

Min-Max (Divide& Conquer)

Bu yöntemde problem daha küçükalt probleme bölünür ve bu alt problemler çözüldüğünde ana problemde çözülmüş olur

Sıralı olmayan dizide en küçük ve en büyük sayının bulunması

Algoritma

```

if  $n - k == 1$  then
    if ( $A[n] > A[k]$ ) return ( $A[k], A[n]$ )
    else return ( $A[n], A[k]$ )
end
else
     $m = (k + n) / 2$ 

     $(x_1, y_1) = \text{MinMax}(k, m)$             $T(\frac{n}{2})$ 

     $(x_2, y_2) = \text{MinMax}(m + 1, n)$        $T(\frac{n}{2})$ 

     $x = \text{Min}(x_1, x_2)$                    1
     $y = \text{Min}(y_1, y_2)$                    1
    return ( $x, y$ )                        + -----
                                            $2 \cdot T(\frac{n}{2}) + 2$ 
    
```

Analiz

$n = 2^k$ // Temel işlem sayısına göre yapılır (Karşılaştırma sayısı)

$T(n) = \text{Karşılaştırma sayısı}$

$$T(n) = \begin{cases} n = 2 & T(2) = 1 \\ n > 2 & T(n) = 2T(\frac{n}{2}) + 2 \end{cases} \quad // \text{ Durma noktaları}$$

k=1 iken $n = 2^1$ $T(2) = 1$

k=2 iken $n = 2^2$ $T(n) = T(\frac{n}{2}) + 2$

k=3 iken $n = 2^3$ $T(n) = 2(2T(\frac{n}{4}) + 2) + 2 = 4T(\frac{n}{4}) + 4 + 2$

k=4 iken $n = 2^4$ $T(n) = 2(T(\frac{n}{8}) + 2) + 4 + 2 = 8T(\frac{n}{8}) + 8 + 4 + 2$

$$T(n) = 2^{k-1} \underbrace{T(\frac{n}{2^{k-1}})}_{T(2) \text{ olabilmesi için}} + \sum_{j=1}^{k-1} 2^j = 2^{k-1} + \sum_{j=1}^{k-1} 2^j$$

Formül : $\sum_{k=1}^n r^{k-1} = \frac{1-r^n}{1-r} \Rightarrow r^0 + r^1 + r^2 + \dots + r^n$

$$\sum_{j=1}^{k-1} 2^j = 2^1 + 2^2 + \dots + 2^{k-2} + 2^{k-1}$$

$$\sum_{j=0}^{k-2} 2^{j+1} = 2^1 + 2^2 + \dots + 2^{k-2} + 2^{k-1}$$

$$\underbrace{\sum_{j=0}^{k-2} 2^{j+1}}_{2^0+2^1+\dots+2^{k-3}+2^{k-2}} = \frac{1-2^{k-2}}{1-2} = 2^{k-2} - 1 + 2^{k-2} = \frac{2^k}{2} - 1$$

$$T(n) = 2^{k-1} + \sum_{j=1}^{k-1} 2^j \Rightarrow 2^{k-1} + \underbrace{\sum_{j=0}^{k-2} 2^{j+1}}_{2^0+2^1+\dots+2^{k-3}+2^{k-2}}$$

$$T(n) = 2^{k-1} + 2 \cdot \left(\frac{2^k}{2} - 1\right)$$

$$T(n) = 2^{k-1} + 2^k - 2$$

$$T(n) = \frac{n}{2} + n - 2$$

$$T(n) = \frac{n}{2} + n - 2$$

O(n)

Böl yönet olmasına rağmen logaritmik bir sonuç çıkmadı. Çünkü atılan taraf yok. Sol ve sağ iki tarafta işlem yapıldığı için sonuç logaritmik değil

QUIZ

```
for (int i = 0; i < data.length; ++i) {
    if (data[i] > key) {
        return -1;
    } else if (data[i] == key) {
        return i;
    }
}
return -1;
```

1. T(n) en iyi ve en kötü durum için nedir? // Buna benzer bir soruydu O(n)
2. $n^2, \log n, n \log n, 2^n, n$ BigO'ya göre yavaştan hızlıya sıralayın