

論文

コンピュータ・アルゴリズムの 「発見・記述・伝達」を導く授業の実践と評価

間辺 広樹¹ 神藤 健朗² 並木 美太郎³ 兼宗 進⁴

受付日 2015年5月4日, 再受付日 2015年11月29日,
採録日 2016年3月4日

概要: 日常的に情報端末を活用している中学校や高等学校の生徒にとって, アルゴリズムやプログラミングの学習は重要である. これらの学習は初学者には難しいが, 生徒のパズルや数学などを解く論理的な思考力を使えば, 自分でコンピュータ・アルゴリズムを発見できると考えられる. これまで生徒が潜在的に有する思考力だけで, コンピュータ・アルゴリズムを発見できるのかどうかは明らかになっていなかった. 本研究では, 整列アルゴリズムを発見させる実習を行い, おもり(データ)の比較を記録させることで, 「正しいアルゴリズムを発見できているか」「どのアルゴリズムを発見したか」を評価した. その結果, 生徒は「難しい」と感じながらも, 80%以上の生徒が1~3回の試行で, なんらかの整列アルゴリズムを発見することができた. また, アルゴリズムは発見できたとしてもそれを他人(プログラムの場合はコンピュータ)に手順として説明する必要があるが, 正しく伝えることが難しいという課題もある. そこで, 操作を記録する際に, ひとまとまりの操作ごとに「操作の意図」と「ひとまとまりの操作をした後の中間状態(データの状態)」を書かせるようにした. その結果, 操作を記録しただけの場合と比較して, 正しく伝えられるようになることが分かった.

キーワード: コンピュータサイエンスアンプラグド, アルゴリズム教育, プログラミング教育

Evaluation of a Teaching Method to Lead Discovery, Description, and Information Conveyance about Computer Algorithms

HIROKI MANABE¹ TAKEAKI KANDO² MITAROU NAMIKI³ SUSUMU KANEMUNE⁴

Received: May 4, 2015, Revised: November 29, 2015,
Accepted: March 4, 2016

Abstract: The purpose of this study is to examine whether high school students can discover computer algorithms without teacher's teaching. It is assumed that they could do it by using their potential thinking abilities, which have cultivated through studying mathematics or solving puzzles. Therefore authors set three view points, (1) Discovery, (2) Description, and (3) Information Conveyance, as the algorithmic abilities for the evaluation. And a teaching method and materials were specially designed. As a result of the experimental lessons to 78 students at a public high school, it was indicated that students had abilities to discover some kinds of sorting algorithm with a couple of trials. Moreover, it was revealed that structural description of the algorithm were needed to convey it correctly to others. Based on these experimental results, we suggest a teaching method which shares the ideas they discovered.

Keywords: computer science unplugged, algorithm education, programming education

¹ 神奈川県立柏陽高等学校
Hakuyo High School, Yokohama, Kanagawa 247-0004, Japan

² 東京都市大学付属中学校・高等学校
Tokyo City University Junior and Senior High School, Setagaya, Tokyo 157-8560, Japan

³ 東京農工大学
Tokyo University of Agriculture and Technology, Koganei, Tokyo 184-8588, Japan

⁴ 大阪電気通信大学
Osaka Electro-Communication University, Neyagawa, Osaka 572-8530, Japan

1. はじめに

情報技術が発達し、情報機器や情報通信ネットワークは中高生らにとっても、なくてはならない存在となった。しかし、その使い方は SNS や Web 検索などを主とした受動的な利用形態にとどまっていることが多く、自分でプログラミングをして問題解決を図る能動的な使い方はなされていない。コンピュータは汎用性が高く、プログラム次第で何にでも使える「無色透明な道具」である [1]。「創造的にプログラミングできる能力は、自己の可能性を大きく広げられる」ことは、現代の児童・生徒らに伝えたいメッセージの 1 つである。そのためには、元となるコンピュータ・アルゴリズムを構築し、プログラミングをして実装することが必要である。

アルゴリズムとは「問題を解決する定型的な手法・技法」である [2]。飯田ら [3] は、アルゴリズムを考える思考力を「課題に対する解決策を考え、適切な操作の単位に分解し、他の人が実行できるように順序立てて記述する能力」として定義している。本稿で扱う「コンピュータ・アルゴリズム」とは、コンピュータの仕組みに関する制約を考慮したアルゴリズムであり、多くの生徒にとっては初めて取り組む課題となる。これらの構築や記述には、論理性が求められるため、初学者にとって難しい内容である。

一方、このような能力は、日常生活の中でも養われるものである。生徒はパズルや数学などを解く論理的な思考力を有しているため、コンピュータ・アルゴリズムを自分の力で発見できると考えられる。これまで生徒が潜在的に有する思考力だけで、コンピュータ・アルゴリズムを発見できるのかどうかは明らかになっていなかった。コンピュータ・アルゴリズムの教育は、様々なアプローチで研究が行われているが、生徒の有する思考力を前提とした教育手法については、これまで十分に議論されてこなかった。

本研究では、コンピュータサイエンスアンプラグド（以下、CS アンプラグドと記す）[4]、[5]の整列アルゴリズム学習法を用いて、高等学校の 2 つのクラスで整列アルゴリズムを発見させる実験授業を行う。コンピュータは一度に 2 つのデータしか大小比較ができないという制約のうえで、おもり（データ）の比較を繰り返し、複数個のおもりを重さの順に並べ替える実習である。その際、おもりの比較順序を記録させることで、生徒が「正しいアルゴリズムを発見できているか」「どのアルゴリズムを発見したか」を評価する。

また、「生徒が発見できたかどうかを確認する手段が存在しない」という課題もある。CS アンプラグドは協調作業でもあるため、「自分が操作できる」だけでは十分でなく、アルゴリズムを他の人（プログラムの場合はコンピュータ）に説明する必要がある。そこで、実験授業では「発見したアルゴリズムを人に伝えて操作してもらう」という活動を

取り入れる。その際に、2 つのクラスで別の教具を使い、伝わり方の違いを比較する。その結果から、客観的で分かりやすい記述や説明のためには、どのような要素が必要となるかについても考察する。

これらの結果を踏まえたうえで、生徒が発見したアルゴリズムを利用することで学習効果を見込めるアルゴリズムとプログラミングの授業展開例を示す。

2. 研究の背景

2.1 アルゴリズム教育

コンピュータは機械であるから、その仕組みに合わせた指示の仕方が必要である。たとえば、データ（要素）の検索や整列処理は、人間であればデータ全体を眺めて行うことができる。しかし、コンピュータは一度に 2 つのデータしか比較することができない、という制約がある。本研究ではコンピュータ・アルゴリズムを、その制約の中で問題解決するためのアルゴリズムととらえる。

コンピュータ・アルゴリズムの思考力を育てるために、これまで様々な方法が提案されてきた。代表的な手法の 1 つが、データに見立てた教具を手作業で操作する手法である。阿部 [6] は、コンピュータでどのようにデータが操作されるかを理解させる目的で、数字の書かれたカードを手作業で並べる操作を取り入れ、アルゴリズムの理解やアルゴリズムに工夫を加えることの大切さに気づかせるための有効性を示している。杉浦ら [7] は、学習者に手作業でアルゴリズムを実行・理解させ、その体験を忠実にプログラミングさせるアプローチを取ることで、学習者のアルゴリズム構築能力が向上し、プログラミング能力が高まることを明らかにした。これらの例は、手作業がアルゴリズムの学習に有効であることを示している。

また、コンピュータ・アルゴリズムを、学習者自身に構築させる「発見学習」も提案されている。発見学習には、「学習者が 1 つの結論を学ぶだけでなく、その結論が出来上がっていく過程に参加する学習」であり、「知識生成の過程を追体験することで、様々な学習能力の育成、伸長が期待」される [8]。加藤ら [9] は、情報科系大学生 25 名を対象にカードを用いた活動から学習者に整列アルゴリズムを発見させ、選択ソート (21 名)、挿入ソート (1 名)、交換法 (7 名) に分類されたと示されている。新開ら [10] は、阿部 [6] や杉浦ら [7] の研究を踏まえて、整列アルゴリズムを学習者に考えさせ、グループ活動による手作業で並べ替えの処理を行う活動を通して、アルゴリズムを構築する実習を実践している。このアプローチが、アルゴリズムを作成する能力をも向上させることを示し、11 グループによって選択ソート (5 グループ)、バブルソート (2 グループ)、クイックソート (2 グループ)、リスト構造を用いた方法 (2 グループ)、配列の添字を用いた方法 (1 グループ) を発見したと報告している。



図 1 天秤を用いたアルゴリズム学習

Fig. 1 Learning computer algorithm using a balance scale.

