

Conceito de Partição

A partição de um array consiste em dispor seus elementos de forma que à esquerda de um elemento $A[s]$ fiquem os menores ou iguais a ele, e à direita, os maiores ou iguais. O elemento $A[s]$ ficará em sua posição final no array ordenado, permitindo a classificação independente dos subarrays à esquerda e à direita.

Pseudocódigo

```
ALGORITHM HoarePartition(A[l..r])
//Partitions a subarray by Hoare's algorithm, using the first element
// as a pivot
//Input: Subarray of array A[0..n - 1], defined by its left and right
// indices l and r (l < r)
//Output: Partition of A[l..r], with the split position returned as
// this function's value
p ← A[l]
i ← l; j ← r + 1
repeat
repeat i ← i + 1 until A[i] ≥ p
repeat j ← j - 1 until A[j] ≤ p
swap(A[i], A[j])
until i ≥ j
swap(A[l], A[j]) //undo last swap when i ≥ j
return j
```

Quicksort vs Merge Sort

Diferentemente do mergesort, que divide seus elementos de entrada de acordo com sua posição no array, o quicksort os divide de acordo com seu valor. Diferentemente do mergesort, onde a divisão do problema em dois subproblemas ocorre imediatamente, todo o trabalho no quicksort ocorre na etapa de divisão, sem a necessidade de combinar as soluções dos subproblemas.

Quicksort

Algoritmo de ordenação baseado na abordagem "divide and conquer".

Execuções

$C_{best}(n) = 2C_{best}(n/2) + n$ for $n > 1$, $C_{best}(1) = 0$.

$C_{worst}(n) = (n + 1) + n + \dots + 3 = ((n + 1)(n + 2))/2 - 3$

$C_{avg}(n) \approx 2n \ln n \approx 1.39n \log_2 n$.

Pseudocódigo

```
ALGORITHM Quicksort(A[l..r])
// Ordena um subarray pelo quicksort
// Entrada: Subarray do array A[0..n - 1], definido por seus
// índices esquerdo e direito
// Saída: Subarray A[l..r] ordenado em ordem não
// decrescente
se l < r
s ← Partition(A[l..r]) // s é uma posição de divisão
Quicksort(A[l..s - 1])
Quicksort(A[s + 1..r])
```

