

プログラミング再学習の支援

—基本アルゴリズムの学習カリキュラムを通して

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。

この科目は対象問題の「アルゴリズムの理解」の後に「アルゴリズムのコーディング (プログラミング演習)」が続く。学習者の多くは、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の再学習を繰り返しながら、次第にプログラミングスキルを熟達させていくと考えられる。「アルゴリズムのコーディング」をプログラミングの再学習の機会とするためには、実際の学習者に応じた話題 (対象問題) の選択と適切な順序づけが必要となる。

そこで、この科目では教科書¹⁾の目次を C プログラミングの難易度順に並び替えている。表 1 に「一次元配列に対する二重ループ」まで、C の制御構文知識が段階的に複雑となるように配置した話題提供順を示す。標準的な教科書は計算量分析の入門の後、第 2 章でスタック、キュー、リストといった基本データ構造を扱うが、この講義では自己参照型構造体の理解に必要なメモリ操作や再帰の知識を準備できる (再学習できる) 段階へ先送りする。

表 1 目次の再構成 (C 言語構文の複雑さ順)

	基本アルゴリズム	データ表現	制御構文
1	線形探索、二分探索 (キーの探索)	一次元配列 変数操作, 値更新	配列処理ループ
2	素朴法, kmp 法, bm 法 (文字列パターン照合)	配列 2 つ 添え字操作	ループ 条件分け
3	ユークリッド互除法 (最大公約数)	複数変数の値更新	ループ、再帰
4	エラトステネスの篩 (N 以下 n 素数列挙)	一次元配列	二重ループ
5	基本ソート (系列の並び替え)	一次元配列	二重ループ 要素交換 (swap)

以下では、5. の基本ソートのうち選択ソートを例に、「基本スキーマの再学習」「アルゴリズムの説明型理解 (アルゴリズムのなぞり)」「コーディングの方向づけ (ヒント)」に関する (再) 学習の支援環境について報告する。

2. 基本スキーマの再学習

プログラミングの基本的知識には、構文や意味の知識

のほか、プログラミングに固有の知識がある。たとえば、プログラムは逐次的にどこかが実行されていることや、変数の値は実行に応じて変化していくこと、複数の文が適切に組み合わせられ組織的に実行されることによって 1 つの処理が表現されていることなどである。このような重要な基本スキーマ²⁾が不十分な初学者は、基本的知識の組み合わせで理解できるはずのプログラムを正しく読み取ることができない。表面的な手がかりから、何をすればプログラムなのかに注目した「物語読み」にとどまってしまう、変数値の変化や実行順序を追いかける「トレース読み」や、基本構文の組み合わせで構成される定型的処理パターンの読みとりまでは進まないことが多い。

そこで、何を出力するかを表面的に読み取ることのできないプログラム例を作成し、深い読みへの注目から基本スキーマの確認を行う再学習ツールを運用している。

図 1 の例題は「適当回ループして条件を満たすと break する」といった表面的な読みができないため、出力結果の推論には誤答が多く見られる。このツールでは変数値の変化と実行順序を追いかけてながら、誤概念 (ループ脱出時の変数値) や基本的知識 (break の動作) を確認する。

```
#include <stdio.h>
int main(void){
    int a,i;
    a = 15;
    for(i=0;i<9;i++){
        if(a<i){
            break;
        }
        printf("i=%d\n",i);
    }
    return 0;
}
```

図 1 for 文例題 実行順序の確認ツール

もう 1 つの再学習ツールは、コーディングに必要な定型的処理パターンを持つ例題について、読むだけではなく、一部のプログラムを再生する (写経) ツールである。

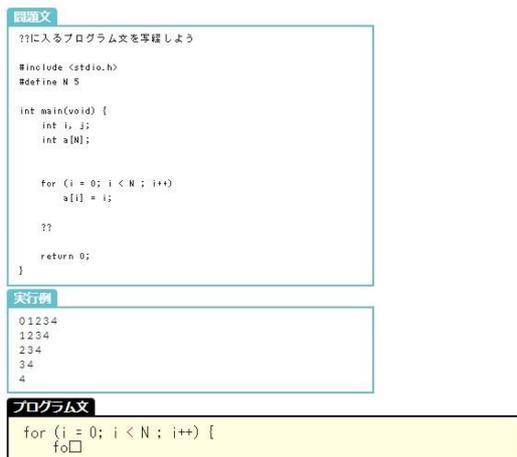


図2 二重ループ例題 再生入力ツール

図2は、一次元配列を二重ループで操作する定型処理にあたる例題である。選択ソートのほか、表1の後半のコーディングで繰り返し利用される知識である。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構成を確認する。

3. アルゴリズムの説明型理解

アルゴリズムの形式的な説明とは別に、具体的な図的情報表現をアルゴリズムにしたがって操作する、いわば「アルゴリズムを正しくなぞる」ツールを提供している。図3は選択ソート用ツールの観察部である。



図3 選択ソートアルゴリズムのなぞり

ツールは、典型的な入力([3, 5, 4, 2, 1])の操作例のほか、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例を表示する。その後、自分自身で操作を行い、検証した手続きの説明を電子掲示板に共有する。最後に教員が講義を通して正解例を示す。何をプログラムするのが明確になったところで、説明コメントを付加したプログラムと実行結果のレポート

が課題になる。多くの学習資料が利用できるはずだが、なぞれるけれど、プログラムにはできないという声も多い。実際には、全くできないわけではなく、個別には理解できている多様な基本的知識を組み合わせる利用しながら、アルゴリズムに対応するプログラムを生成していく方略知識が不十分と考えられる。

4. コーディングの方向づけ

外的な図的情報表現上の操作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(トランプリ列)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、操作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を二重ループのブロックに対応づける必要がある。

このようなコーディング知識の獲得支援は容易ではない。現在は、アルゴリズム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(アルゴリズム読み)に対応する。

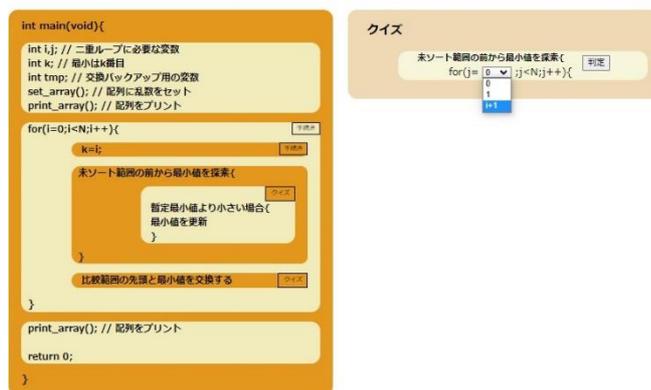


図4 選択ソート 中間表現からプログラムへ

ここから、中間表現内の構文要素やブロックを、クイズ形式を用いて対応するプログラムへ変換する様子を学習者に提供する。クイズ(ヒント)は、関連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. おわりに

再学習を支援するツールの運用を報告した。様々な学習資料(ツール)をメディアに用いて、各自の思考、理解や疑問について、教員や先輩、仲間と協調的に対話しながら学んでいける環境が望ましい。

参考文献

- (1) 広瀬貞樹：あるがごとく、近代科学社(2006)。
- (2) 江川紘美、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)。