本研究も手作業と発見学習の考え方に基づいて進める。ただし、先行研究と異なり、カードなどの具体物ではなく、ソフトウェアの教具を用いる。これをカードを使うように試行錯誤しながら考えさせるが、ソフトウェアとして理解を促す機能を付加しておくことで、より高い学習効果を見込んでいる。また、発見学習の先行研究では、情報科系大学生や高等専門学校生が対象であり、学習者は一定の知識や経験を有していた。本研究では、学習経験がない高校生でも自分でアルゴリズムを発見できるのかどうかを明らかにする。

本研究では「発見」を「既知の概念であっても学習者だけの思考でそこに辿り着くこと」と定義し、既存のものであっても生徒自身で考えついたのであれば、それを発見とする。

2.2 整列アルゴリズムと CS アンプラグド

本研究では、CS アンプラグドで紹介されている整列アルゴリズム学習法を用いる [4], [5]。CS アンプラグドとは、コンピュータを使わず、カードなどの教具を使い、体験的な活動を通して子供でも情報科学の諸概念を学べるように工夫された学習法である。その中の、整列アルゴリズム学習法は、データに見立てたおもりと上皿天秤（以下、天秤と記す）を用いて、手作業でアルゴリズムを実行したり、グループで話し合いながら、ソートのアルゴリズムを理解したり、発見をする学習が想定されている（図 1）。

「同時に大小比較できるデータは 2 つだけである」という制約を明確にするために天秤を使うが、配列や変数などの概念は扱われていないため、学習者の思考力を活かしてコンピュータ・アルゴリズムの発見しやすい状況が作られている。

3. 授業の計画と教具の開発

3.1 授業の計画

授業の目標は「整列アルゴリズムを自分で発見し、きち

表 1 整列アルゴリズムの発見と記述を導く授業計画

Table 1 Lesson plan for leading discovery and description of sort algorithm.

活動	内容	時間	詳細
説明	10 分	授業の目的と教具の使い方を教師が説明する	
活動 1	発見	20 分	教具を用いて整列アルゴリズムを発見する
活動 2	記述	10 分	発見したアルゴリズムを記録カードに記述する
活動 3	交換	5 分	生徒同士で記録カードを交換する
活動 4	実行	15 分	交換したアルゴリズムを実行する
活動 5	評価	5 分	再度記録カードを交換して実行の正否を調べたのち、アンケートを実施する

	P組	Q組	調査内容
活動1	ソフト天秤		・どのアルゴリズムを発見したか ・正しいアルゴリズムを発見できたか
活動2 ～ 活動4	記録カード (P版) 作業の意図と 中間状態も記録	記録カード (Q版) データの記録のみ	・客観的に説明ができていますか

図 2 調査内容と調査に用いる教具

Fig. 2 Research contents and educational materials.

んと記述できること」とした。そのために教具を用意し、5つの活動を行う形で計画した。

まず、試行錯誤しながら整列アルゴリズムを発見するための教具として「ソフト天秤」を用意した。パソコンのスクリーン上で動作する仮想的な天秤である。これを使いながら「おもりの比較順序を記録させる」ことで『どのアルゴリズムを発見したか』『正しいアルゴリズムを発見できたか』を評価する。ソフト天秤の詳細は、3.2 節で述べる。

また、アルゴリズムを発見した生徒は手順としてきちんと説明する必要がある。そこで、『他者への伝達』を取り入れることとした。その際の教具として、アイデアや操作を記録するための 2 種類の「記録カード」を用意した。1 種類は『データの比較順序を記録するだけのカード』である。もう 1 種類は『データの比較順序をまとまり（ブロック）に分け、ひとまとまりの操作ごとに、「操作の意図」と、ひとまとまりの操作をした後の「中間状態」（データの状態）を書かせるカード』である。これを 2 つのクラスで使い分け、説明の客観性に違いが見られるかを比較する。記録カードの詳細は、3.3 節で述べる。

これらを用いて、「発見」「記述」「交換」「実行」「評価」の 5 つの活動に分けた授業を計画した（表 1）。

「活動 1」では 2 つのクラスで同一の「ソフト天秤」を使い、整列アルゴリズム発見の可否とその種類を調査する。「活動 2～活動 4」では 2 種類用意した記録カード（P 版と Q 版）をクラスによって使い分け、「操作の意図」と「中間状態」の記録の有無が記述の客観性に与える影響を調査する（図 2）。

3.2 ソフト天秤の開発と授業法の検討

筆者らは過去に、CS アンブラグドの整列アルゴリズム学習を高等学校の一斉授業で使えるように「ソフト天秤」と名付けたコンピュータのディスプレイ上で動作するソフトウェア教具を開発した [11]. 本授業では、その際に用いた教具と授業法を改良し、アルゴリズムの構築と記述を行いやすくしている.

ソフト天秤は、通常の天秤と同様に学習者がおもりの重さを比較しながら、アルゴリズムを考えたり実行できるようになっている. 図 3 に示すように、アルファベットの付いた 8 つのおもりと天秤のイラストで構成し、「ゲーム」と称した並べ替え操作を何度でも行える.

ゲームごとにおもりのアルファベット (A~H) と重量となる数 (1~8) がランダムに対応付けられる. おもりはマウス操作によって左右の上皿に 1 つずつ載せることができ、2 つのおもりを載せると数の大小関係に従って重い方へと傾く. おもりは天秤のある枠の外側であればどこにでも置くことができる. 考えるときは重量を隠すが、「表示/非表示」ボタンを押せば重量が表示され、正しく並べ替えができたかどうか判断できる. ゲームをやり直す際には、おもりの状態を初期化するか、同一の状態でも繰り返すか、を学習者が判断できるようになっている. ここまでが先行研究で用いたソフト天秤の機能である.

本授業では、発見したアルゴリズムを記録するために、このソフト天秤におもりの [比較順序] を実行履歴として残す機能と、活動 2「記述」と活動 4「実行」に使うおもりの初期状態を設定する機能を追加した. これらについての詳細を表 2 に示す.

以上の機能を実装したことで、教具である本物の天秤の

準備をすることなく、オリジナルの CS アンブラグドの学習を、一斉授業で実施することを実現した. また、本物の天秤では困難なおもりと重量の対応を変えてアルゴリズムを実行することや、あえて同一の対応でアルゴリズムを再実行できるようにした. これによって、理解の強化や誤り箇所の特定ができるようにした. さらに、[比較順序] に履歴を示したことで、実行したアルゴリズムを容易に記述できるようにした.

表 2 ソフト天秤の機能

Table 2 The functions of a software balance scale.

番号	名称	機能と使い方
1	比較	おもりのセットや、比較の指示など学習者の行動を誘導する
2	表示/非表示	実行結果の正誤を確認するために、重さとして設定した数値の表示/非表示を切り替える
3	新ゲーム	おもりと重量の対応を変えて、ゲームをできるようにする
4	再ゲーム	おもりと重量の対応を変えないまま、ゲームをやり直せるようにする
5	確認用 1	B さんが確認するためのおもりと重量の対応 (A-4, B-2, C-5, D-6, E-7, F-3, G-8, H-1) で、ゲームをできるようにする
6	確認用 2	極端なおもりと重量の対応 (A-8, B-7, C-6, D-5, E-4, F-3, G-2, H-1) で、ゲームをできるようにする
7	記録用	A さんが記録カードに記録するためのおもりと重量の対応 (A-3, B-6, C-4, D-8, E-2, F-1, G-5, H-7) で、ゲームをできるようにする
8	比較順序	実行したおもりの比較順序をアルファベットで表示する

