

# Algoritmo de Ordenação: QuickSort

## Análise

Allex Lima   Paulo Moraes   Renan Barroncas   Daniel Bispo

Bacharelado em Engenharia da Computação  
Escola de Exatas  
Centro Universitário do Norte - UniNorte Laureate

Teoria dos Grafos e Computabilidade, 2016

# Sumário

## Conceituação

Como funciona o QuickSort

Formas de implementação

## Análise Assintótica

Estudo assintótico

Aplicações do algoritmo

# Sumário

## Conceituação

Como funciona o QuickSort

Formas de implementação

## Análise Assintótica

Estudo assintótico

Aplicações do algoritmo

# Prazer, sou o Quicksort

- ▶ Baseado no paradigma da **divisão e conquista**;
- ▶ Segmenta o vetor principal  $A[b \dots f]$  em dois arranjos:
  - ▶  $A[b \dots p-1]$
  - ▶  $A[p+1 \dots f]$
- ▶ Tais fragmentos são ordenados recursivamente;

# Prazer, sou o Quicksort

- ▶ Baseado no paradigma da **divisão e conquista**;
- ▶ Segmenta o vetor principal  $A[b \dots f]$  em dois arranjos:
  - ▶  $A[b \dots p-1]$
  - ▶  $A[p+1 \dots f]$
- ▶ Tais fragmentos são ordenados recursivamente;

# Prazer, sou o Quicksort

- ▶ Baseado no paradigma da **divisão e conquista**;
- ▶ Segmenta o vetor principal  $A[b \dots f]$  em dois arranjos:
  - ▶  $A[b \dots p-1]$
  - ▶  $A[p+1 \dots f]$
- ▶ Tais fragmentos são ordenados recursivamente;

# A divisão do vetor

- ▶ Necessita de um elemento **pivô** ( $p$ );
- ▶ Busca um ponto de segmentação tal que
  - ▶  $A[\text{begin} \dots p-1] < A[p] < A[p+1 \dots \text{end}]$

# A divisão do vetor

- ▶ Necessita de um elemento **pivô** ( $p$ );
- ▶ Busca um ponto de segmentação tal que
  - ▶  $A[\text{begin} .. p-1] < A[p] < A[p+1 .. \text{end}]$



# Sumário

## Conceituação

Como funciona o QuickSort

Formas de implementação

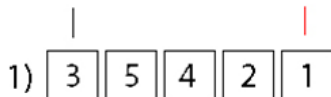
## Análise Assintótica

Estudo assintótico

Aplicações do algoritmo

# Implementação

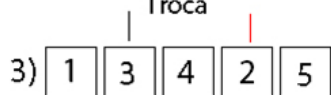
## Quick Sort



Troca



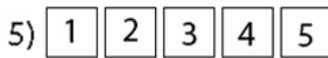
Troca



Troca



Troca



# Sumário

## Conceituação

Como funciona o QuickSort

Formas de implementação

## Análise Assintótica

Estudo assintótico

Aplicações do algoritmo

# Análise do Algoritmo

- ▶ O algoritmo consome tempo proporcional a  $n \log(n)$  em média, e  $n^2$  no pior caso;

	100	1.000	10.000
Ordem Crescente	0,000022	0,000311	0,008077
Ordem Decrescente	0,000017	0,000390	0,023088
Ordem Pseudoaleatória	0,000036	0,000281	0,008279

Table: Testes de execução - Dados expressos em segundos.

# Análise do Algoritmo

- ▶ O algoritmo consome tempo proporcional a  $n \log(n)$  em média, e  $n^2$  no pior caso;

	100	1.000	10.000
<b>Ordem Crescente</b>	0,000022	0,000311	0,008077
<b>Ordem Decrescente</b>	0,000017	0,000390	0,023088
<b>Ordem Pseudoaleatória</b>	0,000036	0,000281	0,008279

Table: Testes de execução - Dados expressos em segundos.

# Sumário

## Conceituação

Como funciona o QuickSort

Formas de implementação

## Análise Assintótica

Estudo assintótico

Aplicações do algoritmo

# Aplicações

- ▶ Banco de dados:
  - ▶ Chaves necessitam de ordenação;
  - ▶ Geralmente possuem altos números de registros, o que é recomendado para utilização do QuickSort;

# Aplicações

- ▶ Banco de dados:
  - ▶ Chaves necessitam de ordenação;
  - ▶ Geralmente possuem altos números de registros, o que é recomendado para utilização do QuickSort;