

# アルゴリズム教育における発見学習の試み

浅野考平<sup>†1</sup> 森戸隆文<sup>†2</sup>

本研究では、「発見学習」のアルゴリズム教育への応用について研究する。発見学習という言葉は、課題に対する解決方法を教授者が与えるのではなく、学習者が自ら見出すように導く教授法という意味で用いている。筆者は、以前、発見学習によるアルゴリズム教育を提案した[1]。具体的にいえば、教授者は、アルゴリズムの発見モデルを示し、ヒントを与え、学習者は、自らアルゴリズムを組み立てるといった教授法である。本年度、筆者は部分的にはあるが、発見学習の実践を行ったので、その結果について報告し、併せて実践に基づいた新たな問題を提起する。

## A Trial of Algorithm Education by Discovery Learning

KOUHEI ASANO<sup>†1</sup> TAKAFUMI MORITO<sup>†2</sup>

This study investigated how “discovery learning” can be applied to algorithm education. Discovery learning is defined as a teaching method that allows learners to discover problem-solving techniques by themselves, rather than be taught by teachers. The present author has already proposed algorithm education through discovery learning in a previous study [1]. Specifically, it is a pedagogical method in which a teacher illustrates how to discover an algorithm by providing students with some hints so that they try to construct it by themselves. This empirical research has partly applied this idea in practice, reported the results, and indicated new emerging issues.

### 1. はじめに

アルゴリズムを自ら組み立てることのできる能力を身につける教育に、教育分野ではよく知られた教授法の一つである「発見学習」を用いて実現することを試みる。

発見学習は、科学者の発見過程をモデルに、学習者が自らの直観的思考や、分析的思考を用いて発見する過程をたどることで、目標となる教育内容を自ら構築するという教授法であるが、ここでは、課題に対する解決方法を教授者が与えるのではなく、学習者が自ら見出すように導く教授法という意味で用いている。

これまでのアルゴリズム教育は、次のような3つのタイプに分類できる。

(1) 代表的なアルゴリズム、特に大規模データを処理するためのソートなど、既存のアルゴリズムを理解させることを中心とする教育である。このような教育の考え方は、「現実の問題に対するアルゴリズムは、たいていは、これらのアルゴリズムを部品として、組み合わせることによって構成することができる」、また、「アルゴリズムを設計する能力は、「優れた」アルゴリズムを理解することを通じて、自然に身につく」[3]というものである。通常、「データ構造とアルゴリズム」という科目名で、コンピュータサイエンスの専門教育の基礎に位置づけられている場合は、このタイプに属している。

(2) アルゴリズム的思考法を重視する教育である。「ある目的を達成するための仕事は、単純な手順の積み重ねによって構成されていること、それぞれの手順は解釈に曖昧さがないように詳細化されている」[2] ことがアルゴリズムの概念であるとして、このような思考法を身につけさせれば、アルゴリズムを組み立てる能力が身につく、という考えに基づいている。基本的には、コンピュータサイエンスの初学者、あるいは、プログラミング教育の導入として行われていると考えられる。

(3) アンブラグド・コンピュータ・サイエンスなどで用いられている手作業などを通じてアルゴリズムを理解、実行させて、その後、手順を忠実にプログラムとして記述させ、アルゴリズムを体感的に理解する。そのことによって、アルゴリズムの構築能力を育成するという教育である。実際、アンブラグド・コンピュータサイエンスの教材を通じて、アルゴリズムの構築を経験させ、その効果が実証されている[4][6]。

プログラミングの教育とは分離し、アルゴリズムを組み立てることに集中させることを、特徴としている。また、設計する能力を養成するという目的を明確にして、アルゴリズムを学習者が自ら組み立てる経験させるている。

本研究で提案するアルゴリズム教育は、構築を経験させることによって、組み立てる能力を育てるという意味で、基本的に(3)のタイプに属している。しかし、次の点で、杉浦ら、間辺ら[4][6]とは異なっている。

<sup>1</sup> 関西学院大学理工学部  
School of Science and Technology, Kwansei Gakuin University  
<sup>†2</sup> 関西学院大学大学院理工学研究科  
Graduate School of Science and Technology, Kwansei Gakuin University

(ア)学習者の前提条件は、プログラミング言語(C言語)は受講済みであること、座学としての“データ構造とアルゴリズム”は、受講済みであるか並行履修しており、ソートなどの基本的なアルゴリズムは、不十分であっても“知っている。”

