

探索、整列アルゴリズム学習支援教材の提案と評価

関下 堅也^{1,a)} 赤池 英夫¹ 中山 泰一¹

概要：高等学校において情報Ⅰが必修化されて取り扱い内容が刷新され、ソートアルゴリズムも学習するようになった。刷新以前よりプログラミングに関しては様々な学習支援の研究がなされているが、アルゴリズムに関しては例が少なく、実際の教育ではアルゴリズムの動作のアニメーションを見ることが出来る教材の利用に留まる可能性がある。そのため探索、整列アルゴリズムを理解するには教員にも学生にも支援が必要だと考え、本稿では先行研究で示唆されたアルゴリズム的思考を習得する際に必要とされる「構成要素」の考え方に従い、探索、整列アルゴリズムの学習支援を行うシステムの提案、構築、評価を行う。

A proposal and evaluation of a system to support the learning searching and sorting algorithms

1. はじめに

日本では論理的思考力の育成を目的にプログラミング教育と情報の教育に着手しており、高等学校では情報Ⅰを必修化する運びであるが中山らの調査によると必ずしも情報学についての幅広い知識や技術をもつ者が情報科の教科担任となるわけではなく、教員不足や地域格差も確認されている [1]。

情報Ⅰの高等学校教員研修用教材によると、ソートアルゴリズムが取り扱い範囲にあり、アルゴリズムの動きだけではなく、実際に動かしたときに計算量が最悪となるパターンも取り扱うこととしているが [2]、教員が習得および学生へ教授する難易度は容易くないと思われる。

情報Ⅰは 2025 年の大学入学共通テストに導入予定で [3]、過去のセンター試験問題を見ると、2008 年の情報関係基礎においてバブルソートの問題が出題されたこともあり [4]、受験においても対策の必要がある可能性がある。現在、教科書ではアルゴリズム図鑑 [5] でどのようにアルゴリズムが動くかをアニメーションで見る指導形式であるが、長瀧ら [6] の調査によると、情報に関する学習を支えるシステムを取り扱う情報処理学会の CE 研究会内ではアルゴリズムに関する学習教材は他の分野よりも支援が少ないことが課題となっている。

また、鈴木ら [7] の研究と安達ら [8] の研究によると学習の支援に当たり、プログラミングの学習には問題文とソースコードの記述に移る間に構成要素を考えるプロセスがあり、このアルゴリズムの構成でつまづくとの声もある。とくに [7] では構成要素も問題文に近い構成要素、ソースコードに近い構成要素があるとされている。杉浦ら [9] は、アルゴリズム構築能力の育成には教具を用いて手作業でアルゴリズムを体験し、体験をそのままプログラミングできるシステムを開発することが重要であるとし、学習支援効果を得た。

そこで構成要素と手作業による学習経験の利用を取り入れたふたつのシステムを提案し、探索、整列アルゴリズムの学習支援を行いたい。1 つ目は「処理手順理解システム」とし、各アルゴリズムにおいて注目すべき要素を自分で操作して流れを経験してもらうことで問題文に近い構成要素を学び、手作業による経験ができるものである。2 つ目は「実践システム」とし、ソースコードに近い抽象的な構成要素として「処理手順理解システム」で学んだ手順を細分化し、共通する処理手順や分岐条件を区別した後にセンター試験用手順記述標準言語 (DNCL) [10] のような疑似言語を用いてアルゴリズムとして記述できるようになることを目指す。

本研究の目的は新学習指導要領の施行に伴って小学校、中学校でビジュアル型プログラミングを学んだ初学者にあたる生徒が高校で情報Ⅰを履修した際に探索、整列アルゴリズムの処理手順の理解と DNCL によるプログラムの記述

¹ 電気通信大学大学院 情報理工学研究所 情報・ネットワーク工学専攻

a) s2031090@edu.cc.uec.ac.jp

ができるようになることと、はじめてプログラミングに触れる際に初心者に限らず、アルゴリズムを学習する際には構成要素の考え方が有効であるかを確認することである。

2. 関連研究

2.1 プログラミング初学者にむけたアルゴリズム的思考習得のための学習支援手法

鈴木らは、初学者はプログラミングの構成要素を組み立てる構成段階で行き詰まる問題を抱えているのではないかと考えた。構成要素とは「問題文」と「ソースコード」の間にある問題を解く際に処理の流れを確認し、その順序を考えるプロセスと定義した。この構成要素も「問題文に近い」短い文章の組み合わせによって処理手順をまとめたもの、「ソースコードに近い」抽象的な処理手順でプログラム構成を表すものに分かれる。

