

## アルゴリズム的思考法の教育

飯田 周作

専修大学 ネットワーク情報学部

飯田 千代

専修大学 ネットワーク情報学部

清藤武暢

横浜国立大学 環境情報研究院

佐藤 創

専修大学 ネットワーク情報学部

### 概要

アルゴリズム教育は、情報系の学部における重要なトピックスである。しかし、同時に学生の理解に大きな個人差が見られ、ミニマムエッセンシャルズとしてのアルゴリズム教育とは何かという難しい問いに直面している。本報告では、専修大学ネットワーク情報学部における「アルゴリズム的思考法」という科目で行ってきた教育を分析することにより、アルゴリズム的思考法を育成するための教育方法を提案する。

## A Method for Educating Algorithmic Thinking Style

Shusaku Iida

School of Network and Information,  
Senshu University

Chiyo Iida

School of Network and Information,  
Senshu University

Takefumi Seito

Graduate School of Environment and  
Information Sciences, Yokohama  
National University

Hajime Sato

School of Network and Information,  
Senshu University

### abstract

At departments of Universities related to computer or information technologies, teaching algorithm is still an important topic. However, there are big differences in the levels of understanding among the students. So, it is important to know what is the essential and minimum part in teaching algorithm. In this article, we analyze student behavior in a subject called "Algorithmic Thinking Style" and propose a new teaching method for algorithm education.

## 1. はじめに

本報告では、我々が大学の学部で実践してきたアルゴリズム教育から得られた知見を基に、アルゴリズムの教育法を提案する。本研究の背景には、大学におけるアルゴリズム教育の経験を、高等学校におけるアルゴリズム教育に生かすことができればという希望がある。最終的には、高等学校におけるアルゴリズム教育のミニマムエッセンシャルズの提案をすることが、この研究の目的である。最終的な研究目的である「高等学校におけるアルゴリズム教育のミニマムエッセンシャルズの提案」は、本年度より開始し 3 年間をかけて一定の結論を得る予定である。本報告は、そのための基礎的な議論を提供する。

専修大学ネットワーク情報学部は、文理融合型の情報系学部として 2001 年 4 月にスタートした。ネットワークシステムコース、コンテンツデザインコース、情報戦略コース(当時は情報ストラテジーコース)の 3 コース構成(現在は情報技術創造コースを加えて 4 コース構成)である。これらのコース名から想像される通り、本学部では情報に関する幅広い内容を対象としており、必修科目を設定することは容易ではない。その中で、我々は 1 年次前期に「アルゴリズム的思考法」を必修科目として配置している。

我々は、アルゴリズムそのものを教育することと、アルゴリズム的思考法を教育することをわけて考えている。情報を扱う上でアルゴリズムは必須の概念である。ただし、状況によって必要とされる理解の広さと深さは大きく異なる。どのような状況でも共通しているのは、アルゴリズムを設計することやそれを使うことの意味とその思考方法である。我々は、これこそがアルゴリズム教育のミニマムエッセンシャルズであると考え

る。

本報告の構成は以下の通りである。2 章で、我々の考えるアルゴリズム的思考法の定義を与える。3 章で、「アルゴリズム的思考法」の講義で実施したテスト結果を分析し、現状と問題点について議論する。4 章で、これまでの議論に基づいて新たな教育法の提案を行う。

## 2. アルゴリズム的思考法とは

アルゴリズム的思考法とは、問題を解決する際に指針とすべき考え方の 1 つである。具体的には、基本的な操作に分解しそれらの順序を意識する思考法である。このような思考法は、実はあまり日常的には行われ(というよりは 10 代の学生の日常生活にはあまりないと言うべきだろう)。また、行われていたとしても、意識的には行っていないと言える。

