

# Entendendo Algoritmos

Nicolas Ramos Carreira

# Sumário

<b>0 Sobre o livro</b>	<b>2</b>
0.1 Panorama geral dos capítulos . . . . .	2
0.2 Como usar o livro . . . . .	3
0.3 Quem deve ler o livro . . . . .	3
<b>1 Introdução a algoritmos</b>	<b>4</b>
1.1 Introdução . . . . .	4
1.1.1 O que aprenderemos sobre desempenho . . . . .	4
1.1.2 O que aprenderemos sobre a solução de problemas . . . . .	4
1.2 Pesquisa binária . . . . .	4
1.2.1 Uma maneira melhor de buscar . . . . .	4
1.2.2 Tempo de execução . . . . .	4
1.3 Notação Big O . . . . .	4
1.3.1 Tempo de execução dos algoritmos cresce a taxas diferentes . . . . .	4
1.3.2 Vendo diferentes tempos de execução Big O . . . . .	4
1.3.3 A notação Big O estabelece o tempo de execução para a pior hipótese . . . . .	4
1.3.4 Alguns exemplos comuns de execução Big O . . . . .	4
1.3.5 O caixeiro-viajante . . . . .	4
1.4 Resumo do capítulo . . . . .	4
<b>2 Ordenação por seleção</b>	<b>5</b>

# Capítulo 0

## Sobre o livro

O livro, em seu inicio, já destaca qual a sua ideia central, que é: Ser um livro de fácil leitura, que irá pegar conteúdos complexos e simplificar através das explicações, exemplos, ilustrações e prática ao longo do livro. O livro deixa claro que não aborda todos os algoritmos existentes, mas sim os mais importantes e considerado úteis pelo autor.

### 0.1 Panorama geral dos capítulos

Os primeiros 3 capítulos do livro se constituirão no seguinte:

- Capítulo 1: Aprenderemos o nosso primeiro algoritmo básico, a busca binária. Além disso, veremos como analisar a velocidade de um algoritmo utilizando a notação Big-O (que será utilizada ao longo do livro inteiro)
- Capítulo 2: Aprenderemos duas estruturas de dados fundamentais, que são os arrays e listas encadeadas. Elas também são usadas na criação de estruturas de dados mais avançadas, como a tabela hash
- Capítulo 3: Aprenderemos o uso da recursão, uma técnica muito útil utilizada em muitos algoritmos

Os capítulos acima são os mais importantes (principalmente por conta da notação Big-O e da recursão), portanto, são os que o autor vai em ritmo mais lento.

O restante do livro apresenta alguns algoritmos de aplicação mais ampla. Veja:

- Técnicas para resolução de problemas: Abordadas nos capítulos 4, 8, 9. São abordadas técnicas, como divisão e conquista (cap 4), programação dinâmica (cap 9) e algoritmo guloso (cap 8) para problemas que não sabemos bem como resolvê-lo de forma eficiente.

- Tabela Hash: É uma estrutura de dados muito útil que é abordada no capítulo 5
- Algoritmos de grafos: Abordados nos capítulos 6 e 7, grafos são uma maneira de modelar uma rede. Veremos sobre a pesquisa em largura (cap 6) e algoritmo de Dijkstra (cap 7)
- K-vizinhos mais próximos: Abordado no capítulo 7, essa é uma técnica simples de aprendizado de máquina. Podemos utilizá-la para criar recomendações de sistema, mecanismo OCR e até um sistema para prever valores (ou seja, tudo que envolve prever um valor).
- Proximos passos: O capítulo 11 é discorrido sobre dez algoritmos que valem a pena uma leitura posterior (quando você já estiver craque em algoritmos)

## 0.2 Como usar o livro

O autor se preocupou bastante com a ordem com que os assuntos seriam abordados, sendo assim, o ideal é que se leia os capítulos em ordem (eles se baseiam uns nos outros).

Execute o código dos exemplos. Isso te fará reter melhor os conteúdos abordados. Você pode baixa-los no github através [DESTE LINK](#) (nesse repositório, o autor disponibilizou os códigos em várias linguagens, como C#, Python, Ruby..). Uma observação é que os exemplos abordados utilizam Python como linguagem.

Obviamente que é primordial que os exercícios passados sejam feitos. Eles nos ajudarão a conferir nosso pensamento (se estamos seguindo a linha de raciocínio correta ou não)

**OBS:** Um detalhe é que EU, Nicolas, gostaria de deixar registrado a forma de estudo que eu usei para estudar o livro. Além de fazer o que foi falado acima, para os conteúdos teóricos, irei ler, entender e depois passar por escrito aqui para o LATEX

## 0.3 Quem deve ler o livro

É destinado a qualquer um que queira aprender sobre programação e imergir no mundo dos algoritmos

# Capítulo 1

## Introdução a algoritmos

### 1.1 Introdução

- 1.1.1 O que aprenderemos sobre desempenho
- 1.1.2 O que aprenderemos sobre a solução de problemas

### 1.2 Pesquisa binária

- 1.2.1 Uma maneira melhor de buscar
- 1.2.2 Tempo de execução

### 1.3 Notação Big O

- 1.3.1 Tempo de execução dos algoritmos cresce a taxas diferentes
- 1.3.2 Vendo diferentes tempos de execução Big O
- 1.3.3 A notação Big O estabelece o tempo de execução para a pior hipótese
- 1.3.4 Alguns exemplos comuns de execução Big O
- 1.3.5 O caixeiro-viajante

### 1.4 Resumo do capítulo

# Capítulo 2

## Ordenação por seleção