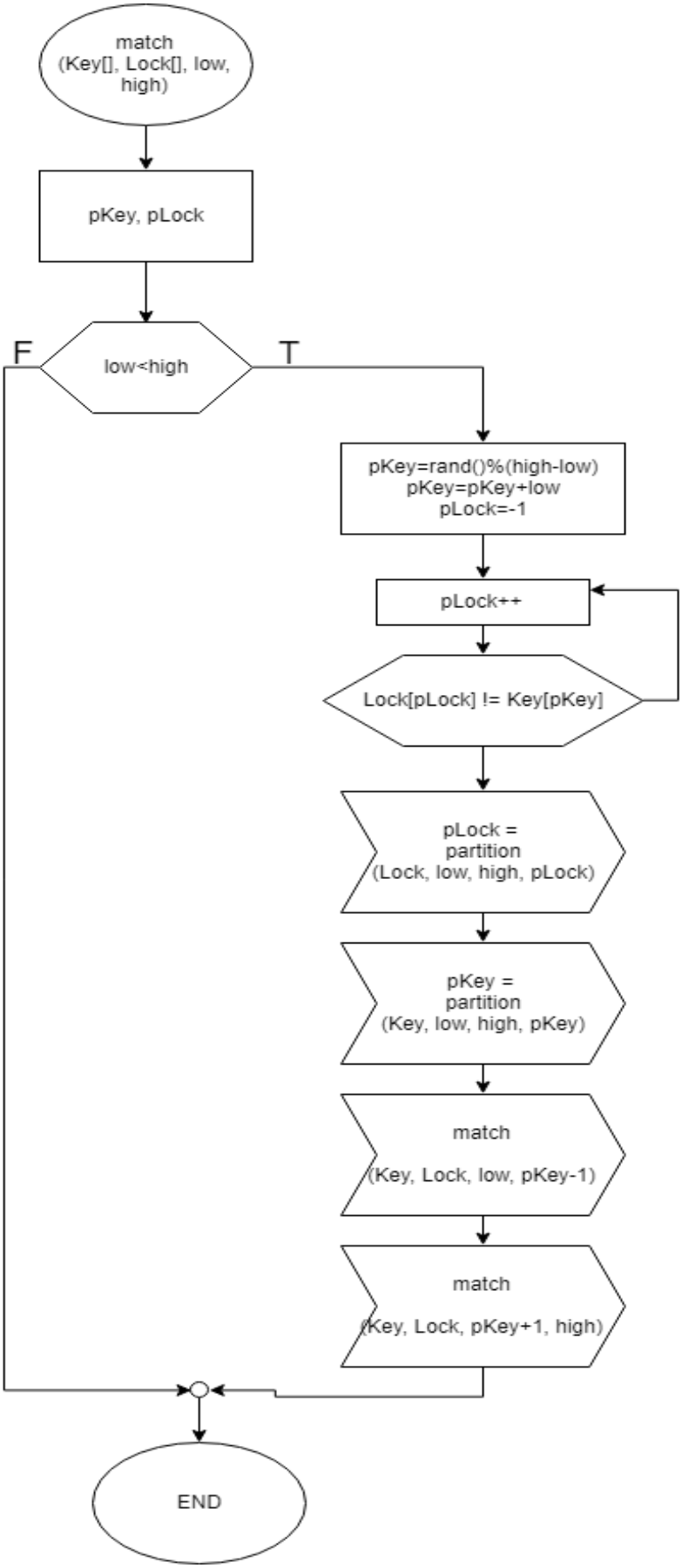
Bölüm I – Yöntem

Soruda bizden aynı sayıları farklı sırlamayla içeren 2 dizinin elemanlarının böl ve önet algoritmaları ile eşleştirilip sıralanması isteniyor. Ben algoritmamı tasarlarken elimizdeki bu iki diziyi aynı anda işleyen 2 adet fonksiyon kullandım: match ve partition.
Akış şemasında gösterilmese de uygulama değerlerin rastgele üretilmesi ve Lock kümesinin rastgele dizilmesi gibi seçenekler sunuyor.