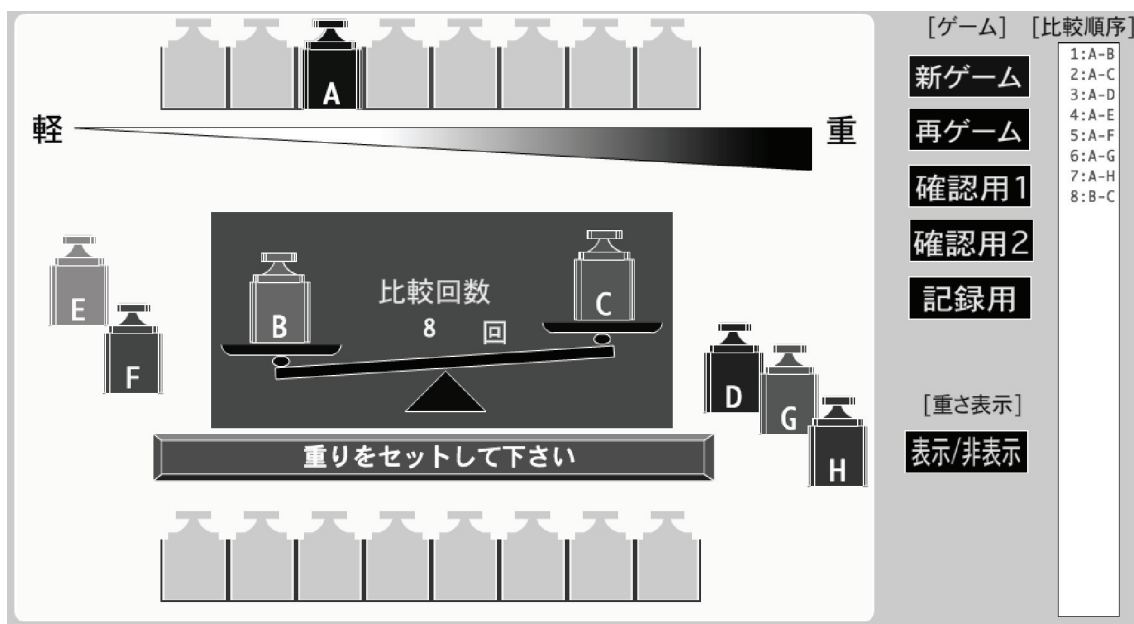


図 3 アルゴリズム実行途中のソフト天秤の状態

Fig. 3 The software balance scale in the way of operation algorithm.

図 4 記録カード (P 版) と記録する内容

Fig. 4 Recording card (P Ver.) and the details for recording.

3.3 記録カードと他者への伝達

発見した整列アルゴリズムをきちんと記述できるようにするためには、友人と考え方を伝え合う『他者への伝達』が効果的であると考えた。そこで、2人1組となって、それぞれにアルゴリズムを発見して記述する「Aさん」、相手が考えたアルゴリズムを実行する「Bさん」の2つの立場を体験する活動によって、それを実現することとした。

アルゴリズムを相手に伝えることができるかどうかは、記述の客観性に関わると考えられる。その際、「アイデアの記述」や「手順の記録」だけでも伝えられる場合もあるが、相手が初学者の場合、手順の意味が理解できないと伝わらないと考えられる。そこで、授業ではアルゴリズムの発見とともに、「意味を考えさせる記録」の有無が伝わり方に影響を与えるかどうか調べることにした。そのために、2種類の「記録カード」を作り、クラスによって使い分けることで、伝わり方に差を検証した。意味を考えさせるための「記録カード (P 版)」, 手順を記録するだけの「記録カード (Q 版)」の順にその詳細を示す。

3.3.1 記録カード (P 版)

記録カード (P 版) は A さんが書く 4 カ所 (A-1, A-2, A-3, A-4) と、B さんが書く 2 カ所 (B-1, B-2) の記述箇所を構成する (図 4)。まずは 2 人がそれぞれに A さんの立場となって、考えたアイデアを A-1 に記述する。次に、[記録用] ボタンで初期設定した状態 (図 6) でアルゴリズムを実行し、その際の比較順序を A-2 に記す。さらに、A-2 をブロック分けし、そのブロックで何をしているのかを A-3 にメモ書きする (操作の意図)。その途中で決定する事項を A-4 にメモ書きする (中間状態)。2 人が A-4 まで書いたら記録カードを交換し、今度はそれぞれが B さんの立場となる。B さんとして、A-1~A-4 に書かれたアルゴリズムを読み取り、分かりやすさを B-1 に書く。理解ができたなら、[確認用 1] ボタンで初期設定した状態 (図 7) でアルゴリズムを実行し、そのときの比較順序を B-2 に記録する。最後に記録カードを戻し、アルゴリズムが正しく伝わったかどうかを確認する、という流れである (表 3)。この中

図 5 「手順の記録」だけで伝達を行う記録カード (Q 版)

Fig. 5 The recording card (Q Ver.) used as a control experiment.



図 6 A さんが伝達・記録用に用いるおもりの初期状態
Fig. 6 The initial state of weights which A use for information conveyance and recording.



図 7 B さんが伝達後の確認用に用いるおもりの初期状態
Fig. 7 The initial state of weights which B use for confirming after information conveyance.

表 3 記録カード (P 版) に記録する内容
Table 3 The contents of the recording card (P Ver.).

活動	記述欄	記述内容	記述内容
活動 1	A-1	アイデア	A さんがアルゴリズムの基本的な考え方やアイデアを記す。
活動 2	A-2	比較順	A さんがアルゴリズムを [記録用] ボタンで初期設定した状態 (図 6) で実行した場合のおもりの比較順序を記す。
活動 2	A-3	ブロックと意図	A さんが比較順序を矢印を用いてまとまりを示し、何をしようとしているのかの「操作の意図」を記す。
活動 2	A-4	決定事項	A さんがブロックの終わりなどに、特定のおもりの順位やそこまでの比較で分かった順序などの「中間状態」を記す。
活動 3	B-1	理解の度合い	B さんが A さんのアルゴリズムが理解できたかどうかを 4 つの選択肢 (4. 理解できた~1. 理解できなかった) から選ぶ。
活動 4	B-2	伝達後の実行	B さんが A さんのアルゴリズムを [確認用 1] ボタンで初期設定した状態 (図 7) で実行した場合のおもりの比較順序を記す。

で「意味を考えさせる記録」となるのが、A-3の「操作の意図」とA-4の「中間状態」である。

授業では活動4までは、話をせずにすべて個人で行わせ、活動5ではじめて話をしてもよいこととする。

3.3.2 記録カード (Q版)

アイデアの記述と「手順の記録」だけで伝達を行うのが記録カード (Q版) である (図5)。記録カード (Q版) はAさんが書く2カ所 (A-1, A-2) と、Bさんが書く2カ所 (B-1, B-2) の記述箇所を構成する。記録カード (P版) から、「操作の意図」を記述するためのA-3と、「中間状態」を記述するA-4の記述箇所を削除したものである。活動2でこの2カ所を記述しないだけで、他の記述や活動は記録カード (P版) と同様に用いる。

4. 検証実験

実験は、H高校で「情報の科学」を学ぶ高校1年生2クラス (P組37名・Q組41名) を対象に各1コマ (65分) で行った。どちらのクラスも、整列アルゴリズムの学習経験がある生徒はいなかった。

発見の可否を確かめるために、活動1ではP組とQ組で同じ「ソフト天秤」を用いるが、記述の客観性を比較するために、活動2～活動4ではP組では図4の記録カード (P版) を使い、Q組ではA-3とA-4の部分を省略した図5の記録カード (Q版) を使った (図2)。

5. 実験結果

実験の結果を、生徒の学習の観察経過と記録カードの読み取りから、以下の6項目で示す。

- (1) アルゴリズム発見時の生徒の様子
- (2) 発見したアルゴリズムアイデア
- (3) 発見・記述・伝達できた例
- (4) 発見と伝達の成功率 (P組とQ組の比較)
- (5) アルゴリズムアイデアごとの記述の特徴
- (6) アンケート結果

5.1 アルゴリズム発見時の生徒の様子

アルゴリズムの発見活動時に、教師が観察した生徒の特徴的な様子を以下に示す。

- (1) 活動1では、P組もQ組も生徒は一樣に真剣な表情で無言で考えていた。
- (2) パソコンの画面に表示されるおもりの配置は生徒ごとに異なり、試行錯誤しながら考えていた (図8)。
- (3) 実行が完了して「表示/非表示」ボタンで結果を確認した瞬間に、どの生徒も表情が崩れ、その正否に一喜一憂していた。
- (4) 実行結果に誤りがあった生徒が、比較履歴を見てどこで間違えたのかを確認していた。
- (5) 実行結果に誤りがあった生徒が、同一状態のゲームを



図8 学習中のコンピュータディスプレイの例

Fig. 8 A computer screen while learning.

