

アルゴリズム教育再考

浅野考平^{†1} 森戸隆文^{†2}

情報科学の専門教育の中で、アルゴリズムの教育は重要な位置を占めており、その基礎的な部分は「データ構造とアルゴリズム」という科目で教えられている。本報告では、この科目の問題点を検討し、教育内容を改善するための提案を行う。ただし、まだ実践による検証は行っておらず、報告は、中間的なものである。

Reconsidering Algorithm Education

KOUHEI ASANO^{†1} TAKAFUMI MORITO^{†2}

The teaching of algorithms holds a pivotal place in professional informatics education. The fundamentals are currently taught in a course called "Data Structures and Algorithms". This study explores issues related to this particular course and offers proposals for improving the content that has been instructed. This study should be regarded more as an interim report at this stage because the proposals still have to be verified with practical cases.

1. アルゴリズム教育の重要性

アルゴリズムは、プログラムにおける計算の手順のことである。コンピュータはプログラムによって動作し、プログラムを作成するときは、まず、アルゴリズムを設計して、それに基づいてプログラムを作成する。したがって、アルゴリズムを設計することはコンピュータを利用する過程において重要な位置を占めている。

教育においても重要である。実際、専門分野として情報科学が成立して以来、その教育課程においてアルゴリズムは、一貫して重要な一部分であった。また、専門教育に限らず、非専門的な情報科学教育においても同様である。例えば、「日本の情報教育・情報処理教育に関する提言2005（2005.10.29情報処理学会情報処理教育委員会）では、「手続き的自動処理の構築」を体験させることの重要性を指摘しているが、手続き的自動処理は次のように定義されている [1].

(1) 問題を自らの判断に基づき定式化し、その解決方法を考える。

(2) 解決方法を、アルゴリズムとして組み上げ、自動処理可能な一定形式で記述した、コンピュータ上で実行可能なものとして実現する。

(3) 実現したものが問題解決として適切であるかを検証し、必要ならば問題の定式化まで戻ってやり直す。

情報科学の専門教育の中で、アルゴリズムの基礎的な部

分は「データ構造とアルゴリズム」という科目で教えられている。本報告では、この科目の問題点を検討し、教育内容を改善するための提案を行う。ただし、まだ実践による検証は行っておらず、報告は、中間的なものである。

2. 「データ構造とアルゴリズム」の教育内容の評価と課題

情報科学の専門教育の中で、アルゴリズムの基礎的な部分は「データ構造とアルゴリズム」という科目で教えられており、内容は、古くから確立されている。例えば、典型的な教科書の内容は以下の通りである。

- ① 時間計算量の定義、オーダーによる分類
- ② データ構造として、リスト、スタック、キュー、グラフ、木、ヒープ
- ③ ソートのアルゴリズム、選択ソート、挿入ソート、バブルソート、クイックソート、基数ソート
- ④ 探索のアルゴリズム、二分探索、二分探索木、ハッシュ法

また、「基本情報技術者試験」のアルゴリズムのシラバスでも、以下のように記述されている[2].

目標：

- ・ アルゴリズム、流れ図の基本的な考え方、表現方法を修得し、適用する。
- ・ 代表的なアルゴリズムの基本を修得し、適用する。
- ・ アルゴリズムの基本的な設計方法を修得し、適用する。

^{†1} 関西学院大学理工学部
School of Science and Technology, Kwansai Gakuin University
^{†2} 関西学院大学大学院理工学研究科
Graduate School of Science and Technology, Kwansai Gakuin University

(1) 流れ図

アルゴリズムや流れ図(フローチャート)の考え方, 記号, 順次, 判定, 繰返しなど処理手順の表現方法を理解する。

用語例: 端子, 処理, 定義済み処理, 判断, ループ端, データ, 線

(2) 代表的なアルゴリズム

① 整列・併合・探索のアルゴリズム 整列のアルゴリズム, 併合のアルゴリズム, 探索のアルゴリズムの基本的な設計方法を理解する。

② 用語例: 選択ソート, バブルソート, マージソート, 挿入ソート, シェルソート

これらを観察すると, 現状の「データ構造とアルゴリズム」の内容は, 大規模データを処理するための代表的アルゴリズムを理解することが中心となっていることがわかる。

最も望ましいアルゴリズム教育の成果は, 学習者がアルゴリズムを自ら設計することができる能力を身に付けることである。次に, この観点から, 上記の「データ構造とアルゴリズム」の教育内容の評価を試みる。

