

Algoritmos - Aula 6

Fernando Raposo

Vamos ver

- Quick sort

Quick Sort

- Um dos algoritmos de ordenação mais usados do mundo;
- **Java** o utiliza

sort

```
public static void sort(int[] a)
```

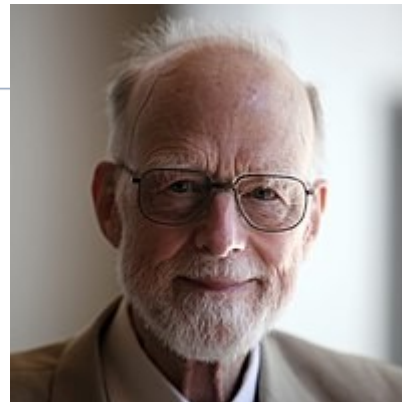
Sorts the specified array into ascending numerical order.

Implementation note: The sorting algorithm is a Dual-Pivot Quicksort by Vladimir Yaroslavskiy, Jon Bentley, and Joshua Bloch. This algorithm offers $O(n \log(n))$ performance on many data sets that cause other quicksorts to degrade to quadratic performance, and is typically faster than traditional (one-pivot) Quicksort implementations.

Parameters:

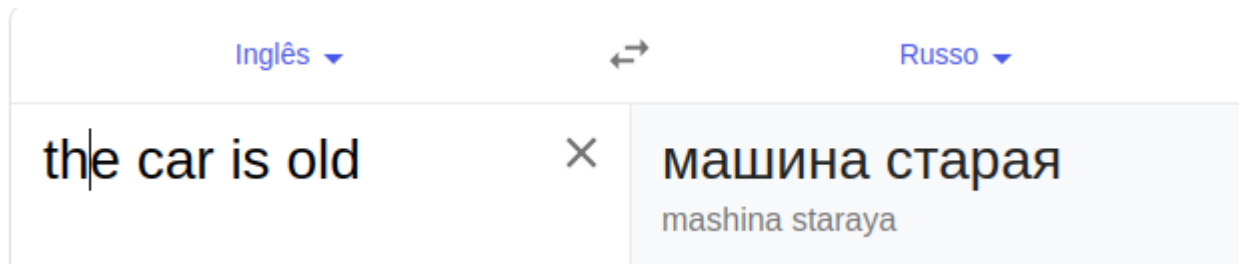
a - the array to be sorted

- Desenvolvido por C.A.R. Hoare
- Dividir-para-conquistar
- Estável
- In-Place



Quick Sort: História

- [Quando Hoare era um estudante em Moscou entrou em um projeto de tradução Russo -> Inglês](#)
- O dicionário era armazenado em fitas magnéticas, então para traduzir uma sentença e evitar percorrer a fita várias vezes, era necessário ordenar as palavras de uma frase em ordem alfabética;
- Ano de 1960.



Quick sort

- O algoritmo escolhe um pivô, e faz partições em volta do pivô escolhido;
- É criada uma função `partição()`
 - Dado um array e um elemento x como pivô que dividirá a lista em duas partes;
 - rearranje a lista de forma que todos os elementos anteriores ao pivô sejam menores que ele, e todos os elementos posteriores ao pivô sejam maiores que ele;
 - De forma recursiva, repetir o processo para ambos os lados.

Quick Sort

- Podemos escolher qualquer elemento como pivô.
- Por exemplo o elemento do meio do array.



Quick Sort

- Escolhido o elemento destacado
- Elementos menores que 7 à esquerda dele;
- Elementos maiores que 7 à direita dele;



Quick Sort

- Escolhido o elemento destacado
- Elementos menores que 7 à esquerda dele;
- Elementos maiores que 7 à direita dele;



6 > 7? Não, fica onde está.

Quick Sort

- Escolhido o elemento destacado
- Elementos menores que 7 à esquerda dele;
- Elementos maiores que 7 à direita dele;



5 > 7? Não, fica onde está.

Quick Sort

- Escolhido o elemento destacado
- Elementos menores que 7 à esquerda dele;
- Elementos maiores que 7 à direita dele;



9 > 7? Sim, fica onde está.

Quick Sort

- Escolhido o elemento destacado
- Elementos menores que 7 à esquerda dele;
- Elementos maiores que 7 à direita dele;



8 > 7? Sim, fica onde está.