正解するまで繰り返していた。

また、教師は机間巡視しながら、どうやってそのアイデアを思いついたのかを生徒に尋ねると、「基準を作ればできるといった」「確実に最大値を選んだ」とあらかじめ考えてアルゴリズムを実行した計画性のある答えや、「おもりを動かして思いついた」「いろいろと試しているうちに思いついた」と試行錯誤しながらアイデアを思いついたという答えが返ってきた。学習の感想を尋ねると否定的な答えはなく、皆「数字がきれいに揃っていると嬉しい」「難しいけど楽しい」と肯定的に答えた。また、「こんなに頭を使ったのは初めて」と答えた生徒もいた。

5.2 発見したアルゴリズムアイデア

本節では、P組とQ組の生徒が、どのようなアルゴリズムの発見と伝達を試みたか、記録カードから読み取ったものを集計して分類する。生徒の記述の中には分かりにくいものや正しい結果を得られない記述もあったが、教師が比較順序をトレースして判断した。

その結果、ほとんどの記述が、クイックソート、挿入ソート、選択ソート、マージソートという4種類の基本的なソート法のアイデアを使ったアルゴリズムを記述していた。その中には、他の考え方とこの4種類のいずれかを組み合わせたものや、4種類のいずれかどうしを組み合わせたものもあったため、流れの中で最も大きな役割を占めた考え方を判断し、4種類へと分類した。

たとえば、整列済みデータの中央値と新しいデータを比較するというアイデアは、ヒープソートやイントロソートなど木構造を用いたソート法へとつながるアイデアであるが、大小関係から挿入位置を決めている点を重視して、挿入ソートに分類した。また、選択ソートのように、最大値を求める処理を繰り返す基本的なアルゴリズムにバブル

表 4 生徒が発見したアルゴリズムアイデアの分類

Table 4 The grouping of algorithm ideas which the students discovered.

アルゴリズムアイデア	P 組 (37 名)	Q 組 (41 名)	合計 (78 名)
クイックソート	16 名 (43.2%)	17 名 (41.5%)	33 名 (42.3%)
挿入ソート	8 名 (21.6%)	12 名 (29.3%)	20 名 (25.6%)
選択ソート	6 名 (16.2%)	6 名 (14.6%)	12 名 (15.4%)
マージソート	5 名 (13.5%)	2 名 (4.9%)	7 名 (9.0%)
トーナメント形式	0 名 (0.0%)	2 名 (4.9%)	2 名 (2.6%)
判別不能	2 名 (5.4%)	2 名 (4.9%)	4 名 (5.1%)

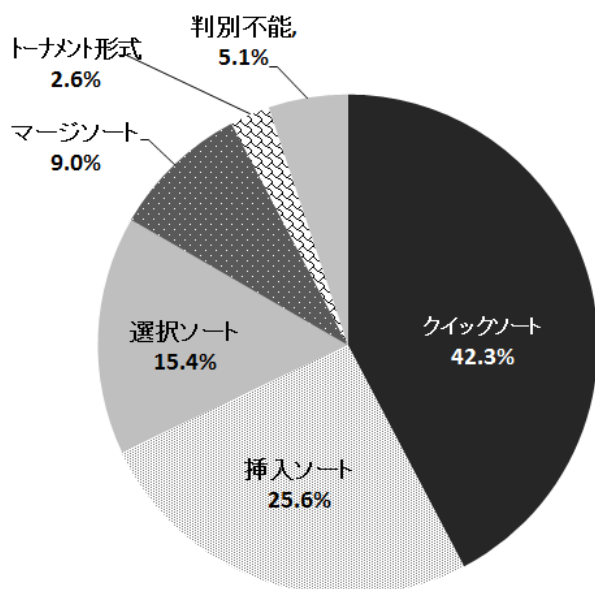


図 9 発見したアルゴリズムアイデアの分類 (n = 78)

Fig. 9 The grouping of algorithm ideas which the students discovered.

(交換) ソートがある。ソフト天秤では配列や交換の概念を扱っていないため、選択ソートとバブルソートの違いは分かりにくい。そこで、これらについてはすべて選択ソートに分類した。なお、これ以降のソート法の名称は、アルゴリズムそのものではなく、「アルゴリズムアイデア」の意味として用いるものとする。

4 種類以外にはトーナメント形式で重量の重いおもりが勝ち上がっていくアルゴリズムもあったが、決勝に上がってきたおもりの重量が必ず 1 位と 2 位になるという誤った考え方を示していた。また、何が書いてあるのかの判断ができない記述もあった。これらも含めて作成した分類表を表 4 に、P 組と Q 組を合計したグラフを図 9 に示す。

結果は、P 組・Q 組ともに出てきたアルゴリズムアイデアは、クイックソート、挿入ソート、選択ソート、マージソートの順番で一致した。また、その比率も近い結果となった。ただし、トーナメント形式の誤りは Q 組だけで見られた。また、教師らが判別できなかった記述はどちらの組も 2 名だった。

Figure 10 shows a recording card with handwritten notes and a table. The table has columns for '発見' (Discovery), '記述' (Description), and '伝達' (Conveyance). The handwritten notes include a diagram of a sorting process and a table of comparisons.

図 10 発見・記述・伝達が正しくできた記録カードの例

Fig. 10 A recording card whose discovery, description, and conveyance are correct.

5.3 発見・記述・伝達ができただ例

本節では、P 組の中から発見・記述・伝達のすべてができただ記録カード (図 10) を選び、A-1~A-4 部分を書き写して示す。つまり、学習目標に達した例である。クイックソート、挿入ソート、選択ソートという 3 種類のソート法で、複数の見本的な例が見つかった。マージソートというソート法は数が少なかったため、最もよくできたものを例示する。

5.3.1 クイックソート

クイックソートは、基準 (ピボット) を決めて振り分ける処理を再帰的に繰り返す整列アルゴリズムである。見本例を図 11 に示す。比較順は 7 回目まで A を基準に重い/軽いに振り分け、A の位置を確定している。その後、8 回目の比較で E と F の位置を確定し、9 回目以降で、残されたおもりの中から B を基準に振り分ける再帰手続きを行っている。矢印、操作の意図、決定事項 (中間状態) よりアルゴリズムを明確に記述している。伝達後の実行も成功し、B さんは理解の度合いに最高点 4 を付けている。

5.3.2 挿入ソート

挿入ソートは、整列済みデータ列の中に、新しいデータを適切な位置に挿入させる処理を繰り返す整列アルゴリズムである。見本例を図 12 に示す。比較順は、まず $A < C < B$ の順で整列させておいて、その後、D, E, F... と各データの適切な位置を見つける意図が読み取れる。たとえば、データ E は右から順番に比較し、「C より小さく、A より大きい」ことを確認して、A と C の間に挿入している (5 回目から 8 回目)。このように新しいデータを整列済みデータとの大小関係から適切な挿入する処理を繰り返して、整列を完了している。矢印、操作の意図、決定事項 (中間状態) よりアルゴリズムを明確に記述している。伝達後の実行も成功し、B さんは理解の度合いに最高点 4 を付けている。

5.3.3 選択ソート

選択ソートは、未整列のデータから最大値 (または最小値) を見つける処理を繰り返す整列アルゴリズムである。

アイデア		おもりの 比較順	何をしようとしているか (ブロック単位で記述)	決定事項 順位の決定等
とにかく二分してみる！ 二分したものをそれまた二分… としていき最終的に2つの重さ のものををはかるだけでよくなる ようにする。	1	A — B	Aを境に 「Aより軽いもの」 「Aより重いもの」 に二分する	
	2	A — C		
	3	A — D		
	4	A — E		
	5	A — F	EとFのどちらが重いかわかる	(E, F) < A < (B, C, D, G, H)
	6	A — G		E < F < A < (B, C, D, G, H)
	7	A — H		
	8	E — F		
	9	B — C	(B, C, D, G, H)について 「Bより軽いもの」「Bより重いもの」 に二分する	
	10	B — D		
	11	B — G		
	12	B — H		E < F < A < (C, G) < B < (D, H)
	13	C — G	CとGのどちらが重いかわかる	E < F < A < C < G < B < (D, H)
	14	D — H	DとHのどちらが重いかわかる	E < F < A < C < G < B < H < D
	15	—		

図 11 クイックソートを発見し伝達できた例

Fig. 11 An example of discovering and conveying quick sort.