アルゴリズム的思考法の教育では、アルゴリズムを知識として教授することを目的とするのではなく、どう考えて、どう表現するかという点が重要となる。

## 2.1. プログラミングとアルゴリズム教育

大学や専門学校、高等専門学校等におけるアルゴリズム教育は、主としてプログラミングの基礎として行われてきたと考えられる。よって、その教育方法は、コンピュータを用いたプログラミングとセットになっていることが多い。これは、ほとんどのアルゴリズムの教科書がコンピュータ言語を用いた説明を採用している[1]ことから推測することができる。これは実に自然なことで、コンピュータによる問題解決の方法がもたらばプログラミングに拠っていた時代が長く続き、基本的なアルゴリズムを習得していることが実用的に重要であったからである。

プログラミングのためのアルゴリズムを理解するためには、コンピュータの仕組みを理解している必要がある。そのため、通常ノイマン型のコンピュータの仕組み、およびプログラミング言語の文法を説明して、いよいよアルゴリズムの説明という順番で教育が行われる。これは、大学・専門学校・高等専門学校等で、プログラミングが重要な位置を占める学部・学科においては全く問題ないカリキュラムであると言える。しかし、我々の学部のように、情報についてより広い範囲を対象とし、プログラミングの比重が相対的に重くない学部においては、あまり適していない。

## 2.2. コンピュータ言語を用いないアルゴリズム教育

本学部の「アルゴリズム的思考法」は、プログラミングとは独立に行われる。そのため、その表記法には、コンピュータ言語ではなくフローチャートを用いる。これには 2 つの意味がある。1 つは、よりいい加減であることである。コンピュータ言語処理系を使うと、どうしても文法への正確な適合が求められる。一方、紙上にペンで記述するフローチャートは、多少の文法の揺れに対して人間が理解できる範囲を限界として許容される。もう 1 つは、機械的な実行が出来ない点である。機械的に実行ができると、どうしてもその実行に頼ってアルゴリズムを設計したり理解したりしようとする。フローチャートの場合には、頭での理解がより求められる。

## 3. 現状と問題点

2007 年度のアルゴリズム的思考法の講義の中で実際に行ったテストを例にとり「アルゴリズム的思考法」という科目で現在我々が直面している現状とその問題点を考察する。例として扱う問題は図 1 に示す通りである。テストは全 7 問から構成され、この問題は 6 問目の問題として出題された。白紙のままの解答も多く見られたが、それらも合わせて全体での正解率は約 1 割に留まった。

この問題に対する答えはいくつか考えられるが、最もシンプルな解答としては図 2 のように、2, 0, 0, 7 の各数字を配列の要素に代入する操作を一行に並べ、代入操作を行うたびにその時の配列の添え字が入力  $n$  に達していないかどうか検査をするものである。

講義のテストでは他にも様々な問題を扱ったが、この問題を本報告の題材として取り上げる理由は、1)日本語で説明された仕様に対してフローチャートで記述されるアルゴリズムは簡潔であり、アルゴリズム教育の問題として優れていること、2)講義中に扱ったどのアルゴリズムにも酷似していないため、暗記では対応できないこと、が挙げられる。これらの理由は、学生の根本的な理解を見る上で重要である。

問) 以下の仕様を満たすフローチャートを描きなさい。

【仕様】

正の整数 $n$ を入力として受けつけ、 $x_0$ から $x_n$ までの各値が、

$$2, 0, 0, 7, 2, 0, 0, 7, 2, 0, \dots$$

と2、0、0、7がくり返し並ぶように配列 $x$ を設定し、出力する。

例)  $n=5$ の時、配列 $x$ の各要素は、

$$x_0=2, x_1=0, x_2=0, x_3=7, x_4=2, x_5=0$$

となる。

(図 1 テスト問題)

