

## 原著論文

# 「情報 I」を対象としたブロック図と GBS 理論を 組み合わせたアルゴリズム学習法

高田悠矢<sup>1)</sup>, 古川 宏<sup>2)</sup>

<sup>1)</sup> 筑波大学情報学群, <sup>2)</sup> 筑波大学システム情報系

## An algorithm learning method combining block diagram and GBS theory for Informatics I

Yuya TAKADA<sup>2)</sup>, Hiroshi FURUKAWA<sup>2)</sup>

<sup>1)</sup> School of Informatics, University of Tsukuba,

<sup>2)</sup> Institute of Engineering, Information and System, University of Tsukuba

**概要:** 現状のプログラミング学習では、プログラムの意味を考えず例題のコードなどを新規問題にもそのまま当てはめようとする、記憶に頼りすぎた学習に陥りやすく、求められる論理的思考力が身につけにくいことが指摘されている。本研究では、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の支援を行うことでその課題の解消が期待されているブロックの並び替え問題機能と、その機能の課題と思われる支援不足の解消のために効果的にフィードバック機能を実装できる GBS 理論を組み合わせた学習法を提案した。提案手法で作った教材を用いた評価実験では、主観評価において一定の支援効果が示唆された。

**Keywords:** algorithm, GBS theory, learning support, Informatics I

**キーワード:** アルゴリズム, GBS 理論, 学習支援, 情報 I

### 1. はじめに

高等学校学習指導要領において、令和 4 年度より必修科目「情報 I」が新設されたのに続き、令和 7 年度大学入学共通テストでは、これまでの内容に加えて「情報」が採用されることとなった。出題方針としては、情報 I で学ぶアルゴリズムをそのまま出題するのではなく、社会や身近な生活の中の題材、及び受験者にとって既知ではないものも含めた資料等に示された事例や事象について、問題の発見・解決に向けて考察する力を問うことが検討されている[1]。文部科学省より公開された大学入学共通テスト「情報」サンプル問題[2]や令和 7 年度に実施された情報 I の共通テスト[3]のアルゴリズム・プログラミング分野では、会話文を基に大学入学共通テスト手順記述標準言語(Daigaku Nyushi Center Language[4]、以下、「DNCL」と表記)で作成することが求められるなど、知識だけでなく、論理的思考力をもって会話文からアルゴリズムを形にする能力(以下、「アルゴリズムについての論理的思考力」と表記)が求められている。共通テストという利用者が多い場でも問われるアルゴリズムについての論理的思考力を養うことは重要であると考え。

しかし、現状のプログラミング分野の学習では、例題をその通り打ち込んで動かしてみるだけ、1 つの例題を丁寧に説明し試験もそれから出すなどの方法が度々見られている[5]が、そ

れはプログラムの意味を考えず、記憶頼りにサンプルや類似問題のプログラムを新規問題にもそのまま当てはめようとする暗記型学習に陥りやすく、十分なアルゴリズムについての論理的思考力を養うことが難しいことが指摘されている[6]。

この課題に対処した研究として、飯田ら[6]の研究がある。飯田らは、目的を達するための処理を基本的な操作に分解し、それらの順序を意識する思考法をアルゴリズム的思考法と定義した。そして、このような思考法を用いて学習する思考型学習と、記憶頼りにサンプルや類似問題のプログラムを新規問題にもそのまま当てはめようとする暗記型学習を比較し、記憶に頼りすぎた暗記型学習は自分なりの考えをアルゴリズムの形にする必要がある段階で限界を迎えるため、思考型学習へ移行させる重要性を示した。また、アルゴリズム的思考法を支援する際にコンピュータを使うことが好ましくない例として、正誤判定などを利用して意味を考えずに修正を行って課題を解決する実行順りの学習に陥ることや、プログラミング言語の学習(特に文法の習得)に注意をとられることなどを挙げた。これらの問題に対して、紙にフローチャートを書く方法を実践し、一定の効果を示したが、暗記型学習の傾向が見られる回答も見られていた。これは、実行結果やフィードバックなどの手がかりがないことから、フローチャートの作成でつまづいた際に、問題文が似ているフローチャートを再利用して解いたことが原因であった。また、学習法として紙のフローチャートを扱うのは、新たにフローチャート用の記述を覚える負担や、回答が 1 つに定まらないため適切な採点が難しく、時間を要するという課題がある。情報 I の授業では、8 割以上の教員が授業時間に不足を感じているという実状があり[7]、この手法は採用が難しい。

そのため、アルゴリズムについての論理的思考力を養うこと

2025 年 2 月 17 日受理。(2025 年 3 月 1 日スマートライフ学会  
2025 年大会にて発表)

著者紹介先: 〒305-8573 茨城県つくば市天王台 1-1-1  
システム情報系 認知支援システム研 高田悠矢

ができる別のアルゴリズム学習法が必要であると考え。本研究では、新たなアルゴリズム学習法を提案するとともに、提案手法で実装した教材と既存手法で作成した教材を用いた評価実験を通してその有効性や改善点を検討する。