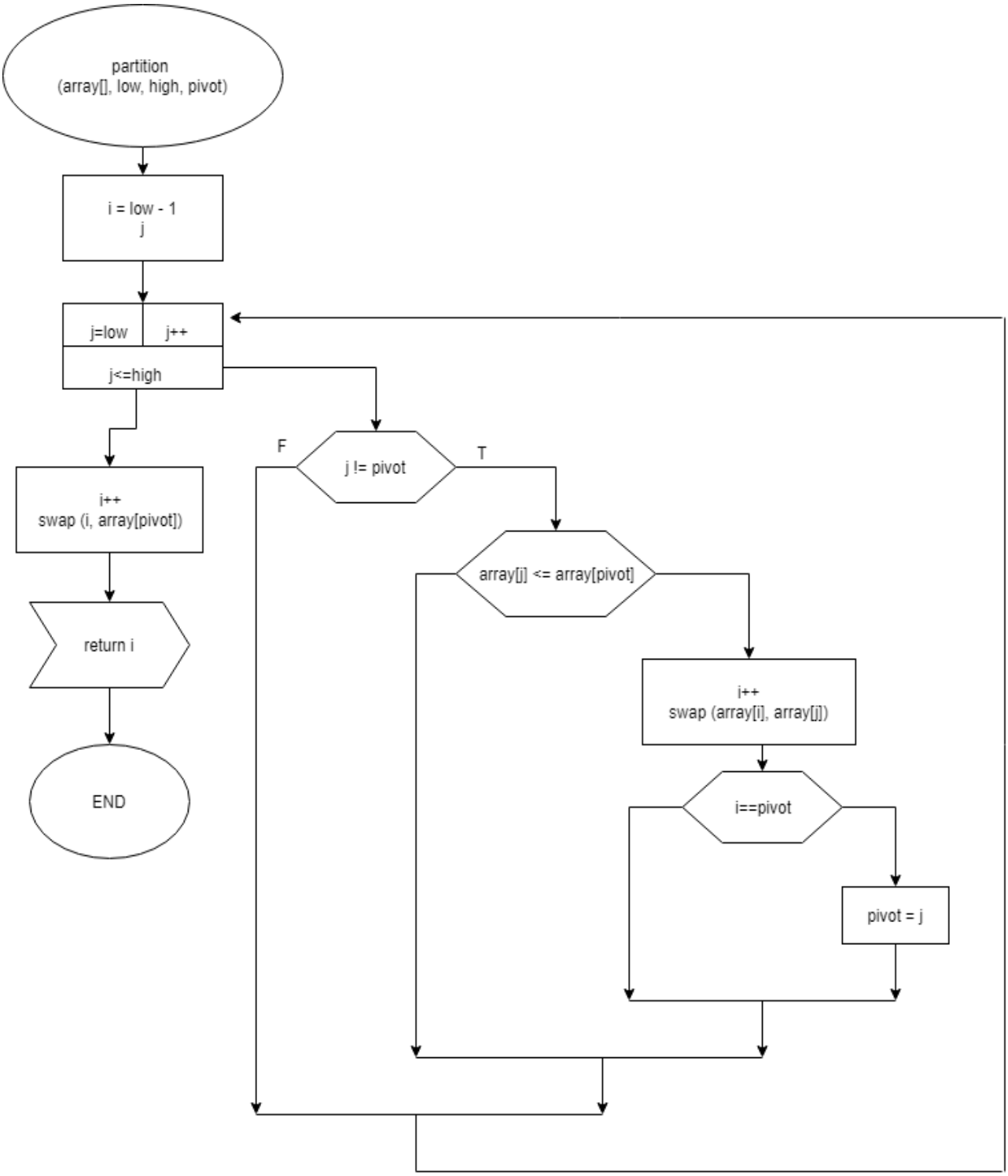
a) Main



b) Match


match
(Key[], Lock[], low, high)
pKey, pLock
F
low<high
T
pKey=rand()%(high-low)
pKey=pKey+low
pLock=-1
pLock++
Lock[pLock] != Key[pKey]
pLock =
partition
(Lock, low, high, pLock)
pKey =
partition
(Key, low, high, pKey)
match
(Key, Lock, low, pKey-1)
match
(Key, Lock, pKey+1, high)
END

c) Partition


partition
(array[], low, high, pivot)
i = low - 1
j
j=low
j++
j<=high
F
j != pivot
T
i++
swap (i, array[pivot])
return i
END
array[j] <= array[pivot]
i++
swap (array[i], array[j])
i==pivot
pivot = j