この問題に対する解答を採点してみると、学生をその解答のパターンによって次に示す7つのタイプに分類できることがわかった。

1. 正解
2. 単純ミス
3. 繰り返しの理解不足
4. 配列の理解不足
5. 問題の読解ミス
6. 混乱
7. 白紙

タイプ1に属する正解した学生の解答には、図2に示した解答例のような素朴な解答や、よりテクニカルな手法を用いた解答等のバリエーションがあり、いくつかのサブタイプに分類し議論する余地がある。しかし、本研究では「アルゴリズム的思考法を理解する上でのつまずきのポイントはどこか」に焦点をあてるため、タイプ1の学生は対象外とする。また、タイプ7も議論の余地がないという理由から対象外とする。

タイプ 2 の単純ミスをしているグループは、教育をする上での問題はないと言ってよい。ここでいう単純ミスとは、例えば繰り返しの条件中に現れる大小比較の記号を左右逆に書いたり、 $<$ と $\leq$ を取り違えたりするといったミスである。これらの学生の中には、プログラミング言語及びその開発環境（コンパイラ等）を用いてじっくり問題に取り組めば、自らミスに気づき、それを修整し、正解にたどり着ける者が多いことが予想される。このタイプの学生には、フローチャートによるアルゴリズムの教育と同時に、従来のプログラミング教育のようなコンピュータを用いた教育を行うことが効果的だと思われる。

タイプ 3 は、処理の流れはおおよそ記述できているが、例えば本来ならば繰り返しに入れない初期化の処理を繰り返し中に記述してしまったり、逆に繰り返しに入れなければならないカウンターのインクリメント処理が抜けていたり、またはそれらのミスが複合したフローチャートを記述しているグループである。このようなフローチャートを記述してしまう学生には、アルゴリズムの理解を暗記に頼っている者が多いように思われる。つまり、講義の中で例題として用

いたフローチャートの表面的な形状を暗記し、それを応用して問題を解こうとしているのである。そのため、例えば最大値を求めるアルゴリズムのフローチャートを応用して最小値を求めるフローチャートを記述するなど、例題に酷似した問題に対してはなんとか対処することができるが、フローチャートを構成する個々のパーツにまで理解が及ばないため、ここで例に挙げた問題のように、非常に単純ではあるが特殊な問題には対応できない。また、処理の構成パーツはあっているのに記述している順番が間違っている、分岐したフロー（矢印）の合流先がずれているなど、アルゴリズムの流れを理解してい