アルゴリズム設計の能力についての典型的な考え方は「アルゴリズム設計マニュアル」の序文に記述されているので, その部分を抜き書きする[3]。

現実の問題に対するアルゴリズムで, 正しく, 効率がよく, しかも実際のプログラムとして実装できるものを設計するには次の2つの知識を必要とする。

技法: 優秀なアルゴリズム設計者は, データ構造, 動的計画法, 深さ優先探索, バックトラック法, それにヒューリスティックな方法などのアルゴリズムの基本設計技法を理解している。おそらく, 最も重要な設計技法はモデル化である。これは, 煩雑な現実世界の応用問題を, アルゴリズムの対象として適切である明確な問題へ抽象化する技法のことである。

資源 (リソース): 優秀なアルゴリズム設計者は, 巨人の肩の上で仕事をする。彼らは, それぞれの仕事のために新しいアルゴリズムを最初から作ることはせず, むしろ問題に関して何が知られているかについて見当をつけることができる。彼らは多数の古典的アルゴリズム問題に精通していて, それらがたいのい応用問題をモデル化するのに十分な材料を提供してくれる。

まとめると, (1) 基本的設計技法, (2) モデル化の技法, (3) 古典的アルゴリズムの理解という3つの知識を用いることによってアルゴリズムを設計することができる。実際, 「データ構造とアルゴリズム」では古典的アルゴリズムが多数, 教えられている。また, ここに述べられている基本的設計技法の説明が付け加えられている

ことも少なくない。一方, モデル化の技法については, おそらく, 対象となる問題の特殊性に依存するので, アルゴリズムそのものを教える教育の対象外であると考えられている。

ここに述べられている考え方は, 少なくとも情報科学の基礎, あるいは, 入門の部分でのアルゴリズムの教育では機能しないと考える。既にあるアルゴリズムや技法を受動的に学習することにならざるをえないからである。ここで, 基本的設計技法とされる動的計画法, 深さ優先探索, バックトラック法, それにヒューリスティックな方法などを知ったとしても, いくつかのアルゴリズムがその技法に基づいていることを理解するぐらいが限界であろう。また, 多数の, 代表的な古典的アルゴリズムを理解しても, それらを材料として組み立てるようなアルゴリズムは, 実際にはほとんど存在しないだろう。

教えられたアルゴリズムも, 実際に使う場面を想像できないのではないのか。例えば, なぜ, 多数のソートアルゴリズムを学ぶのか, という疑問にも答えなければならない。クイックソート以外は, 引き立て役に過ぎないとしか考えられない。クイックソートにしても, ライブラリの中の関数を利用すれば, 自分でアルゴリズムを理解する必要も, プログラムを書く必要もない。実際に使う可能性を実感できないまま, 既にあるアルゴリズムを, 与えられたインスタンス(問題例)に適用することを学ぶだけでは, 学習者がおもしろくない。アルゴリズムの実習科目でプログラムを作成させると, ネット上に“落ちている”プログラムをコピーして終わる, ということになりかねない。

3. 問題解決とアルゴリズム発見のモデル

アルゴリズムは, 問題を解決する手順を定めたルールであるから, アルゴリズムを設計する能力を身に付ける学習は, 教育学などの教育に関連した分野の言葉でいうところの問題解決学習の1種であると考えることができる。ただし, 「問題解決」とは, ここでは解決方法が既知の問題を, 既知のルールに従って問題を解決することではなく, 解決方法が未知の問題を解決するルールを発見することである。

問題解決学習において, 解決方法は“発見”されるものであるから, 以下, “アルゴリズムを設計する”という代わりに, “アルゴリズムを発見する”ということにする。