構成要素は問題文に近いものは自然言語で表記されたブロックを並び替えることで処理全体の流れを学ぶこと、ソースコードに近いものは問題文に近い構成要素で取り扱ったブロックに表記されている内容をプログラミングに近づけるために穴埋め形式でブロックを自然言語混じりの擬似コードにすることである。

プログラム作成の際、構成を十分に考えられる人は問題文から必要な要素を発見し、それらを満たすように機能の作成とデバッグを繰り返すことでプログラムを完成させ、躓く人は問題文をそのままコード化しようとし、なぜそのように修正するのかを理解できていない傾向があると考えた。

そのため構成要素を学習させることがプログラム作成時の手詰まりの解決になると考え、アルゴリズム的思考法を用いた学習システムを提案した結果、一定の支援効果が得られた。

2.2 アルゴリズム構築能力育成の導入教育：実作業による

概念理解の基づくアルゴリズム構築体験とその効果

杉浦らは先行研究の調査によってアルゴリズムを理解するためには手作業でアルゴリズムを体験すること、プログラミング言語を記述するための学習コストを削減してアルゴリズムの構築に注力できる環境を構築することが有効であると結論づけた。そこで教具を抽象データ型としてプログラミングできる環境を開発することで、教具を用いたアルゴリズムを体験と、体験を元にしたプログラミングとの結びつきを強め、学習効果を得た。

3. 提案するシステム

提案するシステムは問題文に近い構成要素とソースコードに近い構成要素の考え方に則って1節で述べたように2つのシステムで構成される。これらのシステムは操作のログが残り、ログと正解例を比

較することでどこで間違えたかがすぐに分かる、ログに飛ぶことで操作のやり直しをしやすいようにする。実行環境に左右されないように web ベースで開発を行う。

3.1 処理手順理解システム

各アルゴリズムにおいて特徴的な部分をシステム利用者が並び替えなどができるようにし、操作した結果をログに自然言語で表示することで問題文に近い構成要素を学ぶことができるかと期待している。

例としてクイックソートを取り上げ、図1に想定しているUIを示す。システム利用者は図1において以下の手順で操作を行い、この操作がログに表示される。

- ピボットにする値の選択する。
- 矢印は配列のどこを指しているかを示す。赤い矢印をピボット以上の値を指すようになるまで「矢印を進める」ボタンで右に進め、矢印確定ボタンを押す（この例では赤い矢印のすでにピボットを指すため、進めることなく「矢印確定」ボタンを押す）。
- 次に青い矢印をピボットより小さい値を指すようになるまで「矢印を進める」ボタンで左に進めると2が該当するので「矢印確定」ボタンを押す。
- 矢印の位置が定まったのでシステムにより交換が発生し、図2に進む。

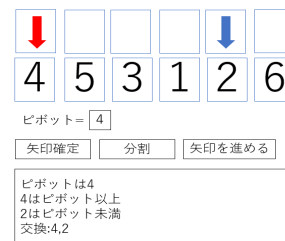


図1 処理手順理解システム1

図2は交換が発生したあとの状態で、交換が発生するとどちらの矢印も自動的に進行するようにする。このあと操作を進めていくと赤と青の矢印が重なるタイミングが訪れるので「分割」ボタンを押すことで配列を分けることができる。