アイデア		おもりの 比較順	何をしようとしているか (ブロック単位で記述)	決定事項 順位の決定等
まずA-B → 重い方とCを計る その後は必ず重い方を残してい き、それよりも軽いおもりが現 れたら、そのおもりがどこに入 るかを1つずつ確かめていく。	1	A — B	ABCの3つの順番を知る	A < B
	2	B — C		
	3	A — C		A < C < B
	4	B — D		A < C < B < D
	5	D — E	Eがどこに入るのかわかる	
	6	E — B		
	7	E — C		
	8	E — A		E < A < C < B < D
	9	D — F	Fがどこに入るのかわかる	
	10	B — F		
	11	C — F		
	12	F — A		F < E < A < C < B < D
	13	F — E	Gがどこに入るのかわかる	
	14	D — G		
	15	B — G		
	16	C — G		F < E < A < C < G < B < D
	17	H — D	Hがどこに入るのかわかる	
	18	H — B		F < E < A < C < G < B < H < D
	19	—		

図 12 挿入ソートを発見し伝達できた例

Fig. 12 An example of discovering and conveying insert sort.

このアルゴリズムは2重ループで実現するが、同一のおもりの組合せで何度も比較することから比較回数が多くなり、効率は悪い。見本例を図13に示す。未整列のデータ列から最大値を求める処理を7回繰り返す、合計で28回の比較（ $7+6+5+\dots+1=28$ ）によって整列を完了している。矢印、操作の意図、決定事項（中間状態）よりアルゴリズムを明確に記述してる。Bさんは理解の度合いに最高点4を付けている。

5.3.4 マージソート

マージソートは、整列済みのデータ列どうしを併合してより長い整列済みデータ列を作る処理を繰り返す整列アルゴリズムである。図14は、まず、1~4回目の比較で隣合う2つのおもりどうしで比較し、[データ数 2×4 組]の整列済みデータ列を作っている。次に、5~10回目の比較で、[データ数 4×2 組]の整列済みデータ列を作っている。

最後に、11回目以降の比較で[データ数 8×1 組]へと併合し、整列を完了している。太線、操作の意図、決定事項（中間状態）の記述はあるが、他のアルゴリズムに比べると分かりにくい。伝達後の実行は成功しているが、Bさんの理解の度合いも3にとどまっている。

5.4 発見と伝達の成功率（P組とQ組の比較）

記録カードから、整列アルゴリズムの発見と伝達の程度を調査した。その際に、次のように扱う数を呼称する用語を定義した。まず、A-1~A-4の記録から、考え方を読み取れるものだけを分析の対象とし、その数と比率を**読取数（率）**と定義した。次に、読み取れるカードからP組はA-1~A-4の記録を、Q組はA-1とA-2の記録を見て、正しい結果が得られるアルゴリズムが記録されていたら、発見が成功したと判断し、その数と比率を**発見数（率）**と定

アイデア		おもりの比較順	何をしようとしているか (ブロック単位で記述)	決定事項 順位の決定等
2つのおもりを比べて重かった方を次のおもりと比べる。そして全部比べたときに最後に残った2つのおもりのうち、重い方が一番重いおもりとなる。次に、その一番重いおもり以外で上記と同じ事をもう一度行い、その中で一番重いおもりを決める。これを繰り返していくと、一番重いおもりから順に決まっていき、最終的には全部見つけることができる。	1	A - B	↑	
	2	C - B		
	3	D - B		
	4	D - E		
	5	D - F	↓	一番重いおもりを決めようとしている
	6	D - G		
	7	D - H		
	8	A - C		Dが一番重い
	9	B - C	↑	
	10	B - E		
	11	B - F		
	12	B - H		
	13	G - H	↓	二番目に重いおもりを決めようとしている
	14	G - A		
	15	G - B		Hが二番目に重い
	16	C - B		
	17	F - B	↓	三番目に重いおもりを決めようとしている
	18	E - B		
	19	E - C		Bが三番目に重い
	20	F - C		
	21	G - C	↑	四番目に重いおもりを決めようとしている
	22	G - A		
	23	E - F		Gが四番目に重い
	24	E - C		
	25	A - C	↓	四番目に軽いおもりを決めようとしている
	26	A - E		Cが四番目に軽い
	27	A - F		
	28	E - F		Aが三番目に軽い Eが二番目に軽くFが一番軽い

図 13 選択ソートを発見し伝達できた例

Fig. 13 An example of discovering and conveying selection sort.

アイデア		おもりの比較順	何をしようとしているか (ブロック単位で記述)	決定事項 順位の決定等
8個を2個ずつ4組に分けてそれぞれの大小をはかる。4組をさらに2組に分けて、その中の大小の順番を決める。残った2組で重さを決めていく。	1	A - H	2個ずつ4組に分けて2つの大小をはかる。	H > A
	2	B - G		B > G
	3	C - F		C > F
	4	D - E		D > E
	5	B - H	4組をさらに2組に分けてその中で大小の順番を決める	H > B
	6	B - A		B > A
	7	G - A		H > B > G > A
	8	D - C		D > C
	9	E - C	↓	C > E
	10	E - F		D > C > E > F
	11	H - D		Dが一番重い
	12	H - C		Hが2番目に重い
	13	B - C	2番目から大きいもの同士を比べて重かったものから順位を決めていく	Bが3番目に重い
	14	G - C		Gが4番目に重い
	15	A - C		Cが5番目に重い
	16	A - E		Aが6番目、Eが7番目、Fが8番目
	17			

図 14 マージソートを発見し伝達できた例

Fig. 14 An example of discovering and conveying merge sort.

義した。さらに、実行結果の B-2 の記録を1つずつトレースし、A さんのアルゴリズムと同一のアルゴリズムで実行していたら伝達が成功したと判断し、その数と比率を伝達数(率)と定義した。そのうえで、B さんの実行結果が正しかったら、すべてが成功したと判断し、その数と比率を成功数(率)と定義した。

読み取ったデータをまとめたものが、表 5 と図 15 である。発見率は、どちらのクラスとも 80% 強と高い割合で正しく実行するアルゴリズムを発見できたことが分かった。

しかし、伝達率は P 組は 70.3%、Q 組は 36.5% と伝達の際に差が生じた。最終的な成功率は、P 組は 64.9%、Q 組は 29.3% であった。統計的な有意差は認められなかったが、クラスによる違いが生じた。

各クラスのアルゴリズムアイデアごとの読取数、発見数、伝達数、成功数、ならびに B-1 に記述した理解の度合いの平均を表 6、表 7 に示す。なお、これらの表に示した比率の母数には読取数を用いている。

表 5 整列アルゴリズムの発見数と伝達数

Table 5 The each number of discovery and conveyance of sort algorithm.

	P 組 (37 名)	Q 組 (41 名)	比率の計算法
読取数 (率)	35 名 (94.6%)	39 名 (95.1%)	読取数 ÷ 生徒数
発見数 (率)	31 名 (83.7%)	34 名 (82.9%)	発見数 ÷ 生徒数
伝達数 (率)	26 名 (70.3%)	15 名 (36.6%)	伝達数 ÷ 生徒数
成功数 (率)	24 名 (64.9%)	12 名 (29.3%)	成功数 ÷ 生徒数

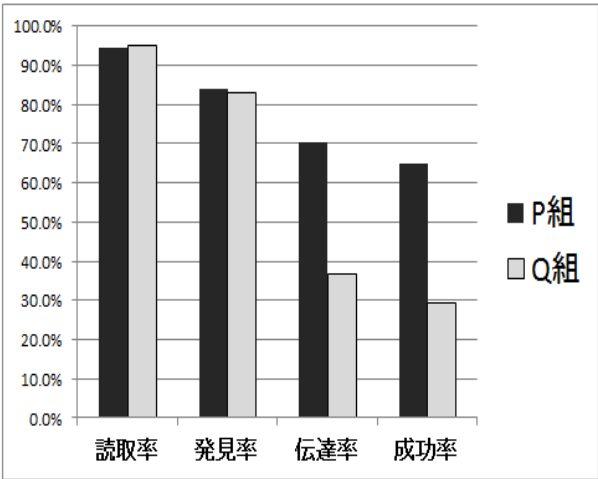


図 15 クラス別の読取率・発見率・伝達率・成功率

Fig. 15 The each number of discovery and conveyance of sort algorithm.

表 6 P 組のアルゴリズムごとの発見数と伝達数

Table 6 The numbers of discovery and conveyance of each algorithm in P class.