Quick Sort

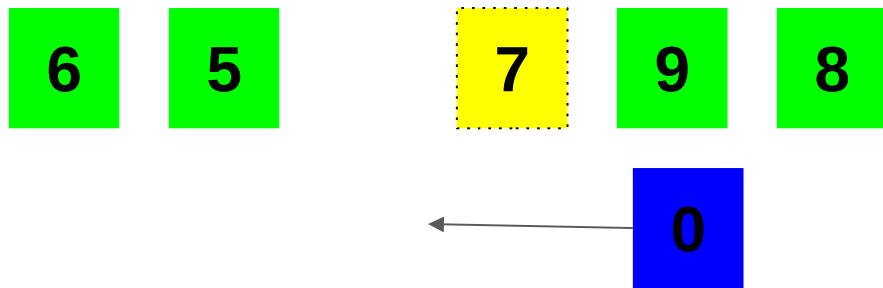
- Escolhido o elemento destacado
- Elementos menores que 7 à esquerda dele;
- Elementos maiores que 7 à direita dele;



0 > 7? Não, joga para o lado esquerdo.

Quick Sort

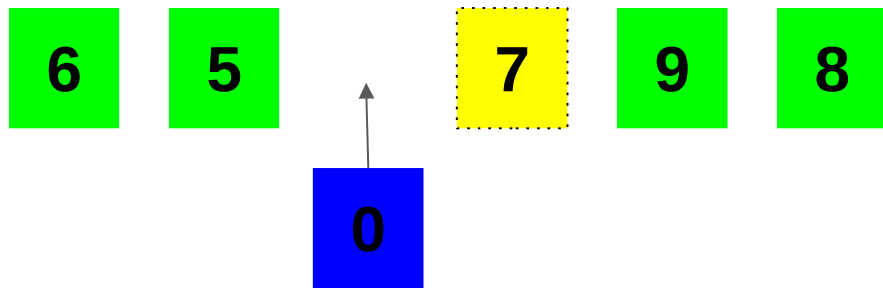
- Escolhido o elemento destacado
- Elementos menores que 7 à esquerda dele;
- Elementos maiores que 7 à direita dele;



$0 > 7$? Não, joga para o lado esquerdo.

Quick Sort

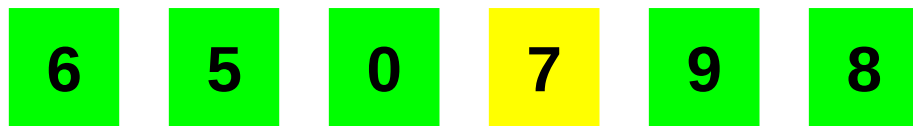
- Escolhido o elemento destacado
- Elementos menores que 7 à esquerda dele;
- Elementos maiores que 7 à direita dele;



$0 > 7$? Não, joga para o lado esquerdo.

Quick Sort

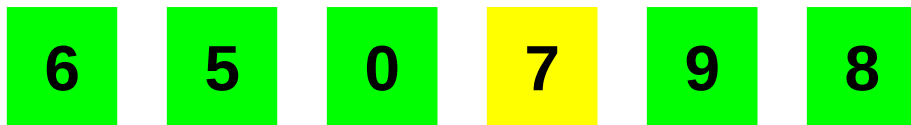
- Escolhido o elemento destacado
- Elementos menores que 7 à esquerda dele;
- Elementos maiores que 7 à direita dele;



Corrigido!

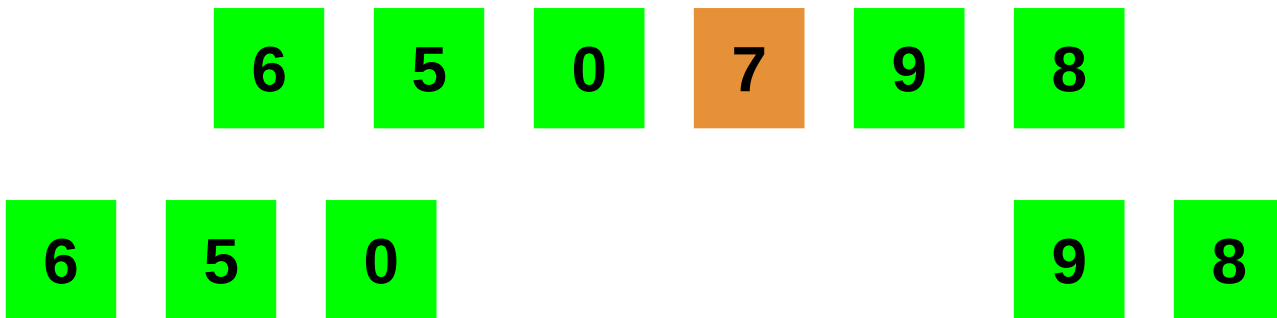
Quick Sort

- Eu agora já sei que o elemento pivô está na posição correta;
- Preciso tratar os sub-arrays do lado esquerdo e direito;



Quick Sort

- Eu agora já sei que o elemento pivô está na posição correta;
- Preciso tratar os sub-arrays do lado esquerdo e direito;



Quick Sort

- Escolhendo o pivô



Início...

