

5) Kurduğum algoritma da; recursive fonksiyon kullandım.
compare Scales fonksiyonunda eklemeye yaptım.

listenin elemanlarının toplamını listenin boyutuna buldum. Bu
şekilde birim başına düşen ağırlığı buldum, birim başına
düşen ağırlık hariç tutulmuş recursive fonksiyonda o listeye
gittim ve küçük cevizin indexini buldum.

recursive fonksiyona 3 tane recursive kol açılmayı, durdurma
durum ekledim;

1) listenin boyutunun 1 den küçük olması;

liste baş, index--1

2) listenin boyutunun 1 olması,

küçük ceviz ilk elemandır.

3) listenin boyutunun 2 olması;

küçük eleman küçük cevizdir, onun indexi return edilir.

Bu şekilde çalışma zamanını hesaplamak istersem;

$$T(n) = 2 \cdot T(n/2) + 1 \rightarrow \text{compare Scales fonksiyonu}$$

2 tane recursive kol var.

→ size in yarısı kadar eleman ile yeni liste isteniyor.

Best case :

recursive fonksiyonu durdurma kollarından 1 ine girmesi;

Bu da $O(1)$

worst case : n tane elemanı bölmesi;

Bu da ön sayfada bulunduyu ütere $O(n)$ dir!

Average case :

$$T(n) = 2 T(n/2) + 1$$

$$n, n/2, n/4, n/8 \dots \dots 2, 1, 0$$

$$\sum_{i=0}^{\lg n} \frac{n}{2^i} = \underbrace{n + n/2 + n/4 + \dots + \frac{n}{2^{\lg n}}}_{\lg n \text{ tane terim var,}}$$

$$\underbrace{\frac{n}{2^{\lg n}}}_{n} \cdot \lg n = \frac{n}{n} \cdot \lg n = \lg n$$

$$T(n) \in \Theta(\lg n)$$