Entendendo Algoritmos

Nicolas Ramos Carreira

Sumário

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 capitulos

Os primeiros 3 capítulos do livro se constituirão no seguinte:

- Capitulo 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)
- Capitulo 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
- Capitulo 3: Aprenderemos o uso da recursão, uma técnica muito útil utilizada em muitos algoritmos

Os capitulos 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:

- Tecnicas para resolução de problemas: Abordadas nos capítulos 4, 8, 9. São abordadas técnicas, como divisão e conquista (cap 4), programação dinamica (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 capitulo 5
- Algoritmos de grafos: Abordados nos capitulos 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 capitulo 7, essa é uma técnica simples de aprendizado de maquina. Podemos utiliza-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 capitulo 11 é discorrido sobre dez algoritmos que valem a pena uma leitura posterior (quando você ja 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 capitulos 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 exercicios passados sejam feitos. Eles nos ajudarão a conferir nosso pensamento (se estamos seguindo a linha de raciocinio 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

Neste capítulo, aprenderemos:

- Como fazer uma busca binária
- Entender o uso da notação Big-O para analisar a velocidade de algoritmos

1.1 Introdução

Um algoritmo é o conjunto de instruções para realizar determinada tarefa. Cada trecho de um código poderia ser um algoritmo, mas este livro se concentra nos mais importantes. Os algoritmos apresentados no livro foram escolhidos porque são rápidos ou porque resolver problemas interessantes.

Em cada um dos casos, o autor irá descrever o algoritmo e apresentar um exemplo. Em seguida, será falado sobre o tempo de execução do algoritmo em notação Big-O. Por fim, serão explorados outros casos de uso (problemas) onde há aplicabilidade daquele algoritmo

1.1.1 O que aprenderemos sobre desempenho

Aprenderemos, principalmente, a comparar o desempenho de diferentes algoritmos. Exemplo: "Para este caso, devemos usar quicksort ou mergesort. Devemos usar uma lista encadeada ou um array?"

1.1.2 O que aprenderemos sobre a solução de problemas

1.2 Pesquisa binária

Para entender, vamos a um exemplo: Suponhamos que tenhamos uma lista telefonica e queremos procurar um contato com a letra K, poderíamos começar foleando a lista até encontrar ou podemos começar do meio (ja que sabemos que

não estará no começo) e partir dali.

Outro exemplo interessante é o Facebook. Suponhamos que o Facebook quer verificar se determinada conta existe. O nome da conta é Nicolas. Ele irá procurar no banco de dados. Poderia até começar do A, mas faz mais sentido começar a busca pelo meio.

Todos os exemplos acima representam a aplicabilidade da busca binária. A busca binária é um algoritmo, onde passamos para ele uma lista de elementos (que deverá estar ordenada) e se o elemento buscado está na lista, será retornada a posição do elemento e caso contrário, será retornado None

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