## 2. 関連研究

この章では、提案するアルゴリズム学習法に関連する研究を紹介する。

鈴木ら[8]は、問題文を読み解いてソースコードに落とし込む際に、その間に処理の流れを確認しその順序を考えるプロセスである、「構成要素」を考えるというプロセスがあり、その支援を行うことで、プログラミング初学者がプログラムの構成段階で行き詰まる問題に対処できると考えた。「構成要素」を考えるプロセスの中にも段階があり、「問題文に近い」構成要素プロセスと「ソースコードに近い」構成要素があるとした。「構成要素」を考えるプロセスを2段階で支援するために、プログラムの処理の流れを考える「ブロック図の並び替え」機能と、具体的な処理手順を考える「穴埋め問題」機能を有した学習システムを提案した。評価実験では、過半数の参加者がこれらの機能により処理の流れを考えることで、問題が解きやすくなったと回答するなど初学者への一定の支援効果が見られたが、普段の解き方と異なり解きにくくなったと回答した参加者もいた。鈴木らの手法の課題として、問題に誤った際に実行結果やフィードバックなどがなく、手詰まりになった学習者が正誤判定を利用した実行順りの学習に陥りやすいという課題があったと思われる。そのため、実行結果やフィードバックを表示する機能を実装することで、この課題に対処できると考えた。フィードバック機能を効果的に実装できる学習法として、GBS理論が挙げられる。

GBS(Goal-Based Scenario)理論とは、R. C. Schank によって提唱された教授法であり[9]、現実的な文脈の中で「失敗する事により学ぶ」学習環境を擬似的に与えるための物語を構築する理論のことである。これを用いる際は、学習者が自然と知識やスキルを身につけることができるように学習教材の設計を行う。GBS 理論は、学習目標・使命・カバーストーリー・役割・シナリオ操作・情報源・フィードバックの 7 つの構成要素で構成され、GBS を設計するには 7 つの要素全てが必要になる。図 1 に、GBS 理論の構成要素を示す。

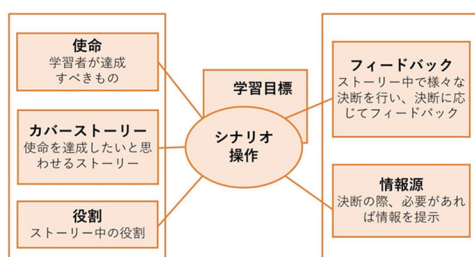


図 1 GBS の 7 つの構成要素[10]

朴らの研究[11]では、ワープロなどのアプリケーションの基本スキルの習得や活用力の向上のための効果的・効率的・魅力的な指導法の開発として、GBS 理論に基づく教材を用いた。朴らは、教科書通りの操作や脈絡のない練習問題の繰り返しだけではその能力の育成が難しいため、現実的な場面設定の中で学習者自らが試行錯誤し、意味ある作業に取り組むというプロセスが大事であると考えて、GBS 理論を活用した。授業での実践を通じて開発した教材を検証した結果、アプリケーションの基本スキルの習得や活用力の向上において、一定の効果が認められるとともに、満足感や学習意欲にお

いて肯定的な反応が示された。

## 3. 提案手法

本章では、アルゴリズムについての論理的思考力を養う学習法の提案と、それぞれの要素で期待する効果について説明する。

### 3.1 ブロック図の並び替え機能

プログラムの処理の流れを考えることは、会話文から DNCL で表現されたアルゴリズムの形を作成する際に必要な思考過程であるため、プログラムの処理の流れを考えるブロック図の並び替え機能は、アルゴリズムについての論理的思考力を養うのにも効果的であると期待する。また、この手法はプログラミング言語の記述を用いないため、本質的でないミス(スペルミスやセミコロンミスなど言語特有のエラー)が起きず、飯田らが指摘するプログラミング言語の学習に注意をとり、アルゴリズムについて考える機会が減りやすい課題の解消も期待できる。穴埋め問題機能については、並び替え機能で不正解の選択肢を用意すれば、並び替え機能だけでも具体的な処理手順を考える機会を与えられることから、本研究では扱わない。

この機能の課題として、実行結果やフィードバックなどの手掛かりがないため、作成したブロック図が誤っていた場合の支援が足りないことが挙げられる。手掛かりがない場合、意味を考えず類似の問題を当てはめる学習に陥りやすく、アルゴリズムについての論理的思考力を養う効果が薄れる。そのため、この課題を解消するために、実行結果やフィードバックを表示する機能と合わせて実装することが必要だと考える。

### 3.2 GBS 理論に基づくフィードバック機能

フィードバック機能を実装する枠組みとして、フィードバック機能と相性が良い GBS 理論を利用することが適当であると考え。GBS 理論によりフィードバック機能を実装することで、意味を考えず類似の問題を当てはめる学習を抑制し、アルゴリズムについての論理的思考力を養う効果を高めることや、学習意欲が向上することを期待する。情報 I の現場では、授業前からプログラミングへ関心を持っている生徒は半分に満たないことが多く[7]、教材の取り組みやすさの支援も重要である。学習意欲の向上が見られやすい GBS 理論の適用はこの点においても効果的だと判断した。アルゴリズム学習に対して GBS 理論を適用した研究はまだ見られておらず[12]、アルゴリズム学習に GBS 理論を適用する点で新規性がある。

