

ALGORİTMA ANALİZİ ÖDEV RAPORU
BÖL VE YÖNET ALGORİTMALAR



Grup 1 – Mine Elif Karslıgil

MÜDAFER KAYMAK

20011093

Mudafer.kaymak@std.yildiz.edu.tr

Video Linki : <https://youtu.be/lbNAKgfqpuo>

Problem Tanımı

Problem 1, müzeden çalınan kolye olayını böl ve yönet algoritmasıyla çözmeyi gerektirir. Doğrulukları farklı olan çalışan gruplarını sorgulayarak, doğruyu söyleyen kişileri belirleyip kolyeyi çalanları tespit etmeyi amaçlar. Problem 2 ise N odalı kulede farklı büyüklükte N anahtarın kullanılması gereken durumu ele alır. Anahtarların uygunluğunu $O(N \cdot \log(N))$ karmaşıklığıyla kontrol eden bir algoritma geliştirilmesi beklenir.

Problemin Çözümü

Bu kod, iki farklı dizinin sıralanması için "QuickSort" algoritmasını uygular. Bu algoritma, kilidin ve anahtarın eşleşmesini sıralamak için kullanılır. İlk olarak, "partition" fonksiyonu, bir diziyi parçalar ve bir pivot elemanı etrafında küçük, eşit ve büyük olanları ayırır. Ardından, "quickSort" fonksiyonu bu bölünmüş dizileri sıralamak için tekrar tekrar çağırılır.

Ana "quickSort" fonksiyonu, kilidin ve anahtarın eşleşmesini kontrol eder. İlk olarak, anahtar dizisi temel alınarak bir pivot seçilir ve bu pivot, kilidin parçalanmasında kullanılır. Daha sonra, eşleşme koşullarına göre dizi elemanları karşılaştırılır ve sıralama gerçekleştirilir. Bu işlem, rekürsif olarak, diziler parçalanıp sıralanana kadar devam eder.

Karşılaşılan Zorluklar

Bu problemde olası inputların çok geniş olacağından ötürü algoritmayı uygularken olabildiğince tekrardan kaçınılması gerekiyordu bundan dolayı rekürsif bir yapı oluşturmak her iki fonksiyonu da olabildiğince kısaltmak gerekiyordu. Bunu sağlamak adına olabilecek en optimal rekürsif yapıyı oluşturma aşamasında zorluk yaşandı.

Karmaşıklık Analizi

Input : Arr: Int Array: Array containing the values to change its order

start: Int: first index of the array

finish: Int: last index of the array

pivot: Int: current pivot number

Partition:

```
i = start
for j = start to finish - 1 do
  if arr[j] < pivot then
    Swap(arr[i], arr[j])
    i = i + 1
  else if arr[j] == pivot then
    Swap(arr[j], arr[finish])
  j = j - 1
```

```
Swap(arr[i], arr[finish])
return i
```

Input : Arr: Int Array: 2 Array containing the values to be sorted

start : Int: first index of the array

finish: Int: last index of the array

QuickSort:

```
if start < finish then
  pivot = CALL Partition(lock, start, finish, key[finish])
  CALL Partition(key, start, finish, lock[pivot])
```

```
CALL QuickSort(lock, key, start, pivot - 1)
CALL QuickSort(lock, key, pivot + 1, finish)
```

Bu algoritmanın ideal senaryosunda dizinin uzunluğuna n dersek her seferinde diziyi bölecektir ve ideal pivottan çok uzak seçmediği sürece rekürsif ağacının derinliği yaklaşık olarak $\log n$ olur. Karşılaştırılma yapıldığında ise her rekürsif katında dizinin boyutu n kadar karşılaştırma yapılacaktır. Matematiksel olarak, ideal senaryoda karşılaştırma sayısı $n \cdot \log n$ olacaktır. Bu hesaplara da $O(n \log n)$ zaman karmaşıklığına ulaşılmış olur.

Master Teori:

$$T(n) = a \cdot T(n/b) + n^d$$

$$a = 2 \quad b = 2 \quad d = 1$$

$$a > b^d$$

Soru işareti '=' olacağından dolayı $O(n \log n)$ karmaşıklığına ulaşırız.

Ekran görüntüleri

```
int main() {
    int lock[] = {4, 1, 2, 3, 5};
    int key[] = {5, 3, 2, 1, 4};

    C:\Users\M...DAFERKAYMAK\Desktop\

    Locks = 1 2 3 4 5
    Keys = 1 2 3 4 5
    -----
}
```

Şekil 1 - Senaryo 1

```
int main() {
    int lock[] = {5, 4, 3, 2, 1};
    int key[] = {5, 4, 3, 2, 1};

    C:\Users\M...DAFERKAYMAK\Desktop\Alg

    Locks = 1 2 3 4 5
    Keys = 1 2 3 4 5
    -----
}
```

Şekil 2 – Senaryo 2

```
int main() {
    int lock[] = {3, 1, 2, 5, 4};
    int key[] = {5, 4, 3, 2, 1};

    C:\Users\M...DAFERKAYMAK\Deskt

    Locks = 1 2 3 4 5
    Keys = 1 2 3 4 5
    -----
}
```

Şekil 3 - Senaryo 3

```
//inputs
int lock[] = {3,4,2,5,1,8};
int key[] = {2,3,5,4,8,1};

C:\Users\M...DAFERKAYMAK\Desl

    Locks = 1 2 3 4 5 8
    Keys = 1 2 3 4 5 8
    -----
}
```

Şekil 4 - Senaryo 4

```
//inputs
int lock[] = {3,2};
int key[] = {2,3};

C:\Users\M...DAFERKAYMAI

    Locks = 2 3
    Keys = 2 3
    -----
}
```

Şekil 5- Senaryo 5