誤った値を矢印で指し示して「矢印確定」をしてしまった際は、ログの部分で「6はピボットより小さい？」というように？マークがログの末尾に表示されるようにする。

3.2 実践システム

3.1節でアルゴリズムの流れを体験し、理解したという仮定のもと利用してもらおう。

例として二分探索を取り上げる。学習者には図3にある3.1で述べた手作業を反映したログが与えられる。図4に示すUIを用いて図5のように手作業で得た経験をコン

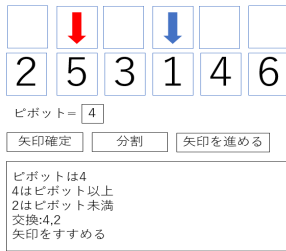


図 2 処理手順理解システム 2

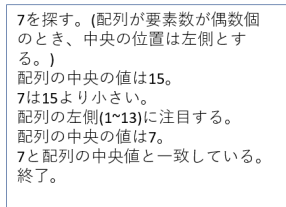


図 3 学習者に与えられるログ

コンピュータ上での表現に近づけるために穴埋めによって細分化した命令を作成してもらおう。



図 4 実践システムにおける穴埋めの UI

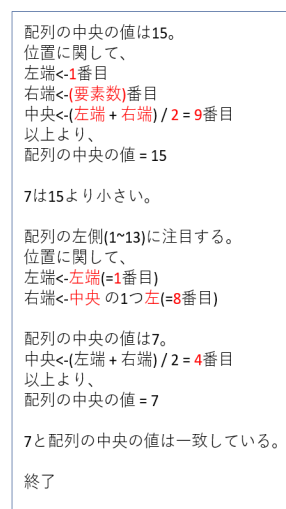


図 5 細分化した命令

図 3 にある「配列の中央の値は 15」という手作業によるログを細分化する場合、図 5 にある赤字の部分を図 4 の

「Obj」欄から適切なものをドラッグアンドドロップ形式で穴埋めする。

次に細分化した命令から共通の処理、分岐条件、終了条件などを図 6 にあるように仕分けする。

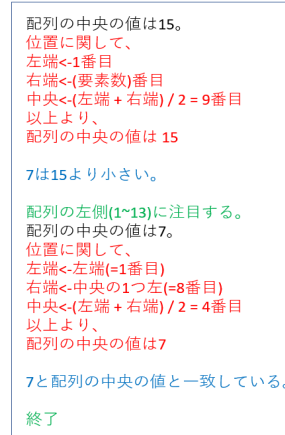


図 6 細分化した命令の区別

赤色の命令は繰り返し行う内容、青色の命令は分岐や終了に関わる条件、緑色の命令は青色の条件を受けて分岐した結果というように命令を分類する。

図 6 を受け、短冊形および穴埋めプログラミングによって擬似言語を用いて図 7 のようにアルゴリズムを作成してもらおう。

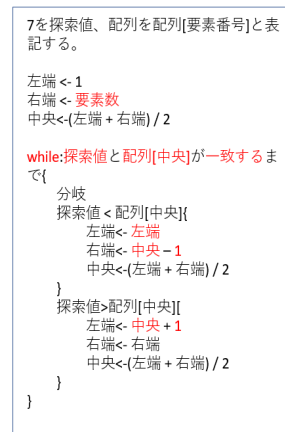


図 7 擬似コード化

こちらはソースコードに近い抽象的な構成要素の学習と擬似言語によるアルゴリズムの記述を行うときに、手作業で行った操作をコンピュータ上で表現するための細分化によって実体験とコンピュータ上での処理との連携を図ることで学習効果につながると考えている。

4. 評価方法

対象は情報Ⅰを習う高校生を想定しており、特にアルゴリズム図鑑を用いて学習した生徒が本システムを用いてソートアルゴリズムを習得できているかをテストで測定する。

テストの内容はソートアルゴリズムに関する説明の穴埋め、あるループ段階までソートされた数字列を示して何回目のループが終了したかを書いてもらう、短冊形プログラミングで疑似コード化をしてもらうことを想定している。

5. 今後の課題

現在は提案システムを製作中であり、完成しだい実験を行う予定である。

また、被験者数、被験者にソートアルゴリズムを学習してもらう時間、確認テストとして出題する問題の作成と妥当性の検証など実験に関する準備についても同時に進めていく。

参考文献

- [1] 中山泰一：高等学校情報科の教員採用と免許外教科担任の現状，情報教育資料，No.50，pp14-16(2020)。
- [2] 文部科学省：高等学校教員研修用教材 情報Ⅰ 第3章コンピュータとプログラミング，入手先 (https://www.mext.go.jp/content/20200722-mxt_jogai02-100013300_005.pdf) (参照 2021-06-26)。
- [3] 独立行政法人 大学入試センター：令和7年度以降の試験，入手先 (https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7ikou.html) (参照 2021-06-26)。
- [4] 独立行政法人 大学入試センター：情報関係基礎 2008 本試験 N 問題，入手先 (<https://drive.google.com/drive/folders/140pQJOKWzYH2-NvzPCyQdqFPHcZhCSOa>) (参照 2021-06-26)。
- [5] Algorithms Project (石田保輝，光森裕樹)：アルゴリズム図鑑，入手先 (<http://algorithm.wiki/ja/app/>) (参照 2021-08-02)。
- [6] 長瀧寛之，白井詩沙香：情報教育における学習支援ツールの提供・提案状況に関する現状調査，情報処理学会研究報告，vol.2020-CE-154，No.15，pp1-5(2020)。
- [7] 鈴木優実，松村敦，宇陀則彦：プログラミング初学者にむけたアルゴリズム的思考習得のための学習支援手法，情報処理学会研究報告，vol.2018-CE-143，No.6，pp1-7(2018)。
- [8] 安達一寿，中尾 茂子：プログラミング学習における学生をつまずき箇所の分析，教育情報研究，vol.10，no.4，pp11-20(1995)。
- [9] 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：実作業による概念理解の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，pp3409-3427(2008)。
- [10] 独立行政法人大学入試センター：センター試験用手順記述標準言語 (DNCL) の説明 (2020.01)，入手先 (<https://www.dnc.ac.jp/albums/abm00039945.pdf>) (参照 2021-07-29)。