(イ)アルゴリズムを組み立てる経験をさせるとき、対象となる問題だけを与えるのではなく、さまざまなヒントを与える必要がある。本研究では、アルゴリズムを組み立てる際に、適切な“ヒント”となる知識が、どのようなものであるか、考察し、一般に、どのような知識がヒントとなるのか、を明らかにしている。

## 2. 教育方法の概略と評価

この節では、提案する教育方法の基礎となる考え方と、方法の概略を説明する。また、これまでのアルゴリズム教育の方法と比較する。

### 2.1 問題解決とアルゴリズム

アルゴリズムは、特定の問題を一定の条件を満足する入力したときに解決する手順である。一般に「問題解決法」と呼ばれるのは、問題解決の方法そのものではなく、“問題を解決する方法を発見する方法”である。それ故、アルゴリズムを組み立てる方法は、問題解決法の一つであると言ってよい。

本研究では、与えられた問題に対するアルゴリズムを組み立てる過程を、ある種の問題解決法に従った形で定め、それに沿ってアルゴリズムを組み立てる経験をさせるためのヒントを作成する。そして、作成されたヒントを与えることによって、アルゴリズムを組み立てさせる。

問題解決法については、ポリアの著作[5]以来、さまざま論じられている。しかしながら、ここで用いる問題解決法は、常識的なもので、特別なものではない。要するに、解きたい問題を、もとの問題よりも解決が容易な小問題へ分割することを繰り返し、解決方法が既知である問題にまで分割する。それを統合することによって問題を解決するというものである。

分割のレベルは不定であるので、ここでは、小問題ではなく、下位問題と呼ぶことにしている。すなわち、ここで用いる問題解決法とは「問題を、解決が容易な下位問題へ、ブレークダウンを繰り返す」というものである。

本研究における教育方法を、まとめると、最初に、基本的な操作のプログラムによる実装させ、十分に“使いこなせる”状態にする。その後で、問題と、その問題を最上位問題とする、下位問題を与える。学習者は、与えられた下位問題と、知識として持っている基本的な操作を使って、アルゴリズムを組み立てる、という経験をするという方法

である。

### 2.2 これまでの教育方法との比較

第1節で分類した、現在のアルゴリズム教育のタイプ、それぞれについて検討し、本研究で提案する方法と比較、検討する。

まず、前節の(1)のタイプの教育は、有効である。実際、現在のアルゴリズムの研究者や、プログラムの開発者は、このようなタイプの教育で育てられたと言っても過言ではない。しかし、「既存のアルゴリズムを理解することで、アルゴリズムを組み立てる能力が自然に身につく」という教育方法は、意欲の高い学習者にとって有効であっても、そうでない学習者には、有効ではないという可能性が高い。

また、コンピュータサイエンスで学ばべき内容が増加し、アルゴリズムの教育にかかる時間が限られてくる中では、もっと効率の良い教育方法が求められていると、言っても良い。さらに、このタイプの教育では、アルゴリズムは、自分で組み立てるものではなく、既存のものを探す、というスタンスに陥る可能性もある。

次に、(2)のタイプの教育について検討する。アルゴリズム的思考法が、アルゴリズムを組み立てる上で基礎として必要であることは明らかであり、教育における有効性については実証済みである[2]。しかし、ある程度以上に複雑なアルゴリズムは、アルゴリズム的思考法を身につけるだけでは、組み立てることができない。例えば、ソート問題に対するアルゴリズムは、数多く知られている。問題を理論的に整理するだけで、これだけ多様なアルゴリズムを、思いつくとは考えにくい。問題を“整理”すること以上の直観的な思考、分析的な思考が必要とされる。

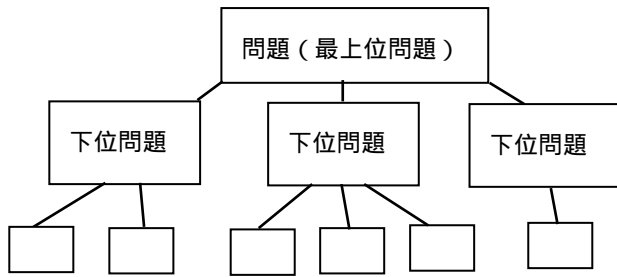
杉浦らおよび間辺らは、[4][6]において、(3)のタイプの教育を提案している。これらの論文では、アンブラグド・コンピュータサイエンスの教材によって、アルゴリズムを、体感的に理解させて、アルゴリズムの構築能力を育成するという方法の有効性を示している。