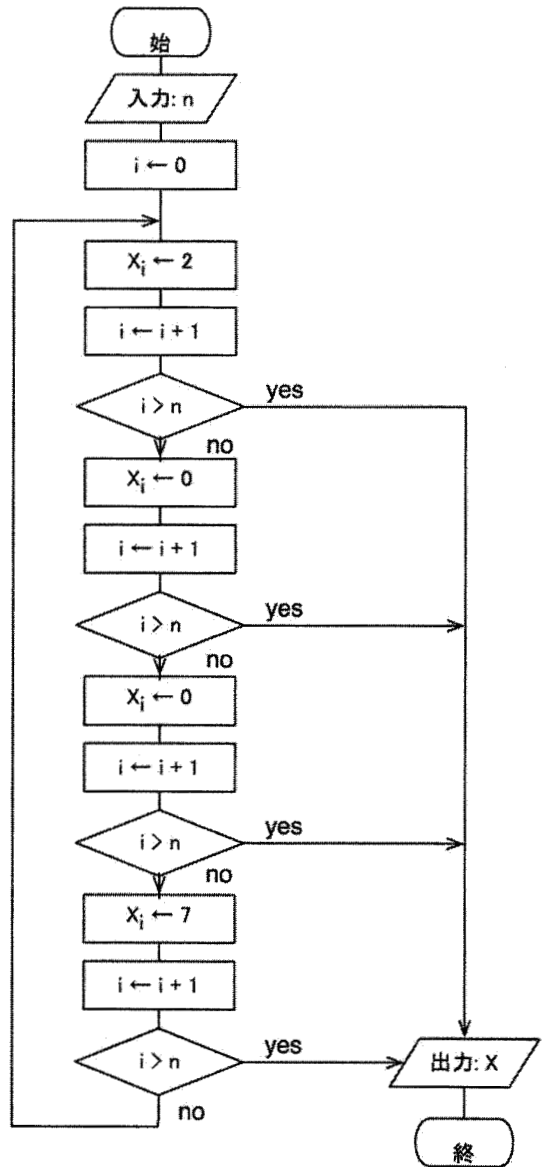


図 2. 解答例

ればありえないようなミスや記憶違いによりおこしてしまう。このタイプの学生は、従来のプログラミング教育のように、言語のシンタックスを解説し、例題を示し、問題を与えて解かせることの繰り返しでは教育効果が薄いと考えられる。これらの学生には、アルゴリズムに関してただ定義を与えるだけでなく、本報告の4章で提案するような「アルゴリズムとはどのようなものか」ということを体感できるような教育を通して、部分と全体、局所と大局ということを理解させることが重要と考える。

タイプ4の学生は、配列や添え字付き変数に対する直感的な理解が不足しているといえる。おそらく、彼らは高校数学の数列等に苦手意識をもっており、 $x_0$ ,  $x_1$ のような表記になじみが薄い。さらに、 $x_i$ ,  $x_j$ のように添え字自体が変数となってしまうと、それらが何を意味するか理解できなくなってしまう学生が多い。中には $x$ のような単純な変数に対する理解すら不足していると思われる学生も存在する。このタイプの学生には、アルゴリズム教育を行う以前に基本的な数学の記法を再教育し、苦手意識を取り除く必要があるだろう。

タイプ5は、何らかの仕事やフローチャートを記述しているが、それが問題の趣旨にあってないグループである。後述するタイプ6との決定的な違いは、記述されたフローチャートが何をしているか理解可能などである。つまり、問題の意図するものとは異なるが、フローチャートとしては理解可能なものを書いているのである。このグループに属する学生には、単純に問題を読み間違えた者と、そもそも読解力に問題がある学生が混在している可能性がある。両者を区別するためには、さらにいくつかの問題を解かせてみればよい。その結果、後者と分類された学生には、タイプ4の学生同様、アルゴリズム教育よりも基本的な日本語の読解力を養う訓練が必要となる。

タイプ6の学生は、例えば配列に入力を受け付けるような処理を記述したり、非常にたくさんの分岐を記述したりするなど、何かの指針を持ってフローチャートを記述しているとは考えにくいグループである。このグループは、便宜上上人グループとはなっているが、おそらく様々な原因をもつ学生が混在していると予想される。たまたま時間がなく非常に中途半端な状態で終了時間となってしまった学生もいるかもしれないし、そうではなく時間が十分に与えられたとしても、やはり支離滅裂な解答を記述する学生もいるかもしれない。よって、このグループは、他のグループと違ってある特徴をもった学生の集まりではなく、「その他」もしくは「未分類」と考えるべきである。

### 3.1. 問題点の分析

本科目では、表現形式の言語固有の文法にとらわれて本質的なアルゴリズムの議論が十分に行えないことが内容に、プログラミング言語より柔軟な記述が可能であるフローチャートを用いて講義を進めた。しかし、それでもアルゴリズムの話をする準備段階として、フローチャートの文法を説明することが避けられず、それらの問題を完全に解決



するには至らなかった。また、初回の講義でアルゴリズムの定義[2]を紹介したものの、簡潔にまとめられた定義は、はじめて「アルゴリズム」という言葉を耳にする学生にとっては実感の得られるものではなかったと言える。特に上記のタイプ 3 に分類されるような学生は、問題を暗記によって解決しようとする意識が強いため、言葉により表面的な定義は、むしろ理解の妨げになると言えるだろう。

