Introdução a Algoritmos

Carlos Camarão
Universidade Federal de Minas Gerais
Doutor em Ciência da Computação pela Universidade de Manchester, Inglaterra

Prefácio

Este livro se propõe a introduzi-lo ao importante ramo da ciência da computação que trata do desenvolvimento e análise da eficiência de algoritmos. O assunto da análise de eficiência é comumemente chamada em computação de complexidade.

Essa análise aborda em geral quanto tempo é gasto na execução de um algoritmo em função do tamanho da entrada: diz-se complexidade de tempo do algoritmo. Além do tempo, pode ser analisada também a complexidade de espaço (quanto espaço é gasto na execução em função do tamanho da entrada).

... notação ...
... funcional ...
... clareza, concisão ...

Conteúdo e Organização do Livro

Este livro foi concebido para ser utilizado como texto didático em cursos introdutórios de programação, de nível universitário. O conteúdo do livro não pressupõe qualquer conhecimento ou experiência prévia do leitor em programação, ou na área de computação em geral, requerendo apenas conhecimentos básicos de matemática, usualmente abordados nos cursos de primeiro e segundo graus. O livro adota a linguagem de programação `C` (veja no anexo **??** uma discussão sobre essa escolha).

Como estudaremos daqui a pouco, a linguagem C é uma linguagem imperativa. Uma visão geral sobre os conceitos básicos da programação imperativa é apresentada inicialmente, com o objetivo de favorecer uma compreensão global sobre o significado e o propósito dos conceitos empregados nesse estilo de programação, facilitando assim um melhor entendimento da aplicação dos mesmos na construção de programas. Cada um desses conceitos é abordado mais detalhadamante em capítulos subsequentes, por meio de exemplos ilustrativos e exercícios.

Além desses conceitos básicos, o livro aborda também, de maneira introdutória, os seguintes tópicos adicionais: entrada e saída de dados em arquivos, e manipulação de estruturas de dados como arranjos e listas encadeadas.

Recursos Adicionais

Uma página na Internet associada a este livro pode ser encontrada no endereço:

`http://www.dcc.ufmg.br/~camarao/ipcc`

O texto desse livro e os códigos da maioria dos programas apresentados no livro como exemplos encontram-se disponíveis nesta página. Outros recursos disponíveis incluem sugestões de exercícios adicionais e projetos de programação, transparências para uso em cursos baseados neste livro e referências para outras páginas da Internet que contêm informações sobre a linguagem `C` ou sobre ferramentas para programação nessa linguagem.

Há diversas páginas na Web com informações sobre a linguagem `C` disponíveis na Web, assim como cursos sobre introdução a prgramação em `C`, dentre os quais citamos:

- `http://pt.wikibooks.org/wiki/Programar_em_C`: Páginas Wiki, criadas pela própria comunidade de usuários da Web, sobre programação em `C`. A versão em inglês pode ser encontrada em `http://en.wikipedia.org/wiki/C_(programming_language)`.

- `http://www.ead.cpdee.ufmg.br/cursos/C`: Material usado no curso de introdução a programação em `C` ministrado no Departamento de Engenharia Elétrica da UFMG.
- `http://cm.bell-labs.com/cm/cs/cbook/`: *The C Programming Language*, B. Kernighan & Dennis M. Ritchie, Prentice Hall, 1988.
- `http://publications.gbdirect.co.uk/c_book/`: Página com a versão gratuita da segunda edição do livro *The C Book*, de Mike Banahan, Declan Brady e Mark Doran, publicado pela Addison Wesley em 1991.
- `http://www.cyberdiem.com/vin/learn.html`: *Learn C/C++ today*, de V. Carpenter. Uma coleção de referências e tutoriais sobre as linguagens C e C++ disponíveis na Internet.
- `http://c-faq.com/index.html`: Perguntas frequentes sobre a linguagem `C`, e suas respostas (em inglês).
- `http://cm.bell-labs.com/cm/cs/who/dmr/chist.html`: "The Development of the C Language", Dennis M. Ritchie (Janeiro de 1993).
- `http://www.livinginternet.com/i/iw_unix_c.htm`: "History of the C Programming Language", Bill Stewart (Janeiro de 2000).
- `http://www.cs.ucr.edu/ nxiao/cs10/errors.htm`: "10 Common Programming Mistakes in C".