問題解決学習については, ガニェらは, 「問題解決は, 学習者に問題を解かせることによって教えられるものである」と言っている[4]。他の方法を否定することはできないが, 実際に見出すことは困難であろう。そこで, この原理をアルゴリズムの学習に当てはめることにすれば, 「アルゴリズムを発見する能力は, 実際に発見させることによって教えられる」ということになる。

現実的には, 解決方法が未知の問題を解くアルゴリズム

を発見する過程は多様であろう。しかし、発見させるという教育の課程を作成するためには、発見する過程の何らかのモデルを定める必要がある。そして、そのモデルに沿って学習者が発見するように導くのである。

アルゴリズム発見のモデルを定める。アルゴリズムは、与えられた問題の解決方法である。与えられた問題の解決方法をワンステップで見出すことが困難な場合には、一般的には、問題解決方法を見出すために、問題を元の問題と同等で、かつ、より解決の容易な小問題へのブレイクダウンする。コンピュータ・アルゴリズムを発見するためには、ブレイクダウンを繰り返し、最終的に、コンピュータの基本的な操作である「加減乗除」、「比較」、「条件による分岐」によって、直ちに解決可能な問題にまで分割されなければならない。このようにしてつくられた問題の階層構造は、多段階になることもあるので、小問題という代わりに、上位問題に対する下位問題と呼ぶことにする。

われわれが設定するアルゴリズム発見のモデルは、問題を、もとの問題より解決が容易な下位問題へ分割を繰り返し、具体的操作によって直ちに解決可能な最下位の問題に至る問題の階層構造を作る、というものである。そして、ブレイクダウンする過程で、問題を解決している者が、以前から持っている知識で、かつ、当該問題とは直接関係しないところで知った解決方法、ルールを寄せ集める。

アルゴリズム発見における問題解決学習の課程とはどのようなものかを、具体的な例をあげて説明する。以下の例は、アンブラグドの教材としてよく知られているものである[5]。この例では、“ねらい”を読めばわかるように、学習者に、アルゴリズムを自ら発見させることを目標としている。

例：重りの整列

ねらい

重さのわからないものを重さの順に並べるよい方法を見つける

必要なもの

砂か水、8個の区別できない入れ物、天びん

やること

(1) 入れ物に砂か水を違った量だけ入れ、こぼれないように封をします。

(2) 順番を入れ替えて、重さの順番がわからないようにします。

(3) 一番軽いものを見つけなさい。簡単に見つけるにはどうしますか？

注意：入れ物の重さははかりで調べます。いちどに2個の重さを比べられます。

(4) 適当に3個の重りを選び、はかりを使って、いちばん軽いものからいちばん重いものまで順に並べなさい。どうやってやりましたか？比べる回数が少なくでき

たのはどんなやり方でしたか？それはなぜですか？

(5) 次に、重り全体を、いちばん軽いものからいちばん重いものまで順に並べなさい。

整列が終わったら、並んでいる隣同士をもう一度比べて、正しい順に並んでいることを確かめなさい。

この教材は、さまざまな小問題を解かせることによって、ソートのアルゴリズムの中で、特に選択ソートを発見するよう誘導する。選択ソートは、配列の最小値を左端に移動させるという操作を繰り返して、ソートを完成させるアルゴリズムである。われわれのモデルに沿った形で言い直すと、ソート問題を、配列の大きさだけの個数の最小値の移動問題という下位問題に分割することによって見出されたアルゴリズムということになる。

具体的に分析すると(1)(2)は条件を認識させるためのものである、ということになる。(3)は、下位問題の解決方法を問うている。そして、(4)は、下位問題の解決方法をもとのソート問題の解決法に結びつけることを求めている。そして(5)で、選択ソートのアルゴリズムにまとめあげてを求めている。