	クイックソート	挿入ソート	選択ソート	マージソート
読取数	16 名	8 名	6 名	5 名
発見数	16 名 (100%)	8 名 (100%)	2 名 (33%)	5 名 (100%)
伝達数	16 名 (100%)	8 名 (100%)	2 名 (33%)	4 名 (80%)
成功数	16 名 (100%)	8 名 (100%)	1 名 (17%)	3 名 (60%)
理解平均	4.0 点	3.9 点	3.9 点	3.9 点

表 7 Q 組のアルゴリズムごとの発見数と伝達数

Table 7 The numbers of discovery and conveyance of each algorithm in Q class.

	クイックソート	挿入ソート	選択ソート	マージソート
読取数	17 名	12 名	4 名	2 名
発見数	17 名 (100%)	12 名 (100%)	2 名 (50%)	2 名 (100%)
伝達数	12 名 (71%)	3 名 (25%)	0 名 (0%)	0 名 (0%)
成功数	12 名 (71%)	3 名 (25%)	0 名 (0%)	0 名 (0%)
理解平均	3.2 点	3.0 点	3.0 点	3.5 点

5.5 アルゴリズムアイデアごとの記述の特徴

本節では、前節で分類したアルゴリズムアイデアごとに、記録カードに表れた特徴を示す。

5.5.1 クイックソート

クイックソートのアイデアは、最も多く発見され、図 11 のように、A を基準に「A より軽いもの」と「A より重いもの」に振り分け、振り分けた先で再度ピボット操作を行

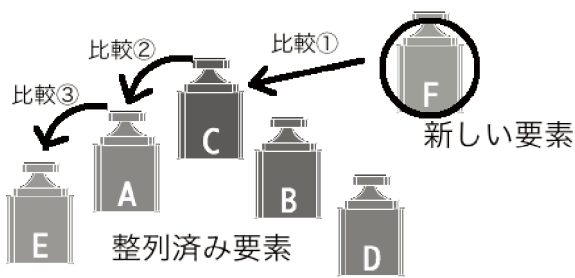


図 16 整列済みデータ列の中央から新規データを比較する挿入ソート

Fig. 16 The insert sort which compared a new datum with the centered datum.

	おもりの比較順	何をしようとしているか (ブロック単位で記述)	決定事項 順位の設定等
1	A	B	
2	B	C	
3	C	D	
4	D	E	
5	E	F	
6	F	G	
7	G	H	
8	B	F	Dが8
9	B	E	BはACより重いためBを基準に2番目に重なおもり
10	B	G	ついでに3番目もわかった
11	B	H	Bが6 Hが7
12	A	C	4番目に重いのを求める
13	C	E	ついでに5番目もわかった
14	C	F	
15	C	G	Cが4 Gが5
16	A	E	6番目に重いのを見つける
17	A	F	Aが3
18	F	F	残った2つを比べる
19			Fが2 Fが1

図 17 2つのデータの順位が同時に分かる例

Fig. 17 An example of deciding the two data at the same time.

うことがおおむね記述されていた。実行結果も正しいものが多かった。クイックソートを発見したのは合計で 33 名であり、伝達数と理解の度合いの平均も高かった。

5.5.2 挿入ソート

挿入ソートは、クイックソートに次いで多く発見されたアイデアであった。この考え方にたどり着いた生徒の多くは、決定事項の欄に確定したおもりの大小順を分かりやすく記載していた。ただし、実際には図 12 のように新しいデータを整列済みの列の端から比較していく「単純挿入ソート」は少なく、図 16 のように新しいデータを整列済みの列の中央値と比較して挿入位置を決める方法を、挿入ソートを発見した P 組 8 名中の 5 名、Q 組 12 名の中の 7 名の合計 12 名もが記述していた。中央と比較するアイデアを出した生徒の中には、実験中にソフト天秤の画面上におもりをツリー状に配置して、2 分木の概念を自分で編み出していることがうかがえた。

5.5.3 選択ソート

選択ソートは、正しい伝達ができないアルゴリズムであった。生徒の記述には「最大値を求める」「確実に 1 つ 1 つ順位を決める」などのアイデアが記述されている。しかし、偶然に 2 つのデータの順位が同時に分かってしまう場合 (図 17) がある。このようなケースは、本来の選択ソートよりも比較回数を減らした効率の良いソートができる可能性があるが、その際に、A さんがどうやって効率化を図ったのかがプリントに記述されていなかったため、B

表 8 アルゴリズムを思いつくまでのゲームの実行回数

Table 8 The number of playing games to hit on any algorithm.

	P 組 (37 名)	Q 組 (41 名)	合計 (78 名)
0 回	1 名	0 名	1 名 (1.3%)
1 回	7 名	9 名	16 名 (20.5%)
2 回	15 名	15 名	30 名 (38.5%)
3 回	7 名	8 名	15 名 (19.2%)
4 回	3 名	4 名	7 名 (9.0%)
5 回	2 名	3 名	5 名 (6.4%)
6 回以上	2 名	2 名	3 名 (3.9%)
平均	2.57 回	2.61 回	2.59 回
標準偏差	1.62	1.45	1.53

表 9 もっと知りたいと思ったこと

Table 9 The list of what the students want to learn more.

もっと知りたいと思ったこと	P (37 名)	Q (41 名)	計 (78 名)
一番効率の良い整列アルゴリズム	16 名	12 名	28 名
他の生徒が考えた整列アルゴリズム	10 名	10 名	20 名
アルゴリズムを分かりやすく伝える方法	0 名	10 名	10 名
コンピュータに実装する方法	4 名	3 名	7 名
他のコンピュータ・アルゴリズム	3 名	2 名	5 名
コンピュータの (高速に処理する) 仕組み	2 名	1 名	3 名

さんには伝わらなかった。

5.5.4 マージソート

マージソートは、P 組が 5 名、Q 組が 2 名と少なかった。マージソートは、データ全体を細かく分割して整列させ、1 列になるまで併合を繰り返すアルゴリズムである。データ数が 8 であれば、[データ数 2×4 組] → [データ数 4×2 組] → [データ数 8×1 組] となる。しかし、マージソートに分類した生徒のアイデアには、最初 [データ数 3×2 組とデータ数 2×1 組] と 3 組に分割した後、併合していくパターンが多く、P 組と Q 組を合わせた 7 パターンのうち、5 パターンが [データ数 3×2 組とデータ数 2×1 組] の [3-3-2] 分割を行っていた。

5.6 アンケート結果

アンケート調査した「ゲームの実行回数」「もっと知りたいと思ったこと」「発見と理解の難しさ」の結果を示す。

5.6.1 アルゴリズムを思いつくまでのゲームの実行回数

生徒がアルゴリズムを発見するまでにソフト天秤を用いてアルゴリズムを実行したゲームの回数を表 8 に示す。P 組と Q 組で大きな差はなく、80% 近い生徒が 1 回から 3 回の実行で発見できたことが分かった。

5.6.2 もっと知りたいと思ったこと

アンケートの「もっと知りたいと思ったこと」については、記述内容を読み取り、同種の回答を表 9 にまとめた。その結果、P 組と Q 組のどちらも「一番効率の良い整列アルゴリズム」「他の生徒が考えた整列アルゴリズム」の順で、『他の整列アルゴリズム』を知りたいと答えた生徒が多かった。一方、「アルゴリズムを分かりやすく伝える方法」

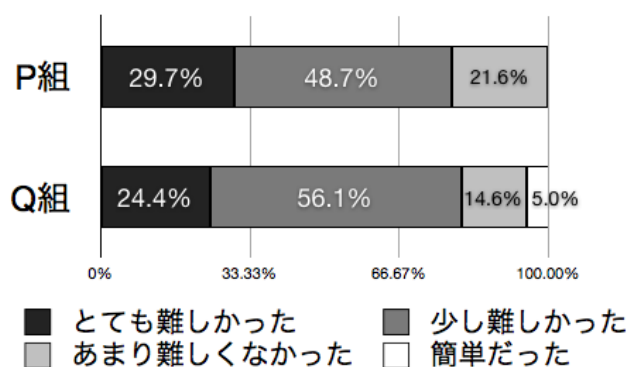


図 18 自分で整列アルゴリズムを作ることについての難しさ

Fig. 18 The difficulty in thinking of an original sort algorithm by themselves.