筆者は、本研究において、このようなタイプの教育方法を、発展させることを意図している。この中で用いられたアンブラグド・コンピュータサイエンスの教材の内容を検討すると、本研究でいうところの、下位問題を与えている。このことを手がかりにして、教育方法を発展させる。先に述べた学習者の前提条件の下で、活用するとともに、さらに発展させ、一般化を可能にしたい。

## 3. 授業の構成

関西学院大学理工学部の情報科学科、人間システム工学科の合同の授業“データ構造とアルゴリズム実習”という名称で、13回の授業で構成されている科目で実践した。PC教室での授業で、2014年度の受講者数は、62名である。





例として選択ソートを“発見”する過程を説明した。下位問題の選び方によって、発見されるアルゴリズムが異なることにも言及した。

ソート問題 A

入力：整数を要素とする長さ  $n$  の配列  $a[]$ 。  
出力：条件  $a[0] \leq a[1] \leq \dots \leq a[n-1]$  を満足する配列。(ただし、 $a[]$ に含まれる要素の集合は、もとの配列と同じ。)

この問題の異なる表現を与えた。

ソート問題 B

入力：整数を要素とする長さ  $n$  の配列  $a[]$ 。  
出力：条件「各  $i (0 \leq i < n)$  に対して、 $a[i]$  は、配列の小さい方から  $i+1$  番目の大きさの要素」を満足する配列。(ただし、 $a[]$ に含まれる要素の集合は、もとの配列と同じ。)

ソート問題 B の下位問題として、次の問題を与えた。すなわち、次の 2 つの下位問題が解ければ、B は解ける。

下位問題 1 (最小値を左端に移動する問題)

入力：整数が並んだ長さ  $n$  の配列  $a[]$   
出力： $a[0]$  は、 $a[]$  の最小値  
ただし、配列  $a[]$  にたくえられたデータの集合は、もとの配列と同じ

下位問題 2 (上位問題であるソート問題 B と同じで、入力サイズが 1 だけ小さい問題)

入力：整数を要素とする長さ  $n$  の配列  $a[]$ 。  
出力：条件「各  $i (0 \leq i < n-1)$  に対して、 $a[i]$  は、配列の小さい方から  $i+1$  番目の大きさの要素」を満足する配列。(ただし、 $a[]$ に含まれる要素の集合は、入力の  $a[]$  と同じ。)

下位問題 1, 2 のアルゴリズムを組み合わせれば、選択

ソートが組み立てられる。この過程を、次のように説明した。授業においては、具体的な例をあげた。

「下位問題 1 を解けば、ソート問題 B は下位問題 2 に帰着する。これを繰り返せば、長さ 1 の配列に対するソート問題に帰着し、長さ 1 の配列に対するアルゴリズムは自明である。」

下位問題 1 を実装した関数は、“アルゴリズムの部品となる基本操作の理解と実装”の段階で説明したものをを用いて、ソートのアルゴリズム (プログラム) を作成してみた。

3.3 モデルに沿ったアルゴリズムの組み立ての経験

下位問題 1, 2 のアルゴリズムを組み合わせれば、選択ソートが組み立てられる。この過程を具体的な配列で説明した。さらに次のように説明した。

下位問題 3

入力：整数が並んだ長さ  $n$  の配列  $a$  で  $a[0], \dots, a[n-2]$  は、ソート済みである。すなわち、  
 $a[0] \leq a[1] \leq a[2] \leq \dots, a[n-3] \leq a[n-2]$   
出力： $a[0], \dots, a[n-2], a[n-1]$  がソート済み。ただし、配列  $a[]$  にたくえられたデータの集合は、もとの配列と同じ

下位問題 4 (上位問題であるソート問題 A と同じで、入力サイズが 1 だけ小さい問題)

入力：整数を要素とする長さ  $n$  の配列  $a[]$ 。  
出力：条件  $a[0] \leq a[1] \leq \dots \leq a[n-2]$  を満足する配列。ただし、 $a$  に含まれる要素の集合は、入力の  $a$  と同じである。