### 3.3 アルゴリズムについての理解度を確認する出題機能

実行結果やフィードバックの機能を実装した際、考える手が増えと共に、これらを利用して自分で意味を考えずに修正を行って課題を解決する実行順りの学習に陥る可能性も生じる。その可能性を抑制するために、並び替え問題の後に作成したアルゴリズムについての理解度を問う問題を毎回挟むことを新たに提案する。これにより、毎回ここでアルゴリズムについての見直しを要求できるので、意味を考えない実行順りの学習を抑制し、アルゴリズムについての論理的思考力を養う効果を高めることを期待する。

提案手法はこれら 3 つの機能を全て組み合わせることで、それぞれの機能の持つ課題を解消し、アルゴリズムについての論理的思考力の習得の支援をより効果的に行えるようになることを期待する。一方、フィードバックの内容次第では、それ頼りにブロック図の並び替え問題を機械的に解き、プログラムの

処理の流れを考える効果が薄れる可能性も考えられる。今回の実験ではフィードバックでは直接の修正箇所のヒントは与えず、論理エラーなのか構文エラーなのかといった間接的なヒントに留めることで、機械的な修正を実現する機能ではなく、考えの手がかりを与える機能となるように注意して実装した。

#### 4. 教材の実装

本章では、評価実験で使用する、提案手法に基づいて実装した教材(以下、「提案教材」と表記する)について述べる。

##### 4.1 教材の概要

提案教材は GBS 理論に基づくシナリオ型の教材であり、以下の流れで学習を進める。

- ① DNCL の構文知識や、探索アルゴリズムについての学習をする。
- ② DNCL ブロック図の並び替え問題を解く。
- ③ ②で作成したアルゴリズムについての理解度を問う問題を解く。
- ④ ②,③を合計 3 回繰り返す。

また、学習は教材単体で完結できるように設計し、学習時間は 1 時間を想定した。また、提案教材は html, css, javascript ファイルのみで構成し、ウェブブラウザ上で操作できるように実装した。

GBS の 7 つの構成要素については、次のように設定した。

- ・学習目標: アルゴリズムについての論理的思考力を養う。
- ・役割: 情報 I の復習をする高校 3 年生の弟。
- ・カバーストーリー: 情報 I の受験対策教材を解いて学習していたが、間違いが多かった。そこでプログラミングに詳しい兄に協力を求め、兄とともに情報 I を 1 から勉強し、受験対策教材で間違えた問題を解き直していく。
- ・使命: 情報 I の受験対策教材の問題を解くことで、受験に対して自信をもつ。
- ・シナリオ操作: 登場人物との会話で進行する。課題解決のために質問されるので、その回答で正しい選択肢の選択や並び替えを行う。
- ・フィードバック: シナリオ操作に応じて異なるものを示す。例えば、並び替えに応じたコードの実行結果や、どのような間違え方なのかについてのフィードバックコメントを与える。
- ・情報源: 登場人物との会話などで得る。また、学習内容について簡単に復習できるように、学習内容を一覧できる機能も搭載する。

学習内容については、「【高等学校情報科「情報 I」教員研修用教材(本編)】第 3 章」[13]を基に作成した。実験の都合上学習時間を 1 時間としたため、演算子や構文の知識、探索アルゴリズムなど、アルゴリズムを考えるうえで基本となる部分のみを扱った。教員研修用教材ではプログラミング言語は Python で学習しているが、本研究は特に情報 I に焦点を当てているため、提案教材では DNCL で学習する。DNCL は構文などが日本語で記述されるため、言語の記述を用いた学習には適さないが、処理内容が直接的に表現され分かりやすいため、プログラムの処理の流れを考えるブロック図の並び替え問題とも相性が良いと思われる。

##### 4.2 教材の実装

教材は、4.1 で示した学習の流れ①②③にそって、シナリオを進行させて学習していく。

- ①: 兄と文法知識などを 1 から学習するというシナリオで実装した。シナリオの進行画面は(1)のように実装した。
- ②: 弟がシナリオの中で間違えていたアルゴリズムの問題を、

ブロック図の並び替え問題という形式で解き直すというシナリオで実装した。出題機能は(2)のように実装した。

- ③: ②の問題を弟がシナリオの中でどのように間違えていたかを表示し、それが誤りである理由、もしくはどこを変更すれば正しいプログラムとなるかを説明することで、アルゴリズムの理解度を確かめるというシナリオで実装した。出題機能は(3)のように実装した。

##### (1) シナリオ進行画面

学習時の基本画面となる、シナリオ進行画面は図 2 のように実装した。画面は登場人物、会話内容を表示するメッセージボックス、画面上部の説明画像などから構成される。基本的に次へボタンを押すことでシナリオを進めて学習をしていき、戻るボタンや画像一覧ボタンで復習を行う。復習のために戻るボタンで逐一シナリオを戻すのは手間が大きいので、画像一覧ボタンで説明画像や並び替え問題の一覧を表示するページを開けるようにすることで、任意のタイミングで復習を行いやすいように配慮した。