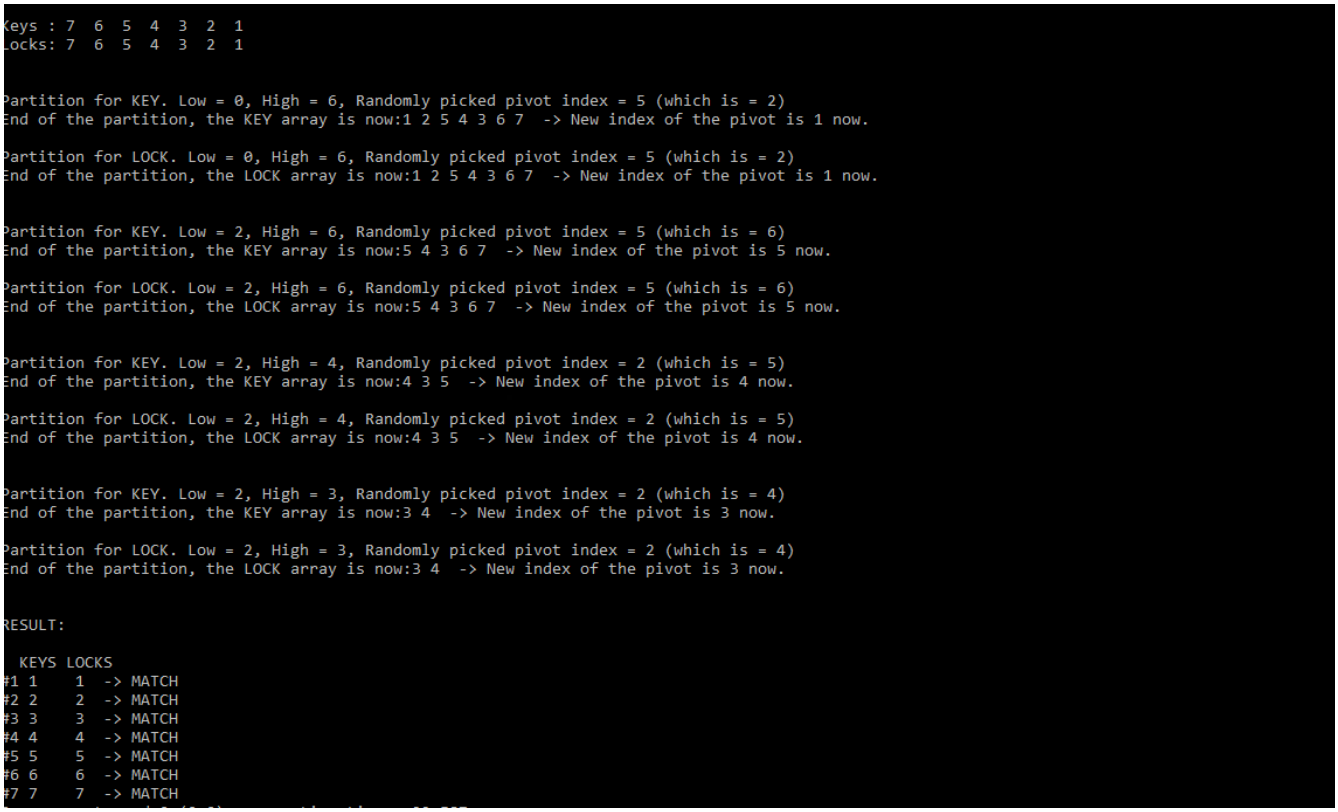
Bölüm II – Uygulama

```
Keys : 41  18467  6334  26500  19169  15724  11478
Locks: 41  26500  15724  11478  19169  18467  6334


Partition for KEY. Low = 0, High = 6, Randomly picked pivot index = 3 (which is = 26500)
End of the partition, the KEY array is now:41 18467 6334 19169 15724 11478 26500  -> New index of the pivot is 6 now.

Partition for LOCK. Low = 0, High = 6, Randomly picked pivot index = 1 (which is = 26500)
End of the partition, the LOCK array is now:41 15724 11478 19169 18467 6334 26500  -> New index of the pivot is 6 now.


Partition for KEY. Low = 0, High = 5, Randomly picked pivot index = 2 (which is = 6334)
End of the partition, the KEY array is now:41 6334 18467 19169 15724 11478  -> New index of the pivot is 1 now.

Partition for LOCK. Low = 0, High = 5, Randomly picked pivot index = 5 (which is = 6334)
End of the partition, the LOCK array is now:41 6334 11478 19169 18467 15724  -> New index of the pivot is 1 now.


Partition for KEY. Low = 2, High = 5, Randomly picked pivot index = 3 (which is = 19169)
End of the partition, the KEY array is now:18467 15724 11478 19169  -> New index of the pivot is 5 now.

Partition for LOCK. Low = 2, High = 5, Randomly picked pivot index = 3 (which is = 19169)
End of the partition, the LOCK array is now:11478 18467 15724 19169  -> New index of the pivot is 5 now.


Partition for KEY. Low = 2, High = 4, Randomly picked pivot index = 3 (which is = 15724)
End of the partition, the KEY array is now:11478 15724 18467  -> New index of the pivot is 3 now.

Partition for LOCK. Low = 2, High = 4, Randomly picked pivot index = 4 (which is = 15724)
End of the partition, the LOCK array is now:11478 15724 18467  -> New index of the pivot is 3 now.


RESULT:

  KEYS LOCKS
#1 41    41  -> MATCH
#2 6334    6334  -> MATCH
#3 11478    11478  -> MATCH
#4 15724    15724  -> MATCH
#5 18467    18467  -> MATCH
#6 19169    19169  -> MATCH
#7 26500    26500  -> MATCH
```