下位問題 3, 4 を解けば、ソート問題 A が解けることを示し、3, 4 のアルゴリズムを組み合わせ、ソートのアルゴリズムを組み立てさせた。

その他、基準値による分割問題を下位問題とする、以下のような問題についても、アルゴリズムを組み立てさせた。

順位決定問題

入力：整数  $k$  と整数を要素とする長さ  $n$  の配列  $a[]$ 。  
出力： $a[k]$  の順位

この問題は、基準値による分割問題に対するアルゴリズムを変形すれば、直ちにアルゴリズムができる。

順位  $k$  要素の選択問題

入力：整数  $k$  と整数を要素とする長さ  $n$  の配列  $a[]$ 。  
出力：順位が  $k$  である  $a[]$  の要素

下位問題としては、基準値による分割問題と、サイズの小さい配列に対する同じ問題（順位  $k$  要素の選択問題）とすれば、アルゴリズムは組み立てられる。しかし、再帰呼び出しを必要とするため、学習者にとっては、難しい問題であった。

#### 4. 評価と今後の課題

提案した教育方法を評価し、改善するために、授業の中で記名式のアンケート調査を実施した。アンケートでは、“はい”または“いいえ”で答える形式の問いが7問と、自由記述形式の回答を求める問いが4問の計11問ある。内容は以下の通りである。

(ア) データ構造とアルゴリズムの授業で学んだアルゴリズムは、プリントや参考書を見ながらであれば、実行過程をトレースできそうだ。

(イ) アルゴリズムは自分で組み立てる能力を身につけることが必要であると思う。

(ウ) ((イ)がはいである場合) 必要かもしれないが、身につけることは不可能だ。

(エ) (イ) (ウ) の回答が、“はい”, または, “いいえ”である理由を説明せよ。

(オ) ヒントを参考にしながら、少なくとも1つのアルゴリズムを組み立てることができた。

(カ) 解答を聞いたら、少なくとも1つのアルゴリズムを組み立てる方法が理解できた。

(キ) (オ) および(カ)の回答が、“はい”であるときは、どのアルゴリズムか。また、“いいえ”であるときは、その理由を、わかる範囲で書け。

(ク) 「下位問題へブレークダウンする」ということの意味は理解できる。

(ケ) (はいである場合) ブレークダウンすることによってアルゴリズムを組み立てることができそうだ。

(コ) (ケ)の回答が、“はい”, または, “いいえ”である理由を説明せよ。  
である。

集計結果をもとに分析した。「アルゴリズムを自分で組み立てる能力は必要である」という問いに対する回答のほとんど“はい”であった。しかし、「既存のアルゴリズム(そのままか、あるいは類似のもの)を使えば十分、授業で組み立てるアルゴリズムも結局は既にあるものばかりである。」という回答もある。また、筆者らが授業の中で感じたことは、学習者が、例えば、ヒントが与えられているとしても、自分でアルゴリズムを組み立てたという実感が十分にはえられていない、ということであった。

これらの結果から、次のような課題が抽出された。

##### (1) 組み立てるアルゴリズムの選択

座学によるアルゴリズム教育を履修済みである点を考慮して、座学では教えていない“選択問題”や、“配列の回転”“部分列の和の最大値”など学習者にとって未知のアルゴリズムも含めたが、自分で組み立てるより、知っているアルゴリズムの中から思い出す、ことになった可能性がある。

##### (2) 下位問題の作成

本研究で提案した教育方法では、下位問題は与えている。「下位問題へブレークダウンする」ことの意味は、多くの学習者が理解するとしていた。しかし、自由記述では「どのような視点で下位問題にブレークダウンすれば良いのか、教えてほしかった」という、本質的な問題点を指摘していた。この記述にあるように、アルゴリズムを組み立てる上で、最も重要なことは、下位問題を見つけることであり、この点は今後の最大の課題である。

#### 参考文献

- 1) 浅野考平, 森戸隆文: “アルゴリズム教育再考”, 情報処理学会 コンピュータと教育研究会報告, vol.2014-CE-125 No.1, 2014
- 2) 飯田千代, 飯田周作, 清藤武暢, 佐藤創: “アルゴリズム的思考法の学習-記憶から思考へ”, 情報処理学会 コンピュータと教育研究会報告, vol.2009-CE-99 No.1, 2009
- 3) スキナー, S.S.: 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) 杉浦学, 松澤芳昭, 岡田健, 大岩元: “アルゴリズム構築能力の導入教育: 実作業による概念理解に基づくアルゴリズム構築体験とその効果”, 情報処理学会論文誌 Vol. 49 No. 10 3409-3427, 2008
- 5) ポリア, G.: いかにして問題を解くか, 柿内賢信訳, 丸善出版 1954
- 6) 間辺広樹, 兼宗進, 並木美太郎: “CSアンプラグドのアルゴリズム学習における教具による理解度への影響”, 情報処理学会論文誌 Vol. 54 No. 1 14-23, 2013