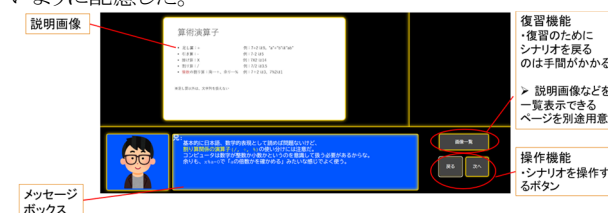


図 2 学習の基本画面であるシナリオ進行画面

##### (2) ブロック図の並び替え問題

ブロック図の並び替え問題は図 3 のように実装した。選択肢の欄には完成形のプログラムを分割し順番を入れ替えたものがある。学習者は解答欄に選択肢を正しい順番で入れて DNCL のプログラムを構成する。実行ボタンを押すことで、回答に応じたプログラムの実行結果およびフィードバックが返される。不正解の選択肢の搭載については、選択肢の欄の選択肢の数が多くなりすぎると、まず選択肢を把握することにつまずき、期待する学習効果が得られにくくなると思われる。そのため、一部の選択肢について、選択肢の欄では処理の一部を「???」で表現し、その選択肢を使う際は更にそれに当てはまる処理内容を選ぶという機能で実装することで、選択肢の欄の選択肢の数は最小限に抑えて不正解の選択肢を搭載し、プログラムの処理の流れを考える効果が薄れることなく具体的な処理手順も考えられるように実装した。

また、DNCL では、構文にインデントを付けて処理範囲を見やすくすることが可能だが、インデントがなくても構文の記述により厳密に処理範囲を特定できる。この並び替え問題ではプログラムの処理の流れを考えることが目的であるため、選択肢に応じたインデントを付けるという可読性向上のための操作は求めなかった。

教材が時間内に終わらない可能性を考慮し、ブロック図の並び替え問題については、正答できなくてもシナリオを進行できるように実装した。

| 解答欄 | 選択肢                        |
|-----|----------------------------|
| -   | num ← i X 10 + j;          |
| -   | i を 0 から 2 まで 1 ずつ増やしなが    |
| -   | を繰り返す (i の繰り返し)            |
| -   | j を 1 から 10 まで 1 ずつ増やしなが   |
| -   | を繰り返す (j の繰り返し)            |
| -   | もし (num % 15 == 0) ならば     |
| -   | ! { ??? }                  |
| -   | を実行し、そうでなければ               |
| -   | を実行する                      |
| -   | もし (??? ) ならば              |
| -   | ! { ??? }                  |
| -   | を実行し、そうでなければ               |
| -   | を実行する                      |
| -   | もし (??? ) ならば              |
| -   | ! { 'bz' を表示する; }          |
| -   | を実行し、そうでなければ               |
| -   | ! { num と ' ' (空白) を表示する } |
| -   | を実行する                      |
| -   | (選択肢を外す用)                  |

図 3 ブロック図の並び替え問題の解答欄と選択肢の実装画面

また、図 4 に実行例を示す。実行結果の欄に作成したプログラムの実行結果を示し、フィードバックの欄に作成したプログラムがなぜ間違っているのかなどについて、コメントを出す。

| 問題                                                                                                                                                                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ・ num は 1 から順に 1 ずつ増やし、各数字 1 回の出力をする (30 まで)<br>・ num が 3 の倍数のとき、"bz" を出力する<br>・ num が 5 の倍数のとき、"fz" を出力する<br>・ なお、num が 3 かつ 5 の倍数のときは、"FB" を出力する<br>・ num が他の数字のとき、num を出力する<br>・ 出力例: 「..., 8, bz, fz, 11, bz, 13, 14, FB, 16, ...」<br>* 1 つだけ使わない選択肢があります |
| 実行!                                                                                                                                                                                                                                                           |
| <b>実行結果:</b><br>1 2 bz 4 fz 5 bz 7 8 bz fz 10<br>11 bz 13 14 FB 16 17 bz 19 fz 20<br>bz 22 23 bz fz 25 26 bz 28 29 FB                                                                                                                                         |
| <b>フィードバック:</b><br>出力が 30 回でないみたいだ。どの数字で余計な出力が出ているかをまず確認し、残っている選択肢と今の解答をよく見て、思い描いたアルゴリズムと違うところを探してみよう。                                                                                                                                                        |
| 画像一覧    ストーリーに戻る                                                                                                                                                                                                                                              |

