

アルゴリズムの理解とコーディングの学習支援 —基本ソートの学習カリキュラム

土屋孝文^{*1}

Email: tsuchiya@sist.chukyo-u.ac.jp

*1: 中京大学工学部

◎Key Words 学習支援, プログラミング

1. はじめに

本研究はC言語入門科目に続く情報系基礎科目「アルゴリズムとデータ構造」を対象に、学習支援環境の開発と運用を行っている。履修者の多くは、プログラミングの学習初期にあたり、カリキュラムを通じて、不完全で不安定な知識を繰り返し再学習しながら、次第にプログラミングスキルを熟達させていく。鈴木(2022)⁽¹⁾は、学習初期に見られる不安定なパフォーマンスを学習者内外の多様な認知リソースの創発的な働きととらえ、その不安定さを問題解決方略構成に必要なパフォーマンスの揺らぎと論じている。このような観点からは、不完全なパフォーマンスに対して、正解に至る手続きと手順を明示的に示す支援のほか、次に続く躓きや誤りの調整や(再)学習に有効な支援の提供が考えられる。たとえば、基本知識の獲得、既有知識の活性化や誘発(想起や利用)、誤概念の修正、問題解決に関するガイド、誤りを含む解の自己調整などを目的とする再学習のための外的支援ツールである。本稿では、基本ソートアルゴリズム(選択ソート)の理解について、再学習に注目したカリキュラムと支援ツールの現状を実践報告する。

授業は大まかに3種のフェーズ、すなわち、対象問題の「アルゴリズムの理解」、「アルゴリズムに対応するコーディング(プログラム生成)」、「プログラムの実行過程とアルゴリズムとの対応(解答例となるプログラムの理解)」の順に進行する。以下では選択ソートの各フェーズに関する(再)学習の支援環境について報告する。

2. アルゴリズムの理解 – 説明生成

アルゴリズムの形式的な説明の前に、操作の具体事例から帰納的な推論による手続きの説明(仮説生成)を促すこととした。具体的な図的情報表現(トランプ)を正解アルゴリズムにしたがって自分で操作する、いわば「アルゴリズムを手作業で正しくなぞってみる」ツールを提供する。図1は選択ソート用ツールの観察部である。

ツールは、まず、典型的な入力([3, 5, 4, 2, 1])の操作事例のほか、バブルソートと誤りやすい入力([5, 1, 2, 3, 4])やユーザ自身の入力について、順に操作事例を表示する。

次に、学習者は与えられた入力に対する操作を試みて、自分が考えた手続きを検証する。手続きを電子掲示板に共有したあと、最後に教員が講義を通して正解を示す。

正解の手続きが確認されたあとに、説明コメントを付加したプログラムと実行結果を例示するレポート作成へ進む。履修者は教科書やネットなど、多くの資料を利用し解答を選択できるが、「なぞれるけれど、自分だけではプ

ログラムにはできない」という振り返りが多い。これは、基本的な既有知識を組み合わせながら、手続きをアルゴリズムとしてとらえ、プログラムを生成していく方略知識が不十分な状況と考えられる。



図1 選択ソートアルゴリズムのなぞり

3. コーディング – 既有知識の準備

プログラム生成の支援に、まず、C言語の構文知識や定型的な処理パターンを再学習するための例題プログラム集を準備し、課題に応じた例題の参照や利用を促す。表1は基本ソートをプログラムするレベルまでに必要と考えられる9つのカテゴリと例題である。例題には、小さくて覚えやすく、多くのプログラムの部品となる操作や定型的処理を含む題材を検討している。

例題の再学習には、例題プログラムを表面的にながめるだけではなく、プログラムの一部を再生してみる(写経)ツールを提供した。図2は、一次元配列の要素を二重ループ処理で操作する例題(▼型出力)で、多くの基本アルゴリズムに繰り返し利用される定型的処理である。ツールは、入力文字への正誤判定を通して、基本構文の組み合わせによる例題構成の確認を促す。このほか、プログラム実行順序(特にループ)など、隠れた誤概念⁽²⁾に気づき修正するためのクイズツールを提供している。

基本ソートは、一次元配列の二重ループ処理と要素交換の例題を適切に変形し組み合わせようとするれば、プログラムの大枠(不完全な部分コード)が得られる。選択ソートでは、「先頭と最小値の要素交換」の繰り返し手続きのコーディングにこの大枠が利用される。

