

## はじめに

単純化していえば、問題を解く手順のことをアルゴリズムといいます。しかし、アルゴリズムだけでプログラムをつくることはできません。扱うデータの表現を変えると、アルゴリズムも変えざるを得ません。データの表現はデータ構造と呼ばれますが、データ構造はアルゴリズムと同様にプログラムの性能を決定する重要な要素です。すなわち、Wirth の名言にあるように、「アルゴリズム+データ構造=プログラム」なのです。どのようなデータ構造を採用し、どんなアルゴリズムを選ぶかで計算の速さが変わります。これまでに、多くの研究者によって多くのデータ構造とアルゴリズムが開発されています。これらを学ぶことによって、問題に適したアルゴリズムとデータ構造を選択する能力を身につけることができます。

アルゴリズムとデータ構造を表現するためには、必要な機能を備えているならば、どんな言語を用いても構いません。本書では、アルゴリズムの具体的な表現法として適宜プログラムコードも示しました。言語としては、現在広く普及している Java 言語を用いましたが、必要に応じて C 言語の例も示しました。なお、本書に掲載したいいくつかの Java プログラムは、Lafore の *Data Structures & Algorithms in Java* (1998) を参考にしています。これらのプログラム例を実際に行ってみることを勧めます。アルゴリズムやデータ構造の意味をよりよく理解できるようになると思います。また、アルゴリズムの性能は計算量で評価します。したがって、計算量を理解することは、アルゴリズムを理解する上で重要です。本書では、最初に計算量について説明します。また、データ構造で表現されたデータの探索や整列などの個々のアルゴリズムの解説でも、できるだけ計算量についてふれています。常に計算量を念頭に置いて本書を読み進んでください。

本書は、大学や高専、専門学校などで情報処理技術を学ぶ学生を対象に、基礎的なデータ構造とアルゴリズムを学習してもらうことを目的として書かれて

います。できるだけ図表を用いて概念を把握しやすいような解説を心がけました。アルゴリズムやデータ構造は、ノートやメモ用紙に図式化しながら学習することを勧めます。スタックやキューを表す図、連結リストや木構造の図、いろいろな整列法を説明した図などはただ眺めるだけでなく、実際に自分で考えながら描いてみるとよく理解できるものです。本書の図とまったく同じ図にする必要はありません。本質を理解し、自分なりの図を描けるようになれば、そのデータ構造やアルゴリズムは理解できたと思ってよいでしょう。整列のアルゴリズムは、トランプの数字カードを実際に並べ替えて理解するのもよいと思います。できるだけわかりやすい説明を心がけましたので、少し意欲のある方なら自学自習の教材としても読み進められると思います。

本書の構成は、次のようになっています。第1章では、データ構造とアルゴリズムの関係を概観してから、計算量について説明します。第2章では、配列、リスト、木構造などいくつかのデータ構造を取り上げます。第3章では、2分探索木、ハッシュ法など代表的な探索法について説明します。第4章では、代表的な整列法について説明します。

最後になりましたが、本書を執筆する機会を与えて下さった共立出版株式会社の加藤敏博氏と編集の過程でお世話になった石井徹也氏に感謝の意を表します。

2009年10月

著 者