Quick Sort

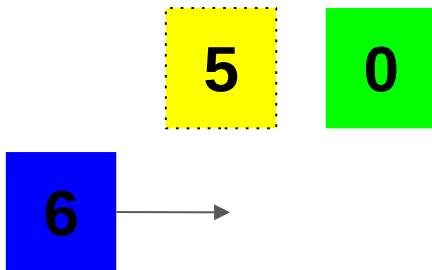
- Escolhido o elemento destacado
- Elementos menores que 5 à esquerda dele;
- Elementos maiores que 5 à direita dele;



6 > 5? Sim, jogar para o lado direito

Quick Sort

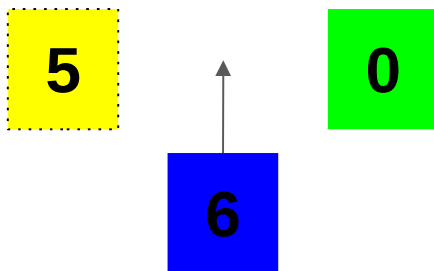
- Escolhido o elemento destacado
- Elementos menores que 5 à esquerda dele;
- Elementos maiores que 5 à direita dele;



6 > 5? Sim, jogar para o lado direito

Quick Sort

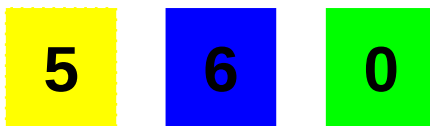
- Escolhido o elemento destacado
- Elementos menores que 5 à esquerda dele;
- Elementos maiores que 5 à direita dele;



6 > 5? Sim, jogar para o lado direito

Quick Sort

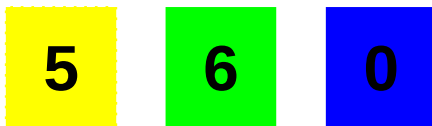
- Escolhido o elemento destacado
- Elementos menores que 5 à esquerda dele;
- Elementos maiores que 5 à direita dele;



Pronto!

Quick Sort

- Escolhido o elemento destacado
- Elementos menores que 5 à esquerda dele;
- Elementos maiores que 5 à direita dele;



0 > 5? Não! Jogar para o lado esquerdo.

Quick Sort

- Escolhido o elemento destacado
- Elementos menores que 5 à esquerda dele;
- Elementos maiores que 5 à direita dele;



Pronto

Quick Sort

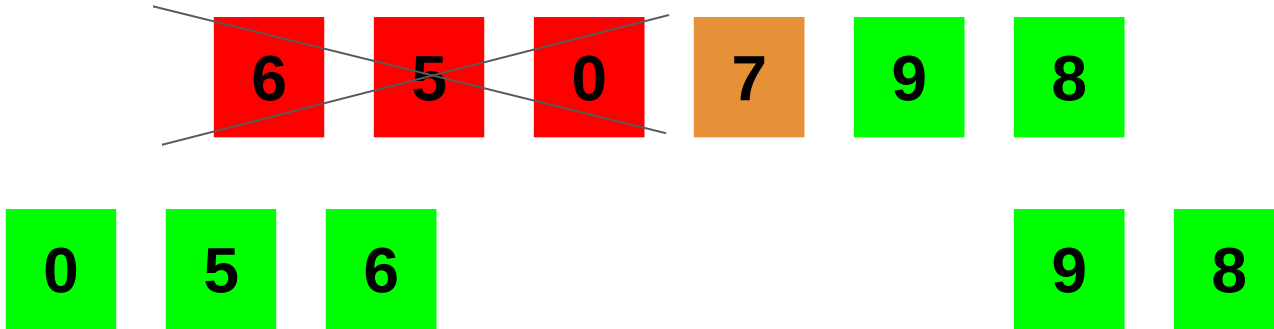
- Escolhido o elemento destacado
- Elementos menores que 5 à esquerda dele;
- Elementos maiores que 5 à direita dele;



Sub-array ordenado.