次章では、これらの問題点に対する解決策を提案する。

## 4. 提案

3 章で述べたように、アルゴリズムを記述する表記法でつまづく学生を減らすことが出来れば、かなりの学生をより効率良く教育することが出来ると思われる。そのためには、特定の表記法を導入する前に、アルゴリズム的思考法を体験させることが重要である。また、例題として取り上げるアルゴリズムは、コンピュータのアーキテクチャや数学的な知識を前提としないものを選ぶ必要がある。

### 4.1. 体験による教育

アルゴリズムを体験的に学ばせるにための方法として、以下の 2 つを提案する。

1. 教員が記述した指令書通りに学生がアルゴリズムを実演する
2. ペアを組ませて、一方が記述した指令書通りに他方が実演する

指令書は、自然言語の箇条書きが適当と思われる。この体験により以下のことを学生に意識させる。

- ◆ 基本的な操作とは何か(何が最小の単位なのか)
- ◆ 指令書に記述されていることしか行動してはいけないということ(ルールの遵守)
- ◆ 指令書を正確に記述することは大変だということ(表記法の重要性)

この方法によって、表現形式に依存することなくアルゴリズム的思考法とは何か、またなぜそれが重要であるのかにつき、理解得ることができると考える。さらには、基本操作への分割とその整列、局所と大局という視点の重要性が認識される。

具体的な方法として、数が書かれたカードを使った方法等が考えられる。

### 4.2. 例題の選択

例題としては、コンピュータのアーキテクチャや数学的な知識を前提としないアルゴリズムを選択すべきである。これは、特に学生のバックグラウンドが多彩であればあるほど重要である。

### 4.3. 日常生活とアルゴリズムの連結

編み物は、日常生活におけるアルゴリズム的思考法を考える上で良い例となるだろう。

編み物は、一見非常に複雑な作業である。しかし、よく分析するとやるべきことは幾つかの単純な作業に分解される。単純化して言えば、1つの目を編むことと、それを何回繰り返すかで表現される。作業になれた人は、他人とおしゃべりをしながらでも編むことが出来る。この例からアルゴリズムの効用を説明することができる。

## 5. 終わりに

専修大学ネットワーク情報学部で行われている、「アルゴリズム的思考法」という科目では、情報を学ぶ際の基礎となるアルゴリズム教育を目指して教育を行っている。情報という広い範囲をカバーするアルゴリズム教育のミニマムエッセンシャルズとして、「基本的な操作に分解しそれらの順序を意識する」というアルゴリズム的思考法が特に重要であると考えている。

本報告では、アルゴリズム的思考法の教育には、体験による教育と正しい例題選択が重要であるという提案を行った。この提案は、これまでの我々の教育の分析と反省から導かれた仮説であり、具体的な教育効果についてはさらなる研究を必要とする。なお、体験による「科学的思考の実践」を高等学校の情報Bに取り入れた事例として[3]があり、今回の提案はその「アルゴリズム的思考法」版とも言える。

今後の課題として、体験による教育のための教材開発を計画している。

## 謝辞

本報告執筆に当たっては、山口裕之氏(専修大学経営学)、横山敬之氏、三枝亜由美氏、五十嵐友起氏、土屋絵梨氏(専修大学ネットワーク情報学)にご協力いただきました。

本研究は、平成19年度文部科学省科学研究費補助金(課題番号19300285)「高等学校情報化における科学的ミニマムエッセンシャルズのための教育プログラムの開発」の助成を受けています。

## 参考文献

- [1] R. セジウィック:「アルゴリズム C・新版」, 野下浩平, 星守, 佐藤創, 田口東訳, 近代科学社, 2004.
- [2] Donald E. Knuth, "The Art of Computer Programming", Volume 1, "Fundamental Algorithms", second printing, Addison-Wesley, 1969.
- [3] 今野篤志, 香山瑞恵:「コンピュータの特性理解に向けた情報の科学的思考を実践する指導事例」, 第3回日本情報教育開発協議会第3回全国大会 講演論文集, pp.43-46, 2007.