図 4 ブロック図の並び替え問題の実行例

### (3) アルゴリズムについての理解度を問う問題

シナリオとしては、弟(教材の役割)がシナリオ上で作成した間違いを含むプログラムについて、ブロック図の並び替え問題形式で 1 から解き直した後、間違いを含むプログラムを表示する。その誤ったプログラムに対して、何が間違いであるのか、もしくはどこを変更すれば正しい動作を行うプログラムとなるかを問う選択問題を出題することで実装した。具体例を図 5 に示す。この問題においても、フィードバック機能を実装した。

| 誤ったプログラム                                                                                                                                                                                | 間違い                                                                                                                                                                                                                                                                 | 選択肢                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> num ← 1 while num ≤ 30   num ← num + 1   if num % 3 == 0     print "bz"   if num % 5 == 0     print "fz"   if num % 15 == 0     print "FB"   print num end while           </pre> | 1. num は 1 から順に 1 ずつ増やし、各数字 1 回の出力をする (30 まで)<br>2. num が 3 の倍数のとき、"bz" を出力する<br>3. num が 5 の倍数のとき、"fz" を出力する<br>4. なお、num が 3 かつ 5 の倍数のときは、"FB" を出力する<br>5. num が他の数字のとき、num を出力する<br>6. 出力例: 「..., 8, bz, fz, 11, bz, 13, 14, FB, 16, ...」<br>* 1 つだけ使わない選択肢があります | 1. num が 1 から順に 1 ずつ増えたり減ったりする<br>2. num が 3 の倍数のとき、"bz" を出力する<br>3. num が 5 の倍数のとき、"fz" を出力する<br>4. num が 3 かつ 5 の倍数のときは、"FB" を出力する<br>5. num が他の数字のとき、num を出力する<br>6. 出力例: 「..., 8, bz, fz, 11, bz, 13, 14, FB, 16, ...」<br>* 1 つだけ使わない選択肢があります |
| フィードバックの表示<br>対照教材で間違えていた解答がどうなるか、さっきの正解と違うところは赤文字にしてみました。改めて読んでみよう。それによって何が間違っているか、もしくはこれがどうすれば正しいになるか説明できるか? (右上の選択肢を選んで「確認の質問」に決定を押してね)                                              |                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                   |

図 5 理解度確認問題の画面

## 4.3 比較教材

評価実験にて、提案教材の効果を比較するために、既存手

法であるブロック図の並び替え機能のみを用いた教材(比較教材)を作成した。構文知識などの学習内容は提案教材と同じである。比較教材は毎回のアルゴリズム確認問題がない分学習量が少なくなるため、ブロック図の並び替え問題を 1 問多く出題することで学習量が公平になるように作成した。

## 5. 評価実験

### 5.1 実験方法

越谷北高等学校に協力してもらい、越谷北高等学校の在校生から 20 名実験に参加していただいた。本実験計画は、筑波大学システム情報系倫理審査委員会により承認されている(審査承認番号:2024R920)。

実験参加者は以下の共通の属性をもつ

- ・情報 I のプログラミング分野について学習済み
- ・言語としては Jupyter Notebook 上で、Python を学習
- ・DNCL での表現についてもある程度補足されている
- ・謝金は支払われない

実験は次の手順で、休憩や説明の時間も含め合計約 2 時間、越谷北高等学校のパソコン室で実施した。提案教材の表示や動作について問題は生じなかった。

#### (1) 事前アンケート

参加者の属性を得るため、事前アンケートを実施した。目的は、実験にて提案手法を用いた教材を使う群(以降、「提案群」と表記)と、比較教材を使用する群(以降、「比較群」と表記)に振り分ける為である。アンケートの内容は次の通りで、選択式である。

- ・設問 1. 性別
  - 男女の他に、「その他、回答したくない」の選択肢も用意したが回答者はいなかった
- ・設問 2. 学年
- ・設問 3. 1 日のパソコンの利用時間
  - 選択肢:「1:30 分未満」、「2:30 分~1 時間」、「3:1 時間~2 時間」、「4:2 時間以上」
- ・設問 4. 中学で、プログラミングについて学んだことがあるか
- ・設問 5. 高校で、プログラミングについて学んだことがあるか
- ・設問 6. 「アルゴリズム」についての理解度
  - 選択肢:「1:知らない」、「2:聞いたことがある」、「3:簡単なアルゴリズムならわかる」、「4:簡単なアルゴリズムなら説明できる」
- ・設問 7. アルゴリズムの理解についての自信はあるか
  - 選択肢:「1:全くない」、「2:少しない」、「3:少しある」、「4:かなりある」

#### (2) 教材を用いた学習

実験参加者 20 名を、提案群と比較比較群に、10 名ずつ振り分けて約 1 時間学習の自己学習を実施した。提案群と比較群で参加者の属性の偏りが起きないように、事前アンケートの結果より男女比やパソコンの利用時間、アルゴリズム、プログラムについての理解度や自信などについて、提案群と比較群で大きな差が出ないように振り分けた。

#### (3) 確認テスト

学習の効果を見るため、文部科学省より公開された大学入学共通テスト「情報」サンプル問題[2]の内、プログラミング分野についての出題である第 2 問を出題した。共通テストを模した問題であり、単純な文法知識だけでなくアルゴリズムについての論理的思考力についても問われる。