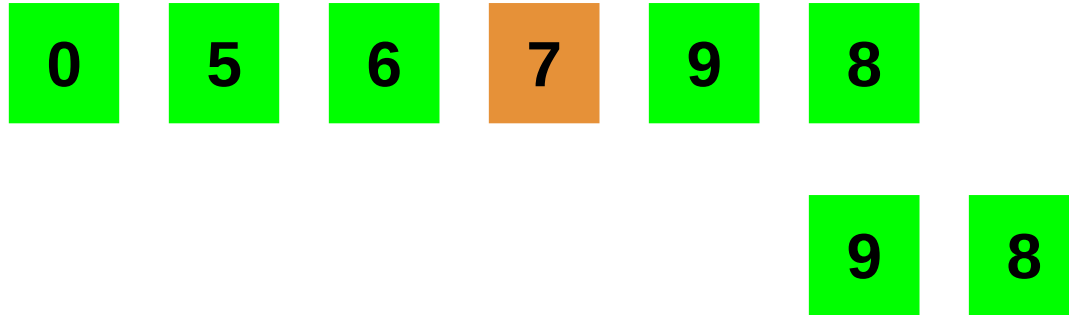
Quick Sort

- Lado izquierdo está OK!



Quick Sort

- Vamos tratar o lado direito agora...



Quick Sort

- Vamos escolher o pivô



Quick Sort

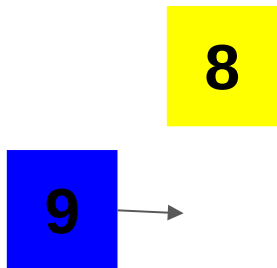
- Escolhido o elemento destacado
- Elementos menores que 8 à esquerda dele;
- Elementos maiores que 8 à direita dele;



9 > 8? Sim! Jogar para o lado direito.

Quick Sort

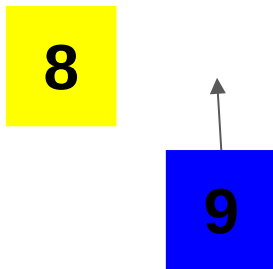
- Escolhido o elemento destacado
- Elementos menores que 8 à esquerda dele;
- Elementos maiores que 8 à direita dele;



9 > 8? Sim! Jogar para o lado direito.

Quick Sort

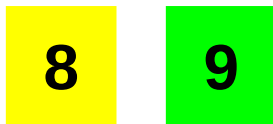
- Escolhido o elemento destacado
- Elementos menores que 8 à esquerda dele;
- Elementos maiores que 8 à direita dele;



9 > 8? Sim! Jogar para o lado direito.

Quick Sort

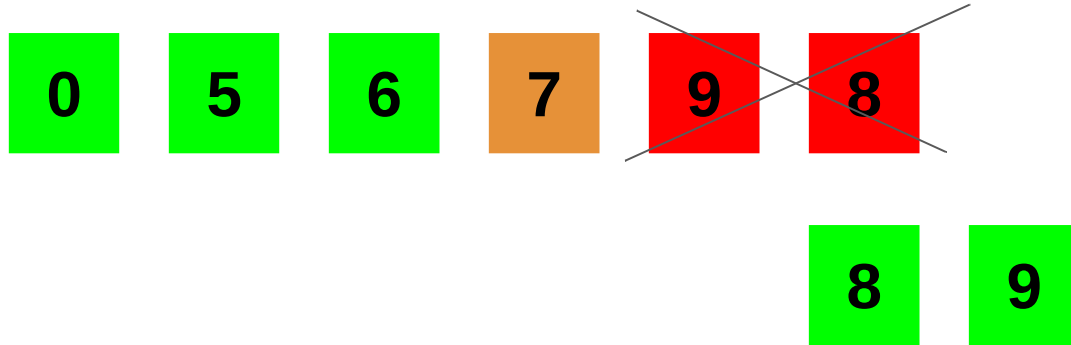
- Escolhido o elemento destacado
- Elementos menores que 8 à esquerda dele;
- Elementos maiores que 8 à direita dele;



Sub-array ordenado!

Quick Sort

- Lado direito Ok



Quick Sort

- Fim do Quicksort!



Quick Sort

- Complexidade:
 - Quicksort tem a mesma complexidade de para o caso médio, e o melhor caso: **$O(n \log n)$**
 - Para o pior caso ela chega a ser quadrática (**$O(n^2)$**)

Quick Sort

- **Eixo Horizontal:**
 - Tamanho do problema
- **Eixo Vertical:**
 - Número de Instruções