表1 基本例題集

	カテゴリ	例題1	例題2
1	入出力	おうむ返し	配列格納
2	条件分岐	偶数/奇数判定	月 ⇒ 季節
3	ループ	N個合計	N個/終了キーまで合計
4	配列処理	要素合計	連続要素縮約
5	文字列	長さ	1文字検索
6	二重ループ	九九表	一次元配列 ▼型出力
7	その他 (基本処理)	配列要素交換 (swap)	乱数ライブラリ
8	関数	sum 関数	配列要素の合計
9	再帰	階乗	二分探索

問題文

??に入るプログラム文を専程しよう

```

#include <stdio.h>
#define N 5

int main(void) {
    int i, j;
    int a[N];

    for (i = 0; i < N; i++)
        a[i] = i;

    ??

    return 0;
}

```

実行例

```

0 1 2 3 4
1 2 3 4
2 3 4
3 4
4

```

プログラム文

```

for (i = 0; i < N; i++) {
    fo□
}

```

図2 二重ループ例題 再生入力ツール

4. コーディング — 問題解決のガイド

外的な図的情報表現上の手続きをプログラムに対応づけるには、大きく2つの視点が必要と考えられる。1つはデータの表現や制御に使用される変数、定数およびデータ構造の設定である。選択ソートの図的情報表現(トランプ列)は配列で表現され、最小値の決定や配列操作には変数が必要となる。もう1つは、視覚や手指で行っている処理を適切に一連のプログラムに対応づけることである。選択ソートでは、最小値の発見と交換の繰り返しの手続きを二重ループのブロックに対応づける必要がある。

```

int main(void){
    int i,j; // 二重ループに必要な変数
    int k; // 最小値の索引
    int tmp; // 交換バックアップ用の変数
    set_array(); // 配列に乱数をセット
    print_array(); // 配列をプリント

    for(i=0;i<N;i++){
        k=i;
        未ソート範囲の中から最小値を探索(
            暫定最小値より小さい場合(
                最小値を更新
            )
        )
        比較範囲の先頭と最小値を交換する
    }

    print_array(); // 配列をプリント
    return 0;
}

```

クイズ

未ソート範囲の中から最小値を探索(

```

for(j=i+1; j<N; j++){
    if(a[j] < a[k])
        k=j;
}

```

図3 選択ソート 中間表現からプログラムへ

このようなコーディング方略の獲得支援は容易ではない。そこで手続きのなぞりとプログラムとの間に、コンピュータ側の操作にあたる中間的なアルゴリズム表現を提供する支援を検討している。現状の中間表現は、変数の役割の記述と日本語による疑似コードで、プログラムの説明コメント(アルゴリズムとしての読み)に対応する。

ここから、中間表現内の構文要素やブロックを、プログラムへ変換する様子をクイズ形式で学習者に提供する(図3)。クイズ(ガイド)は、既有知識の想起や問題解決1ステップの推論にあたる。最終的に中間表現(アルゴリズムの読み)と解答プログラムを比較表示する。現在は5つの基本アルゴリズムについて運用を行っている。

5. 解答プログラムの理解 — 自己調整

選択ソート課題には、一定数、「先頭と、より小さい値の要素交換の繰り返し」を繰り返す、誤ったプログラムの提出がみられる(図4)。

```

mis-selection-sort() {
    for (i = 1; j =<= n - 1; i = i + 1){
        min ← i;
        for (j = i + 1; j =<= n; j = j + 1){
            if (A[j] < A[min]){
                swap(A[j], A[min]);
            }
        }
    }
}

```

図4 誤りのある疑似コード

課題後の授業では、図1のようにトランプを用いた手続きと、誤答コードに従った操作を対応づけながら、正解と誤答を比較する。解説後に振り返りの質問を行ったところ、「なぞりは正解したが、誤った手続きのコードを書いた」という選択肢は26%で、講義後は誤りの納得や修正を期待できる。一方、提出プログラムに説明コメントを求めるだけでは、誤りの気づきを誘発しにくい事例であり、誤りの発見と修正に関する支援が今後の課題である。

6. おわりに

なぞりからのアルゴリズム理解は、教科書の形式的な解説文や解答プログラムの読解に対するトップダウン情報となる。なぞりと解答プログラムからプログラムの生成方略を推論するための支援を引き続き検討したい。

参考文献

- 鈴木宏昭: "私たちはどう学んでいるのか 創発から見る認知の変化", ちくまプリマー新書403, 筑摩書房 (2022).
- 江川絃美, 三宅芳雄: "詳細な観察に基づくプログラミング学習過程の研究", 中京大学人工知能高等研究所ニュース (IASAI News), 12, pp.10-15 (2003).