#### (4) 事後アンケート

主観評価を得るため、紙媒体で事後アンケートを実施した。提案群で実施したアンケート内容を次に示す。提案群、比較群とも最後 2 つの設問は自由記述、他の設問は「1: そう思わない」、「2: どちらかといえばそう思わない」、「3: どちらかといえばそう思う」、「4: とてもそう思う」の 4 段階評価である。設問 1~10 は共通の設問であり、設問 11 以降は提案群と比較群の間で設問が異なる。

#### 共通質問

- ・設問 1. 学習教材の説明の表現は分かりやすかったですか
- ・設問 2. 自分から取り組みたいと思える内容でしたか
- ・設問 3. 学習の時間は適切でしたか
- ・設問 4. アルゴリズムの説明は分かりやすかったですか
- ・設問 5. 確認問題は理解の役に立ちましたか
- ・設問 6. 並び替え問題は理解の役に立ちましたか
- ・設問 7. 学習時の並び替え問題の難易度は適切でしたか
- ・設問 8. 確認テストは理解の役に立ちましたか
- ・設問 9. 確認テストの難易度は適切でしたか
- ・設問 10. 情報 I のプログラム作成について自信はつきましたか

#### 提案群のみの質問

- ・設問 11. 確認問題を解くうえで、フィードバックが表示されるのは役に立ちましたか
- ・設問 12. 並び替え問題を解くうえで、実行結果が表示されるのは役に立ちましたか
- ・設問 13. 並び替え問題を解くうえで、フィードバックは役に立ちましたか
- ・設問 14. 並び替え問題の後の、そのアルゴリズムについての理解度を問う問題は理解の役に立ちましたか
- ・設問 15. シナリオ形式の教材は通常の教材より取り組みやすいと感じましたか
- ・設問 16. 操作方法は分かりやすかったですか
- ・設問 17. このようなウェブサイトがあったら利用したいと思えますか
- ・設問 18. 学習や実験全体を通して不明点があれば自由に記入してください
- ・設問 19. その他意見・感想があれば自由に記入してください

#### 比較群のみの質問

- ・設問 11. 並び替え問題を解くとき、選択肢に応じた実行結果が表示される機能があったら利用したいですか
- ・設問 12. 確認問題や並び替え問題を間違えた際、なぜ違うのかなどのフィードバックがあったら利用したいですか
- ・設問 13. 情報 I を学べるウェブサイトがあったら利用したいと思えますか
- ・設問 14. 学習や実験全体を通して不明点があれば自由に記入してください
- ・設問 15. その他意見・感想があれば自由に記入してください

## 5.2 事前アンケートの結果

確認テストの結果に特に影響する属性と思われる設問 6, 7 を優先して、各属性がなるべく均等となるように参加者を提案群と比較群に振り分けた。振り分けた結果の提案群と比較群の男女比やアルゴリズムについての理解度や自信についての平均値を表 1 に示す。

表 1 事前アンケートの振り分け結果

| 平均  | 設問1     | 設問2 | 設問3 | 設問4  | 設問5  | 設問6 | 設問7 |
|-----|---------|-----|-----|------|------|-----|-----|
| 提案群 | 男:女=5:5 | 1.4 | 1.4 | ない1名 | 全員ある | 2.1 | 1.5 |
| 比較群 | 男:女=6:4 | 1.7 | 1.7 | ない3名 | 全員ある | 2.1 | 1.5 |

## 5.3 確認テストの結果

図 6 に設問毎の正答者数を示す。設問イとウ、タとチ、ツとテは各々 2 問を完答で正答とする。

各設問について、フィッシャーの正確 2 乗検定を実施したが、有意水準  $\alpha = 0.05$  での有意差は見られなかった。16 点満点中、平均点は提案群 12.0 点、比較群 12.1 点であった。

