

QuickSort

### Melhor Caso

O pivô escolhido divide o array em duas partes aproximadamente iguais. A recorrência é:

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

Cálculo do Custo Total em Formato de Somatória:

Nível 0 (Raiz):

Custo é  $n$

Nível 1:

O array é dividido em duas sublistas de tamanho  $n/2$ .

Custo total neste nível é  $2 \cdot n/2 = n$

Nível 2:

Cada sublista é dividida em duas partes de tamanho  $n/4$

Custo total neste nível é  $4 \cdot (n/4) = n$

Nível k:

Existem  $2^k$  sublistas, cada uma de tamanho  $n/(2^k)$

Custo total neste nível é  $2^k \cdot n/(2^k) = n$

### Número Total de Níveis:

O número de níveis é  $\log_2 n$

Somatória :  $T(n) = \text{Custo Nível 0} + \text{Custo Nível 1} + \text{Custo Nível 2} + \dots + \text{Custo no Nível } \log_2 n$

$$T(n) = \sum_{k=0}^{\log_2 n} n$$

$$T(n) = (\log_2 n + 1) \cdot n$$

**Recorrência:**  $T(n) = n + 2 \cdot T(n/2)$

**Custo Total em cada Nível:**  $n$

**Número Total de Níveis:**  $\log_2 n$

**Somatória Total dos Custos:**  $T(n) = (n \log_2 n + n)$

**Complexidade:**  $O(n \log n)$

### **Médio Caso:**

pivô divide o array de forma que as partes são aproximadamente iguais, em média. A recorrência para o caso médio é a mesma do melhor caso

### **Pior Caso**

pivô escolhido é sempre o menor ou o maior elemento, levando a partições altamente desbalanceadas. O array é dividido de forma que uma das partições tem  $n-1$  elementos e a outra tem 0 elementos

### **Recorrendo:**

Nível 0 (Raiz):

Custo é  $n$

Nível 1:

O array é dividido em uma sublista de tamanho  $n-1$  e uma de tamanho 0.

Custo é  $n-1$

Nível 2:

A sublista de tamanho  $n-1$  é dividida em uma sublista de tamanho  $n-2$  e uma de tamanho 0.

Custo é  $n-2$ .

Nível  $k$ :

- A cada nível, o custo é  $n-k$ , até que a sublista tenha tamanho 1

$$T(n) = \sum_{k=0}^{n-1} (n - k)$$

$$T(n) = n + (n - 1) + (n - 2) + \dots + 1$$

$$T(n) \downarrow \frac{n(n+1)}{2}$$

**Recorrência:**  $T(n) = n + T(n-1)$

**Custo Total em cada Nível:**  $n-k$  (onde  $k$  é o nível atual)

**Número Total de Níveis:**  $n$

**Somatória Total dos Custos:**  $T(n) = (n(n+1))/2$

**Complexidade:**  $O(n^2)$