Livros adicionais sobre a linguagem `C` incluem:

- *A linguagem de programação padrão ANSI C*. B. Kernighan & D.C. Ritchie. Editora Campus, 1990.
- *C — completo e total*. H. Schildt. Editora McGraw-Hill, 1990.a
- *C: A Reference Manual*, Samuel P. Harbison & Guy L. Steele, 5ª edição, Prentice Hall, 2002.
- *C Programming: A Modern Approach*, K.N. King, Norton, 2008.

E divirta-se assistindo:

`http://www.youtube.com/watch?v=XHosLhPEN3k`

Pré-requisitos

Os pré-requsitos são:

1. Experiência inicial em programação de computadores: em particular, para entender definições de funções e de estruturas de dados recursivas simples, como listas e árvores.
2. Experiência inicial com provas por indução. (??)

Capítulo 1

Introdução

Vamos identificar neste livro algoritmo com função, no sentido de prover uma sequência de passos que associa a cada valor de (um conjunto de valores de) entrada um único valor de (um conjunto de valores de) saída.

A diferença que existe entre o conceito usual de função é a notação usualmente empregada para especificação da sequência de passos. Em computação, é usual o emprego de uma notação ou lingugem *imperativa*, ao passo que usualmente definições de funções empregam uma notação mais *declarativa*, ou *funcional*.

Por exemplo, considere o problema de ordenação, especificado formalmente como a seguir (um problema computacional especifica a relação que deve existir entre a entrada e a saída):

Entrada: sequência de elementos S_0.

SaÃda: sequência de elementos ordenada S tal que S é uma permutação de S_0.

Uma sequência $a_1, \ldots, a_n$ é ordenada se $a_i \leq a_{i+1}$ para $i = 1, \ldots, n-1$.

1.0.0.1 Sobre Permutação

Uma permutação (ou arranjo) é uma redisposição de (um conjunto ou sequência de) elementos em uma certa sequência (se contrapõe a uma *combinação*, na qual a ordem dos elementos resultantes não é relevante). Por exemplo, há 6 permutações distintas dos elementos 1,2,3, que são, escritas como tuplas: (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1).

Outro nome, usado no contexto de palavras, é *anagrama*.

Um anagrama é o resultado de rearranjar as letras de uma palavra ou frase para produzir uma nova palavra ou frase, usando todas as letras originais exatamente uma vez. Por exemplo, "ovo"pode ser rearranjado para "voo".

Permutações ocorrem em diversas áreas da matemática e proeminentemente no estudo de algoritmos, particularmente de ordenação, em ciência da computação.

O número de permutações de n elementos distintos é igual ao fatorial de n (usualmente escrito em matemática como $n!$), que é igual ao produto de todos os inteiros positivos menores ou iguais a n.

1.1 Ordenação por Inserção

Um algoritmo ou função que resolve o problema de ordenação especificado acima, chamado de *ordenação por inserção*, é mostrado a seguir. Ele reflete o modo como um jogador de baralho usualmente ordena uma sequência de cartas recebidas (por exemplo, em um jogo de buraco).

Neste livro, usamos sempre pseudo-códigos com notação funcional e imperativa para representar cada algoritmo. A notação funcional é a linguagem `Haskell` e a notação imperativa é um pseudo-código semelhante a C, Pascal ou Java.

`A notação funcional será explicada sempre que necessário, isto é, sempre que houver alguma possibilidade de dúvida. Uma descrição sucinta da linguagem Haskell é incluída`