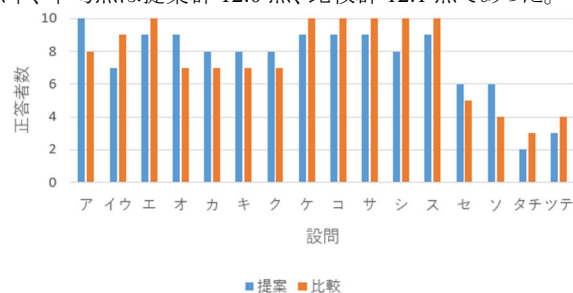


図 6 確認テスト結果

## 5.4 事後アンケートの結果

図 7 に後アンケートの共通質問の結果を示す。

各設問について、カイ二乗検定を実施したが、有意水準  $\alpha=0.05$  での有意差は見られなかった。

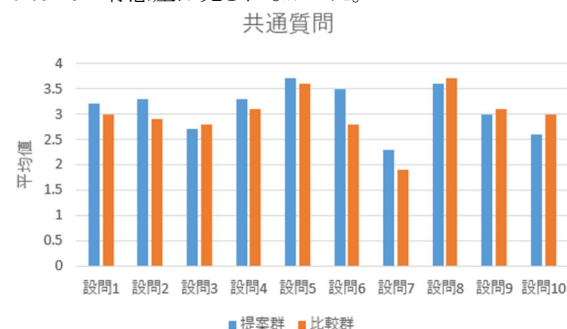


図 7 共通質問の結果

表 2、表 3 に提案群、比較群で実施した設問 11 以降の平均値や標準偏差を示す。

表 2 提案群のみの質問の結果

| 提案   | 設問11     | 設問12     | 設問13     | 設問14     | 設問15 | 設問16     | 設問17     |
|------|----------|----------|----------|----------|------|----------|----------|
| 最大値  | 4        | 4        | 4        | 4        | 4    | 4        | 4        |
| 最小   | 1        | 2        | 2        | 2        | 3    | 1        | 2        |
| 平均   | 3.6      | 3.3      | 3.3      | 3.4      | 3.5  | 3.6      | 3.4      |
| 標準偏差 | 0.916515 | 0.640312 | 0.781025 | 0.663325 | 0.5  | 0.916515 | 0.663325 |

表 3 比較群のみの質問の結果

| 比較   | 設問11     | 設問12 | 設問13     |
|------|----------|------|----------|
| 最大値  | 4        | 4    | 4        |
| 最小   | 2        | 2    | 1        |
| 平均   | 3.4      | 3.8  | 3.4      |
| 標準偏差 | 0.663325 | 0.6  | 0.916515 |

## 5.5 考察

### (1) GBS 理論による学習意欲の考察

自分から取り組みたいか(設問 2)について、比較群は 7 名から肯定的だったのに対して提案群では 10 名全員から肯定的であったことや、シナリオ教材は通常の教材より取り組みやすいか(設問 15)について、それぞれ提案群で全員から肯定さ

れたことから、アルゴリズム学習において、GBS 理論によるシナリオ形式の学習は親しみやすさや学習意欲の向上に貢献したと思われる。

## (2) GBS 理論に基づくフィードバック機能の考察

ブロック図の並び替え問題において実行結果、フィードバックは役に立ったか(提案群の設問 12,13)、あったら利用したいか(比較群の設問 11,12)について、それぞれ 10 名中 8,9 名が肯定したことから、GBS 理論に基づくフィードバック機能が支援不足の課題の解消として効果的であったと思われる。

## (3) ブロック図の並び替え問題機能についての考察

事後アンケートにて、提案群では全員、比較群では 10 名中 7 名が役に立ったと回答しており、確認テストの正答率も両群共 75%と高かったことから、並び替え機能はアルゴリズムについての論理的思考力を養うのに貢献したと思われる。一方、確認テストにおいて、提案群と比較群の間で有意差は見られなかったことから、フィードバック機能が並び替え機能の効果を高めたことは確認できなかった。これは、提案手法ではフィードバック機能により初学者でもより容易に解き直しが行えるようになりプログラムの処理の流れを考える効果が高まった反面、自らシミュレーションして実行結果を考えるプロセスがなくなり、このプロセスから得られる思考力の向上効果が小さくなったことなどが考えられる。

## (4) 理解度を確認する出題機能の考察

アルゴリズムについての理解度を問う問題(設問 14)について、10 名中 9 名が理解の役に立ったと回答したことから、この機能によりアルゴリズムを見直すことで実行順の学習が抑制され、アルゴリズムについての論理的思考力を養うのに貢献した可能性が高いと思われる。

## (5) 操作性などの考察

提案教材の表現や操作方法の分かりやすさ(設問 1,16)、についての主観評価は高かった。また、実験中に教材の操作方法や学習の仕方が分からず学習が止まっている生徒は見られず、質問されることもなかった。よって、教材は学習を教材単体で完結できるつくりになっており、提案手法は自己学習教材にも適用できる可能性が高いことが示された。また、情報 I を学べる web サイトを利用したいか(提案群の設問 17、比較群の設問 13)について、それぞれ 10 名中 9 名が肯定した。提案手法を用いた教材は web 上で実装可能なため、この点においても提案手法は有用である可能性が高いことが示された。

## (6) 改善すべき点の考察

ネストが深いプログラムの並び替え問題において、構文エラーを起こし、スムーズに学習できなかった参加者が見られた。図 8 にエラーの一例を示す。これは、ネストが深い問題に対して、インデントなどによる可読性向上の支援がなかったために、構造を把握しにくかったことが原因であると思われる。よって、並び替え問題の実装において、回答結果に応じて自動でインデントを付けるなどの可読性向上の支援が改善すべき点として確認された。

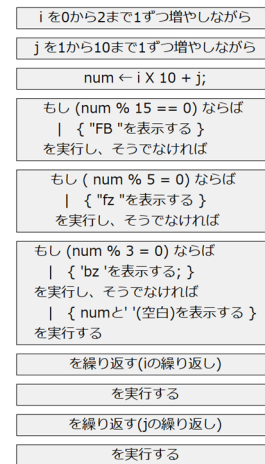


図 8 並び替え問題で見られた構文エラーの例

## (7) 提案手法の考察

以上の考察の結果より、提案手法は学習意欲の向上や論理的思考力の効果的な習得が期待できるアルゴリズム学習法であり、論理的思考力を養う教育の実現に貢献できる可能性が高いことを示すことができた。一方、今回の実装では並び替え問題において可読性が低くスムーズな学習が難しいという課題が見られ、インデントを付けるなどの可読性向上の支援が改善すべき点として確認された。