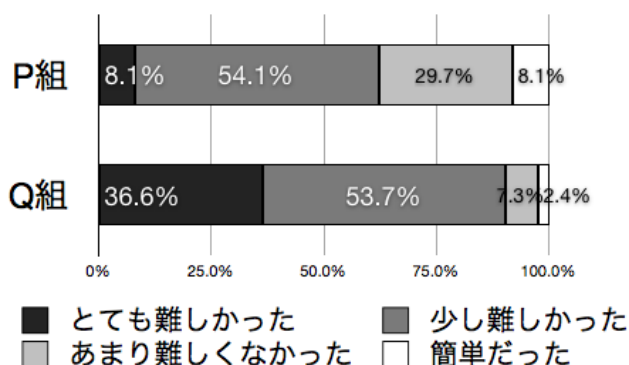


図 19 人の整列アルゴリズムを理解することについての難しさ

Fig. 19 The difficulty in understanding the sort algorithm come up with by others.

を P 組は 1 人も答えていないのに対して、Q 組が 10 名と差が生じた。ほかには P 組と Q 組のどちらにも、実装の方法などコンピュータの仕組みを知りたいと答えた記述があった。

5.6.3 難しさについてのアンケート結果

授業の最後に「自分でアルゴリズムを作ること」と「人のアルゴリズムを理解すること」の難しさについてアンケート調査した。それぞれの設問について「とても難しかった」から「簡単だった」までの 4 段階から選択させ、まとめたものを図 18 と図 19 に示す。

「自分で整列アルゴリズムを作ることについての難しさ」については、どちらのクラスも「とても難しかった」「少し難しかった」を合わせて約 80% となり、同様の傾向を示した。「人の整列アルゴリズムを理解することについての難しさ」については、「とても難しかった」「少し難しかった」を合わせた難しさを感じた生徒の比率が、P 組では 62.2%、Q 組では 90.3% と差が生じた。

6. 考察

本研究では、生徒が潜在的に持つ思考力だけで、コンピュータ・アルゴリズムの基本である整列アルゴリズムの (1) 発見、(2) 記述、(3) 伝達の 3 点ができるかを高等学校

の2つのクラスで実験し、生徒がアルゴリズムを発見できることと、どのようなアルゴリズムを発見するかを示した。また、2つの教具の比較を通して、その記述の客観性に影響を与える要素を明らかにした。本章では、これらの結果に至った要因を考察する。また、生徒が発見したアルゴリズムを利用して可能になるアルゴリズムやプログラミングの授業展開例を示す。

6.1 発見の可否

本授業が一般的な授業と異なるのは、教師から知識を伝えるのではなく、発見や記録に関する思考のすべてをそれぞれの生徒に行わせたことである。そのうえで、「一度に比較できるデータは2つだけ」というコンピュータの制約を加えた状態で整列アルゴリズムを考えさせた。そして「他者への伝達」を取り入れて、考え方を記録カードに記述させた。

その結果、90%以上の生徒が教師が読み取れる考え方を記述し、80%以上の生徒がクイックソート、挿入ソート、選択ソート、マージソートという4種類の基本的な整列アルゴリズムにつながるアイデアの発見と、正しい結果を得るためのデータの比較順序を書けるようになったことが分かった(表5, 図15)。また、約80%の生徒がアルゴリズムを考えることを「難しい」と感じる(図18)が、集中して考えることで平均1~3回程度アルゴリズムの実行で発見にたどり着いた(表8)。これらのことから、初学者でも、短時間でコンピュータ・アルゴリズムを構築して記述できることが確かめられた。

この達成率に至った理由として、次の3点をあげる。第1に「生徒の思考力で解けるコンピュータ・アルゴリズムを題材にした」という点である。生徒は整列アルゴリズムを初めて学習した。しかし、すでに有していた思考力と、ソフト天秤を用いての試行錯誤などを組み合わせたことによって、客観的に記述されたコンピュータ・アルゴリズムの形にすることができた。生徒の多くは難しさを感じたものの、それが無理なレベルではなかったことで、多くの生徒が学習を達成した。

第2に、「パズルを解くような面白さがあった」という点である。実行結果を確認したときの生徒の表情や感想から、活動に楽しさや達成感を感じていた様子がうかがえる。正解したときは、素直に喜んでいて、正解しなかったときは、正解するまで何度も実行を繰り返したり、間違えた箇所を確認しようと比較順序を見直していた。比較回数を少なくしようと、アイデアを絞る生徒もいた。このように、発見時の活動は「できた/できない」がはっきりと分けられたり、比較回数がポイントの役割となったことで、娯楽性のあるパズルゲームのような活動となり、生徒の集中力を引き出したと考えられる。

第3に、「ソフト天秤が活動を実現した」という点であ

る。ソフト天秤は操作が容易であり、試行錯誤しながら考えること、短時間でのアルゴリズムの実行、同一条件でのやり直しなどを可能にした。また、比較回数の表示や「表示/非表示」のボタンなどでゲーム性を高めた。これらの仕組みが生徒を考えることに集中させ、短時間での学習を達成した。

このように「適切な課題設定」「パズルのような面白さ」「操作が容易なソフトウェア教具」が揃えば、他のコンピュータ・アルゴリズムについても同種の発見授業が可能であると考えられる。それぞれのアルゴリズムで、適切な条件設定や教具の開発が必要になるが、これらについては今後の研究課題としたい。

6.2 「操作の意図」と「中間状態」の記述の影響

説明の客観性を評価するためにアルゴリズムを記述するための記録カードを、P組とQ組で分けた。P組では、データの比較順序をブロックに分割し、それぞれのブロックで何をしているのかの「操作の意図」を書かさせた。また、アルゴリズムの実行にともなって変化する「中間状態」を記録させた。その結果、P組においては、多くの生徒の記録カードから、「比較順のまとまり(ブロック)への分割」、「まとまりごとの操作の意図」、「ひとまとまりの操作をした後の中間状態」が適切に記述されていて、アルゴリズムを構築できていたことが確認できた。

発見したアルゴリズムは2人1組となってお互いに交換と実行を行う「他者への伝達」を活動として取り入れた。その結果、記録カードにA-3の「操作の意図」とA-4の「中間状態」を記述したP組では伝達率が約70%、成功率が約65%だったのに対し、A-3とA-4を記述しなかったQ組では、伝達率が約37%、成功率が30%と差が生じた。生徒のアンケートの「もっと知りたいと思った内容」の記述から、「アルゴリズムを分かりやすく伝える方法」をP組が0人、Q組が10人と差が生じた(表9)。このことから、A-3とA-4の記述の有無が、伝達の可否に大きく影響していることが分かる。経験ある教師であればA-1とA-2の記述だけでも、そこに生徒の意図やアルゴリズムをある程度は読み取ることができる(表4, 図9)が、初学者には困難である。しかし、A-3とA-4の「操作の意図」と「中間状態」の記述を付け加えることで、説明の客観性が向上し、初学者であるBさんにも伝達することができた。その結果、初期状態が異なったおもりでも同一のアルゴリズムを正しく実行できるようになったと考えられる。

一般的な授業においても、教授したアルゴリズムが学習者にきちんと伝わらないことがある。本研究で示したような記述や説明を付け加えることで、学習者の理解を促すことが考えられる。この件については今後の研究課題としたい。

6.3 アルゴリズム学習への応用

アンケートでは、生徒の多くが「一番効率の良い整列アルゴリズム」や「他の生徒が考えた整列アルゴリズム」を知りたいと考えるようになることから、本授業がアルゴリズム学習への動機付けにもなる。これらのことから、本授業の後、本格的なアルゴリズム学習へと展開するために効果的と思われる教授法を提案する。

高等学校情報科の学習指導要領 [12] は、共通科目「情報の科学」の「問題解決とコンピュータの活用」単元で、『整列や探索などの基本的なアルゴリズム、簡単なアルゴリズムを生徒に表現させ、それを自動実行させるなどの体験的な学習活動を通じて行うことが考えられる。また、処理手順に簡単な変更を行うだけで処理結果に違いが出たり、少しでも処理手順に誤りがあると想定どおりの結果が出なかったり、処理時間に大きな違いが生じたりすることも理解させる。』とその目標を示している。そこで、教授法として提案するのは、アルゴリズムの**理解・修正・発展**を目的に、発見した整列アルゴリズムを生徒に発表させて、**アイデアの共有化**を行う授業である。

理解とは、他のアルゴリズムや計算量などの理解である。2組ともそれぞれに4種以上のアルゴリズムアイデアを発見した。生徒も「他の生徒が考えた整列アルゴリズム」を知りたいと考えるようになった。したがって、図 11～図 14 のようにきちんと記述できた生徒を指名して説明をさせれば、他の生徒にもその考え方は伝わりやすく、生徒は複数のアルゴリズムを同時に学べるようになる。このことは、アルゴリズムの効率や計算量の違いを意識することにもつながり、学習指導要領の目標とも一致する。

