

7章

アルゴリズムの基礎

アルゴリズムは、問題を解決するための処理手順である。

普通教科の「情報」では、あまり深入りしないで、整列や検索などの基本的なアルゴリズムを学習し、これを通してアルゴリズムの大切さを理解することが重要である。専門教科の「情報」では、将来、情報関連の産業に就く者が多いことを考え、システム設計、管理や運用に関する基礎的・基本的な知識・技術として、アルゴリズムとデータ構造に関する知識・技術を習得させる。

1. 目的とねらい

日常生活を考えると、朝目覚めてから、夜寝るまでの間に、さまざまな問題に遭遇し、それを解決するために処理手順を考えて行動している。たとえば食事の支度、着替え、洗濯、目的地までの道順と交通手段、などすべてである。これらに対してその場しのぎで行動したり、デタラメな順番で処理していたのでは、アルゴリズムにはならないだけでなく、目的が達成できないはずである。

コンピュータを使った処理も同様である。プログラミング言語をたくさん知っていても、文法を理解していなくても、問題を解決する手順、アルゴリズムができていないと実行することができない。

アルゴリズムにおいての基本的な要素は、効率と分かりやすさである。この2つの要素を取り入れて、アルゴリズムについて理解させ、実習を通して、解決すべき問題の内容に応じて、適切なアルゴリズムを選択する。また、実行の結果により、選択したアルゴリズムが適切であったかどうかを評価し、改善することの大切さを指導する。

2. アルゴリズムとは

アルゴリズムは、日本語で「算法」と訳されている。また、JISでは「明確に定義された有限個の規則の集まりで、有限回適用することにより問題を解くもの」と定義している。

アルゴリズムは、文章で表記したもの、流れ図(フローチャート)による表記、構造チャートによる表記、プログラミング言語による表記などいくつかの表記方法がある。一般的には、アルゴリズムは、流れ図で表記する。

文章で表記したものは、「であればしなさい。でなければ××しなさい」とか「をが×になるまで続けなさい」とか、そのまま話し言葉で表現できる。プログラミング言語は、これらを表現するために、独特な用語(予約語や変数の決まり)や文章(スクリプト)のルールを決めて、文法を作っている。

2.1 流れ図

アルゴリズムを表現するための流れ図で使用する記号は、JIS規格で制定されている。実際には、この規格の記号を使用するが、一部企業ごとに特別な記号を使用する場合がある。

表7.1 流れ図の記号

| 記号 | 名称 | 意味 |
|---|----------|--|
|  | データ | 媒体を指定しないデータ。または、それらからのデータ入力や出力。 |
|  | 直接アクセス記憶 | 磁気ディスクやフロッピーディスクなどに記憶されている直接アクセス可能なデータ。 |
|  | 書類 | 人間が読める媒体上のデータ。プリンタ出力帳票など。 |
|  | 表示 | ディスプレイなどの表示装置に表示されるデータ。 |
|  | 処理 | 演算、転送など任意の種類処理機能。 |
|  | 定義済み処理 | サブルーチンやモジュールなど別の場所で定義された処理。 |
|  | 準備 | 初期値設定など、その後の動作のための準備を行う。 |
|  | 判断 | 1つの入口といくつかの出口を持ち、記号の中に記述された条件にしたがって、唯一の出口を選択する。 |
|  | ループ端 | 繰り返し処理の始めと終わりを示し、それぞれに同じ名前を記述する。初期値、増分値、終了条件などを記述する。 |
|  | 線 | データや制御の流れを示す。流れの向きを明示するときは矢印をつける。 |
|  | 結合子 | 流れ図の別の場所への出力、または別の場所からの入口。対応する結合子には同じ名前をつける。 |
|  | 端子 | プログラムの開始、終了、または外部からの入口、出口を示す。 |

2.2 基本的なアルゴリズム

アルゴリズムは、いくつかの論理構造の組み合わせで構成される。問題解決の処理手順を考えるときに、複雑な論理構造をできるだけ簡単な論理構造に分解していき、単純で小さな単位の処理に落として整然とした論理構造にすることが重要である。こうした考え方は、プログラミングだけでなく、文章作成などについても同じである。ここで、基本的な論理構造について説明する。

(1) 順次構造

上から下へ順番に処理を実行する。プログラムの基本となる。



図7.1 順次構造の流れ図

(2) 選択構造

条件によって、処理内容を振り分ける。二分岐選択 (if ~ then ~ else) と多岐選択 (case) がある。

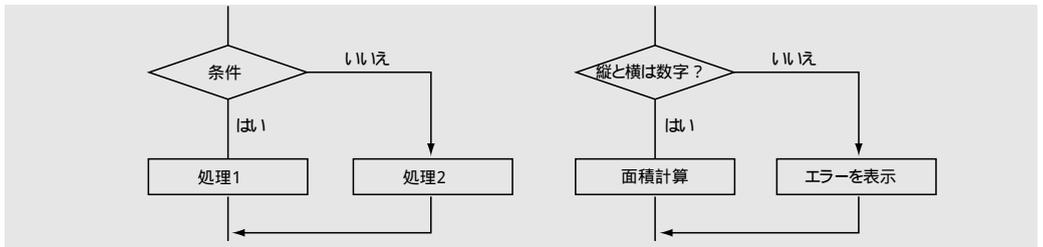


図7.2 選択構造の流れ図(二分岐選択)