## 6. まとめ

本研究では、既存手法であるブロック図の並び替え機能で見られた支援不足の課題を解消することで、より効果的にアルゴリズムについての論理的思考力を習得できる学習法の提案とその評価をすることを目的とした。アルゴリズム学習法として、ブロック図の並び替え機能、支援不足の解消や学習意欲の向上が期待できる GBS 理論に基づくフィードバック機能、プログラムの意味を考えない実行順の学習を抑制するアルゴリズムについての理解度を確認する出題機能、という 3 つの要素を組み合わせたものを提案した。また、提案教材と比較教材を用いた評価実験を実施して、提案手法の有効性や問題点を検討した。

確認テストの結果や事後アンケートの共通質問の主観評価において、提案群と比較群の間での有意差は見られず、提案手法が既存手法より効果的であることを示すことはできなかった。しかし、提案手法の各要素について高い主観評価を得られ、確認テストの正答率も高かったことから、提案手法はアルゴリズムについての論理的思考力を養うのに一定の学習支援効果があることが示唆された。また、主観評価からフィードバック機能がブロック図の並び替え機能の課題である支援不足の解消に貢献することや、GBS 理論を用いたアルゴリズム学習が学習意欲や取り組みやすさの向上に貢献することが示唆され、提案手法を用いた学習が論理的思考力を養う教育の実現に貢献できる可能性が高いことを示した。

一方で、ブロック図の並び替え機能の実装方法については改善すべき点があったことが示された。改善案として、並び替え結果に応じて自動でインデントを付ける機能を追加することなどが挙げられた。

## 参考文献

- [1] 文部科学省: 令和7年度大学入学者選抜に係る大学入学共通テスト問題作成方針, web サイト (最終閲覧日:2025 年 2 月)
- [2] 文 部 科 学 省 : サ ン プ ル 問 題 ( 情 報 ), [https://www.mext.go.jp/content/20211014-mxt\\_daigakuc02-000018441\\_9.pdf](https://www.mext.go.jp/content/20211014-mxt_daigakuc02-000018441_9.pdf) ( 最 終 閲 覧 日 : 2025 年 2 月 )
- [3] 予備校の東進, 共通テスト情報 I | 共通テスト解答速報 2025: web サイト (最終閲覧日:2025 年 2 月)
- [4] 独立行政法人大学入試センター: 共通テスト手順記述標準言語 (DNCL) の説明, web サイト (最終閲覧日:2025 年 2 月)
- [5] 久野靖: プログラミング教育/学習の理念・特質・目標, 情報処理学会誌, 57(4), pp.340-343. (2016)
- [6] 飯田千代ほか: アルゴリズムの思考法の学習-記憶から思考へ-, 研究報告コンピュータと教育(CE), 99(1), pp.1-7. (2009)
- [7] 特定非営利活動法人みんなのコード: プログラミング教育 実態調査報告書 2022 「高校教員の意識調査 単純集計結果」, web サイト (最終閲覧日:2025 年 2 月)
- [8] 鈴木優実ほか: プログラミング初学者に向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, pp.1-7 (2018)
- [9] R. Schank, T. Berman: Time for Content: The Real Role of Technology in Education, Educational Technology, Vol44 No6, pp.5-13 (2004)
- [10] 根本淳子, 鈴木克明: ゴールベースシナリオ (GBS) 理論の適応度チェックリストの開発, 日本教育工学会論文誌, 29(3), pp.309-318 (2005)
- [11] 朴ほか: ゴールベースシナリオ(GBS)理論に基づく情報活用能力育成教育の実践, 日本教育工学会論文誌, 34(SuppL), pp.165-168 (2010)
- [12] 根本 淳子ほか: 問題解決型学習デザインの研究動向--GBS と SCC を中心に, 本教育工学会研究報告集, 10(5), pp.151-158 (2010)
- [13] 文部科学省: 【高等学校情報科「情報 I」教員研修用教材(本編)】第 3 章, web サイト (最終閲覧日: 2025 年 2 月)

## 著者紹介

### 高田 悠矢(学生会員)



2025 年筑波大学情報学群情報科学類卒業、学士(情報工学)。2025 年同大大学院修士課程在学中。現在は GBS 理論を用いた、初学者向けのデバッグに関する学習法の研究に従事。

### 古川 宏(正会員)



1990 年東北大学工学部原子核工学科卒業。1995 年東北大学大学院工学研究科博士課程修了、博士(工学)。1996 年日本原子力研究所博士研究員原子炉安全工学部人的因子研究室。1998 年筑波大学電子情報工学系講師を経て、現在、システム情報系准教授に至る。ヒューマンマシンインタラクション、認知的インタフェース、空間認知とナビゲーション支援、メンタルモデルの獲得に関する研究に従事。2003 年アメリカカソリック大学認知科学研究室客員研究員。2016～2022 特定非営利活動法人モバイル学会理事長および学会長。