1) Rastgele değerler ile oluşturulmuş diziler için program işleyişi ve çıktısı

```
Keys : 1  2  3  4  5  6  7
Locks: 1  2  3  4  5  6  7


Partition for KEY. Low = 0, High = 6, Randomly picked pivot index = 5 (which is = 6)
End of the partition, the KEY array is now:1 2 3 4 5 6 7  -> New index of the pivot is 5 now.

Partition for LOCK. Low = 0, High = 6, Randomly picked pivot index = 5 (which is = 6)
End of the partition, the LOCK array is now:1 2 3 4 5 6 7  -> New index of the pivot is 5 now.


Partition for KEY. Low = 0, High = 4, Randomly picked pivot index = 3 (which is = 4)
End of the partition, the KEY array is now:1 2 3 4 5  -> New index of the pivot is 3 now.

Partition for LOCK. Low = 0, High = 4, Randomly picked pivot index = 3 (which is = 4)
End of the partition, the LOCK array is now:1 2 3 4 5  -> New index of the pivot is 3 now.


Partition for KEY. Low = 0, High = 2, Randomly picked pivot index = 0 (which is = 1)
End of the partition, the KEY array is now:1 2 3  -> New index of the pivot is 0 now.

Partition for LOCK. Low = 0, High = 2, Randomly picked pivot index = 0 (which is = 1)
End of the partition, the LOCK array is now:1 2 3  -> New index of the pivot is 0 now.


Partition for KEY. Low = 1, High = 2, Randomly picked pivot index = 1 (which is = 2)
End of the partition, the KEY array is now:2 3  -> New index of the pivot is 1 now.

Partition for LOCK. Low = 1, High = 2, Randomly picked pivot index = 1 (which is = 2)
End of the partition, the LOCK array is now:2 3  -> New index of the pivot is 1 now.


RESULT:

  KEYS LOCKS
#1 1    1  -> MATCH
#2 2    2  -> MATCH
#3 3    3  -> MATCH
#4 4    4  -> MATCH
#5 5    5  -> MATCH
#6 6    6  -> MATCH
#7 7    7  -> MATCH
```

2) Küçükten büyüğe sıralı değerler ile oluşturulmuş diziler için program işleyişi ve çıktısı



3) Büyükten küçüğe sıralı değerler ile oluşturulmuş diziler için program işleyişi ve çıktısı

Bölüm III – Sonuç

Partition fonksiyonunun bir tarafa N/7, diğer tarafa 6N/7 eleman atadığı bir senaryoyu düşünelim.

$T(N) = T(N/7) + T(6N/7) + \theta(N)$

Bu eşitliğin çözümünden **O(NlogN)** sonucuna varırırız.