修正とは、あえて誤った考え方を示すことで、同じ間違いを生じさせないようにすることである。本授業でもトーナメント形式では、決勝まで上がってきたデータを1位と2位にするという誤った考え方で、「解けない」アルゴリズムを考えてしまう生徒もいた。したがって、このような誤りとともに、正しい考え方を示すことで、同種の誤りが生じなくなることが期待できる。

発展とは、生徒の考え方に関連する、より高度な概念を話題にすることである。本授業でもクイックソート、挿入ソート、選択ソートで、他の概念へとつながる考え方を示した生徒がいた。クイックソートは最も多く発見されたが、それが再帰構造で実現することまで生徒は認識できているとはいえない。しかし、多く発見されていることは、それだけ考えやすいアイデアであるにとらえれば、その概念の理解は容易であるといえる。また、挿入ソートは基本挿入ソートのように整列済みデータ列の端から比較するのではなく、中央と比較するアイデアを示した生徒が多かった。これは、データ構造としての2分木へとつながる概念である。選択ソートは同時に2つの位置が分かる状況などを利用して、効率化を図った生徒もいた。このように、教

師は生徒のアイデアをうまくとらえることで、より高度な概念の習得につながる展開が可能と考えられる。

これらの**修正や発展**として取り上げた例は本授業の生徒のものであり、他の授業で同程度のものが出てくるとは限らないが、アイデアを生徒に発表させたり、教師が生徒のアイデアを拾い上げるなどの**アイデアの共有化**は、生徒の学習意識や思考力を向上させると考えている。この授業の検証についても、今後の研究課題である。

6.4 プログラミング学習への応用

本授業の、プログラミング学習への展開の可能性を考察する。P組では、記録カードには、ブロック分割と「操作の意図」、「中間状態」を記録させた。これは、プログラミング学習につながる客観的な記述であると考えている。

たとえば、挿入ソートの記述(図 12)を見ると、「○よりは小さく、△よりは大きい」という複合条件で新規のデータを処理する考え方が構築できている。したがって、「if 文による条件分岐を組み合わせること」を指導すれば、プログラムの作成につながる。if 文を組み合わせることは初学者には難しい。しかし、図 12 のような形で考え方を理解できているという前提があれば、if 文との組み合わせ方に意識を集中できることになり、通常の教え方よりも理解しやすくなると考えられる。

同様に、選択ソートの記述(図 13)からは、2重ループが構築できている。したがって、「for ループによる入れ子を作ること」を指導すれば、プログラムの作成につながる。繰返しに入れ子を作ることは、カウンタ変数の適切な使い方が必要であり、初学者には難しい。しかし、図 13 のような形で2重ループを理解しているという前提があれば、for ループの組合せ方に意識を集中できることになり、通常の教え方よりも理解しやすくなると考えられる。

本授業では、「2つのデータしか比較できない」というコンピュータの制約は残したが、実際のプログラミングに必要な「変数」「配列」「メモリ」「データの交換」などの概念は扱っていない。これらは新たな制約となり、理解を難しくするが、アルゴリズムを理解できていることは、プログラミング修得へのアドバンテージになると考える。本授業からプログラミング学習への展開も、今後の研究課題である。

7. まとめ

本研究は、コンピュータを創造的に利用させるために、生徒の思考力だけで、整列アルゴリズムの発見・記述・伝達ができるかを、高等学校の2クラスで調査した。その結果、生徒が整列アルゴリズムを発見できることを示し、発見しやすい整列アルゴリズムを明らかにすることができた。また、アイデアとデータの比較順序の記録だけでなく、操作の意図と中間状態を記録させることで、客観的で伝わり

やすい記述となることが分かった。

生徒は自分が考えたアイデアが他の人に伝わって、正しく実行されたことで、達成感や創造することの喜び、論理的に考えることの重要さなどを感じてくれたと思っている。教師の立場としては、生徒に知識を伝えるのではなく、生徒の能力を前提にした授業や生徒から出てきたアイデアを活用した授業を行うことで、効果的に授業を進められると考えている。

本研究は初学者を対象としたものであり、よりレベルの高い学習へとつなげていくための通過点である。本稿で提案した教授法やさらにその先のアルゴリズムやプログラミングの学習に向けて多くの研究課題が残っているが、生徒に楽しみながら創造的にプログラミングできる能力を身に付けさせるために、これらの課題を1つずつ解決していきたい。

謝辞 本研究は、科学研究費補助金（基盤研究（C）25350214，奨励研究 15H00219）の補助を受けている。

参考文献

- [1] 坂村 健：痛快！コンピュータ，集英社インターナショナル（1999）。
- [2] 新村 出：広辞苑第六版，岩波新書（2008）。
- [3] 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，情報処理学会研究報告 コンピュータと教育（CE）2009-CE-99(1)，pp.1-7（2009）。
- [4] Bell, T., Alexander, J., Freeman, I. and Grimley, M.: Computer Science Unplugged: School Students Doing Real Computing Without Computers, *The NZ Journal of applied computing and information technology*, Vol.13, No.1, pp.20-29（2009）。
- [5] 兼宗 進（監訳）：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，イーテキスト研究所（2007）。入手先（<http://www.etext.jp/>）
- [6] 阿部哲也：アルゴリズム教育のための教材の開発，日本教育工学雑誌，27(suppl)，pp.45-48（2004）。
- [7] 杉浦 学，松澤芳昭，岡田 健，大岩 元：アルゴリズム構築能力育成の導入教育：実操作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, pp.3409-3427（2008）。
- [8] 水越敏行：発見学習入門，明治図書（1970）。
- [9] 加藤 聡，富永浩之：プログラミングへの導入としてのアルゴリズム的思考を養成するアンプラグドなグループ演習，情報処理学会研究報告，コンピュータと教育研究会報告 2008(64)，pp.63-70（2008）。
- [10] 新開純子，宮地 功：手作業による体験的アルゴリズム学習の実践，日本教育工学学会論文誌，35(Suppl.)，pp.129-132（2011）。
- [11] 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, pp.14-23（2013）。
- [12] 文部科学省：高等学校学習指導要領解説情報2編（2010）。



間辺 広樹 （正会員）

1986 年東京理科大学理工学部数学科卒業。同年より神奈川県立高校にて数学科教諭として勤務。2010 年より社会人学生として大阪電気通信大学医療福祉工学研究科博士課程に在籍。2013 年同課程修了。博士（工学）。2013 年より神奈川県立柏陽高等学校にて情報科・数学科教諭として勤務。情報科学教育とオンライン学習教材の研究に従事。2010 年情報処理学会山下記念研究賞受賞，日本情報科教育学会会員。



神藤 健朗 （正会員）

1997 年東京理科大学理学部第一部数学科卒業，1999 年東京工業大学大学院社会理工学研究科人間行動システム専攻修了。企業勤務を経て，2005 年より武蔵工業大学付属中学校・高等学校（現東京都市大学付属中学校・高等学校）情報科教諭，2012 年より芝浦工業大学非常勤講師勤務。修士（学術）。日本情報科教育学会，日本教育工学会各会員。



並木 美太郎 （正会員）

1984 年東京農工大学工学部数理情報工学科卒業。1986 年同大学大学院修士課程修了，同年 4 月（株）日立製作所基礎研究所入社。1988 年東京農工大学工学部数理情報工学科助手，1993 年 11 月電子情報工学科助教授，1998 年 4 月情報コミュニケーション工学科助教授。現在，東京農工大学大学院工学研究院教授。博士（工学）。オペレーティングシステム，言語処理系等のシステムソフトウェア，並列処理，コンピュータネットワーク，計算機アーキテクチャ等の研究・開発，計算機科学の教育に従事。ACM，IEEE，電子情報通信学会，ソフトウェア科学会各会員。



兼宗 進 (正会員)

1987 年千葉大学工学部電子工学科卒業。1989 年筑波大学大学院理工学研究科修士課程修了。2004 年筑波大学大学院ビジネス科学研究科博士課程修了。博士 (システムズマネジメント)。

企業勤務後、2004 年から一橋大学総合情報処理センター准教授。2009 年から大阪電気通信大学医療福祉工学部/総合情報学部を経て工学部電子機械工学科教授。プログラミング言語、情報科学教育に興味を持つ。ACM, IEEE Computer Society 各会員。