4. 教育課程の提案に向けて

本研究の目的は、「データ構造とアルゴリズム」の新たな教育課程を提案することである。課程の概略を述べると、

(1) 汎用性の高いさまざまな操作を、基本的な操作で実現する方法をアルゴリズムを教える前に教えておく。

(2) アルゴリズムに対応する問題の階層構造に沿って、問題をブレイクダウンするように導く。

(3) (1)で学んだ手順を(2)の逆をたどって、アルゴリズムを組み立てさせる。

ということになる。

情報科学のカリキュラム全体を変えることは現実的ではない。それ故、他の科目の教育内容との関連を考えれば、現状で学習することになっているデータ構造やアルゴリズムは、原則的には入れ替えたり排除することができない。また、エビデンスのある形で効果を確認するためには実際に行われる授業の中で実践できる課程を提案しなければならず、この意味でも内容に制約を受ける。

当面の目標は、選択ソート、バブルソート、挿入ソート、クイックソートなど、複数のソートを教える教育課程を設計することである。この場合には、先に述べた(1)の「汎用性の高い操作」は、配列の操作に限定されるので、設計することが容易であるからである。配列以外のデータ構造を用いるアルゴリズムの教育課程を考える際には、別途、二分木やヒープ等のデータ構造の教育課程を設計する必要

がある。

本報告では、選択ソートに限った教材を提示する。対象は、情報科学を専門とする学生である。なお、配列をソートするという意味、配列から最小の要素を出力する操作および配列の2つの要素を入れ替える操作（いわゆるスワップ関数）は、既知のものとする。この節の冒頭で述べた概略の中では、①の部分で教えておく。

(1) 前提となる知識を思い出させる。

問い：配列 $a[0], \dots, a[n-1]$ の2つの要素 $a[i]$ と $a[j]$ を入れ替える関数を書きなさい。

問い：配列 $a[0], \dots, a[n-1]$ の最小値を出力する関数を書きなさい。

(2) 新しい概念を提示する。

配列が与えられたとき、各要素について、ソートが完成したときの位置を、要素の“正しい位置”と言います。

問い：配列

	0	1	2	3	4	5
a	2	6	1	3	4	7

において、左端を正しい位置とする要素はどれですか。また、その右隣を正しい位置とする要素はどれですか。

(3) 問い：配列 $a[0], \dots, a[n-1]$ の最小値の位置を返す関数を書きなさい。

(4) 問い：配列 $a[0], \dots, a[n-1]$ を、左端から順に、その位置を正しい位置とする要素に入れ替えることによって、ソートする関数を書きなさい。

5. 今後の課題

当面は、次の2つが課題である。第一に、配列のみをデータ構造とするアルゴリズム、探索アルゴリズム、バブルソート、挿入ソート、クイックソート、クイックセレクトをまとめた教育課程を設計することである。そのためには、共通する汎用的な操作を確定する必要がある。また、それぞれのアルゴリズムに対応する問題の階層構造も必要である。

第二に、設計した教育課程の効果を、エビデンスのある形で証明することである。すなわち、実際に学習者が、首尾よく想定したアルゴリズムを“発見”してくれるか、実践によって確認する。

「データ構造とアルゴリズム」の内容には、データ構造として、リスト、二分木、ヒープなど配列以外も含まれている。これらの教育に対しても同様の考え方が成り立つのか、検討する。

参考文献

- 1) 日本の情報教育・情報処理教育に関する提言2005, 情報処理学会情報処理教育委員会, 2005
<http://www.ipsj.or.jp/12kyoiku/proposal-20051029.pdf>
- 2) 基本情報技術者試験(レベル2)シラバス, 独立行政法人情報処理推進機構, 2013
https://www.jitec.ipa.go.jp/1_13download/syllabus_fe_ver3_0.pdf
- 3) S.S.スキナー: アルゴリズム設計マニュアル上, 平田富夫訳 丸善出版, 2012
- 4) R.M.ガニェ, W.W.ウェイジャー, K.C.ゴラス, J.M.ケラー: インストラクショナル・デザインの原理, 鈴木克明・岩崎信監訳, 北大路書房, 2007 p.83
- 5) T.ベル, I.H.ウィッテン, M.フェローズ: コンピュータを使わない情報教育, アンプラグドコンピュータサイエンス, 兼宗進 監訳, イーテキスト研究所, 2007