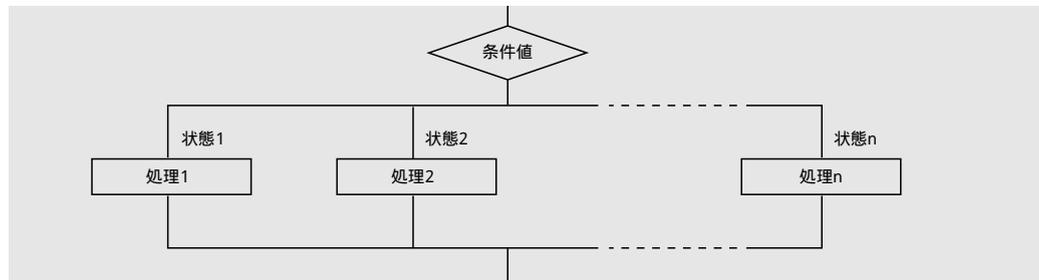


図7.3 選択構造の流れ図(多岐選択)

(3) 繰り返し構造

条件が成立している間は処理を繰り返し、条件が成立しなくなったら、次の処理に移る。条件の設定方法によって、while型やfor型などがある。

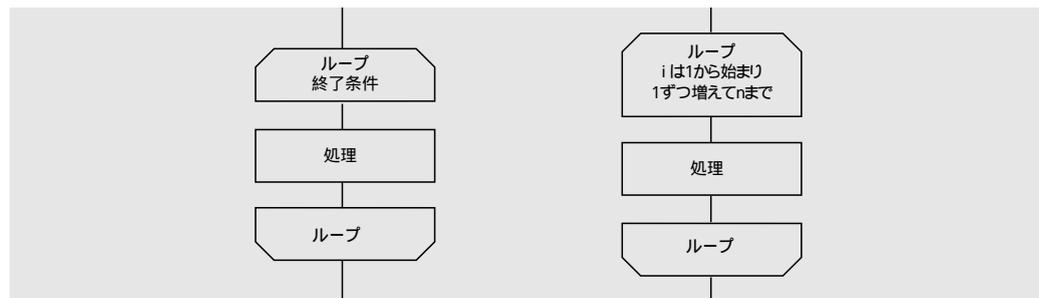


図7.4 繰り返し構造の流れ図

2.3 データ構造

アルゴリズムを考える上で必要なものが、データ構造である。データ構造とは、コンピュータで扱うことができるようにしたデータの集まりのことである。最も基本的なものとして、レコードと配列がある。

(1)レコード

データを処理するときの1件分で、処理単位のこと。たとえば、名簿(氏名、住所、電話番号)や成績(氏名、中間テストの点数、期末テストの点数)の1人分にあたる。レコードは、データとして意味を持つ最小単位である「項目」で構成される。また、レコードが複数件集まって、1つの仕事の単位「ファイル」を構成する。プログラミングでは、レコード単位で処理する場合が多い。

(2)配列

同じ基本データ型で、同じ大きさからなるデータ構造のこと。配列のデータは、配列名と添字でそのデータを参照する。数値処理などのプログラミングで使用されることがある。

3. 指導のポイント

アルゴリズムの実習として、代表的なアルゴリズムを例にとり、文章で記述したり、流れ図を作成する。生徒の実態や興味、コンピュータ教室の環境に応じて、適切な情報技術を活用する。プログラミング言語がないとか、プログラミングに興味がないようであれば、パソコンのアプリケーションソフトの中で、表計算ソフトなどを活用して、マクロを作成したり、判断をする関数の使用を指導する。

【演習例】

- [1] 集計処理：繰り返し構造の例。レコードの中の指定された項目について、すべての件数が終わるまで集計する。
- [2] 最大値、最小値：二分岐選択構造の例。複数のレコードの中から、指定された項目の最大値を持つレコードと、最小値を持つレコードを探す。
- [3] 整列(並び替え)：繰り返し構造と二分岐選択構造の組み合わせの例。データを指定された項目について、レコードを一定の順番に並び替える。
- [4] 検索：配列構造のデータを使用した例。配列から、条件に合致するデータを探し出す。

また、アルゴリズムは、プログラミング以外、情報教科以外でも使用できるので、さまざまな教室外の活動や行事などの問題解決において応用する。