

段階的なヒントを利用するプログラミング - 基本アルゴリズムの学習場面を例に -

土屋孝文*1・坂倉毅俊・鹿内拓哉・齋藤真琴
Email: tsuchiya@sist.chukyo-u.ac.jp

*1: 中京大学工学部

◎Key Words アルゴリズム、プログラミング、学習支援

1. はじめに

情報系基礎科目「アルゴリズムとデータ構造」の学習には、主に3種類の領域知識が相互に関係していると考えられる(たとえば教授事例の報告³⁾)。すなわち、対象問題のアルゴリズム(解法)の理解、アルゴリズムに対応するコーディング(プログラムの生成)、プログラムの実行過程とアルゴリズムの対応の理解である。本研究はそれぞれの領域を対象に学習支援環境の設計と運用を検討している。

学習支援環境の設計方針は、以下のとおりである。

- ① 学習者の自己説明や仮説生成を中心におく。
- ② 演習室における同時作業となるので、活動結果の集計/共有/比較や、学習者間コミュニケーションなど協調的な学習環境を活用する。
- ③ 講義では活動結果と理論的知識の橋渡しを行う。

基本的なアルゴリズム例題の理解については、①自分の(素朴な)解法の内省と、学習対象の(賢い)解法の具体的な操作事例に仮説的な説明を行う学習環境の運用、②自己説明や感想の共有環境の運用、③説明レベルや誤答の分析から、学習環境の再設計を行っている⁴⁾。

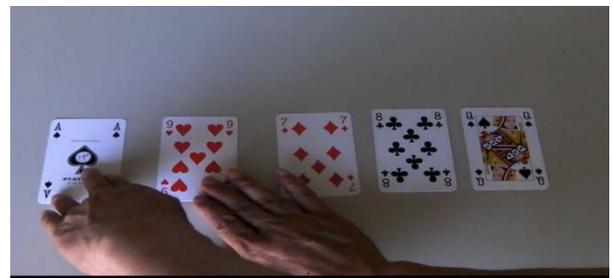
本稿では続くコーディングの支援について、手続き的に重要な意味を持つ部分コード(プログラム中の穴部分)生成に、段階的なヒントを利用可能とした学習環境を報告する。具体的には、穴周辺のコード提供量や穴埋め方法(生成/選択)の切り替えを検討した。

このような環境の運用状況は、学習者の知識や学習方略に依存する。基本ソートに関する環境の具体例と運用結果を報告し、答えを探すだけになりがちな初学者に、関連知識プロンプトとしてのヒント利用や生成プロセスの振り返りを促す方法について議論したい。

以下では、まず基本ソートのアルゴリズム理解環境について報告する。続いて、プログラミングを問題解決ととらえた場合、問題解決に、どんなヒントをどのように提示すればよいかという一般的な設計問題を整理し、試作環境を報告する。

2. 基本ソート法 - アルゴリズムの理解

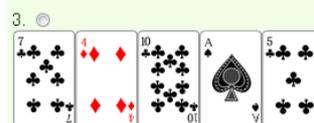
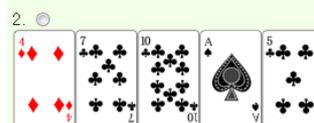
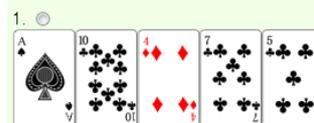
選択ソート、挿入ソート、バブルソートは代表的な基本ソート法である。自分の解法を対象化する活動として、5つの数を並びかえてみるページを作成し、操作履歴を参考にしながら自分の解法を説明、共有することとした。



次は、どうなるだろう



上のカードを、SortAで並び替えています。次は、どの順番のカードになりますか?



確認

図1 観察・仮説生成用動画と仮説テスト用クイズページ

続いて3つの基本ソートに関する動画を観察し、各ソートについて次の操作結果を問うクイズページを作成した(図1)。そのあと、3つのソート法の説明と感想を共有した。2012年度の運用では97名の

参加者は全員がクイズに正解しており、手続きを模倣できるはずだが、各ソートの説明を分析してみると、コーディングに十分で正しい記述にある説明は、選択ソート(54例, 55.7%)、挿入ソート(12例, 12.4%)、バブルソート(31例, 32.0%)となった。これらの結果をみると、一度に多くの自己説明を求めていることが問題と思われる。わかりやすいソートから順に説明や理解を積み上げていく必要があるだろう。そこで2013年度の運用では、選択ソート、バブルソート、挿入ソートの順に説明を行った。ただし手続きをある程度形式的に言語化することや、他人に伝えられるレベルで十分に表現することは、なかなか難しいと考えられる。そこで具体的な表現例やプロンプトといった表現支援ツールの利用を検討している。

3. 問題解決への段階的なヒント

プログラミングを探索型の問題解決としてとらえた先駆的な研究に、初学者のLISPプログラミングに関するAndersonらの認知的モデル⁽¹⁾がある。このモデルは初学者の問題解決事例に、(A) 長期記憶からの関連知識の想起、(B) 例題など外部情報メディアに含まれている知識利用、(C) ヒューリスティクスに基づく問題解決エンジンを用いた記述を行っている。

Hume⁽²⁾らは、循環器生理学問題に関するチュータの教授対話事例を基に、ヒントの機能の定義を試みている。それらを上の問題解決過程と対応させると、ヒントの機能は、(I) 必要な知識を問題解決に利用可能にする効果(主に(A)(B)の支援)と、(II) 行き詰まった問題解決過程を部分的に前進させる効果((C)の支援)ととらえられる。

さらにHumeらはヒント提示法に、関連情報の想起や注意に間接的なプロンプトを与える方略、関連情報の説明や部分解を直接与える方略、問題解決活動に必要な一定の推論を一連の対話によりガイドする方略の3つを抽出している。これらは学習者の問題解決状況に応じて提供されるヒントの質や量を変化させたり方略とみることができる。一般的には、問題解決に必要な情報探索や推論といった情報処理を促したり、その負荷をより軽減したりするヒントほど、より問題解決に貢献するため、より解に近いと考えられる。以上を背景にした段階的なヒント構成の試作について、次に述べる。

4. 基本ソート法 — コーディング

講義によってアルゴリズムを確認した後、手続き的に重要な意味を持つ部分のコードを穴埋めする未完成プログラムを提供した。コード生成のヒントは、穴周辺のプログラム提供量(直接の部分解の量が増え、穴は減少する)と、穴埋め方法の切り替え(生成から選択へ)によって変化させることができる。

穴埋め方法の切り替えは、周辺プログラムの提供による間接的な問題解決推論プロンプトから、より直接的で解に近いヒントへの変化に対応する。

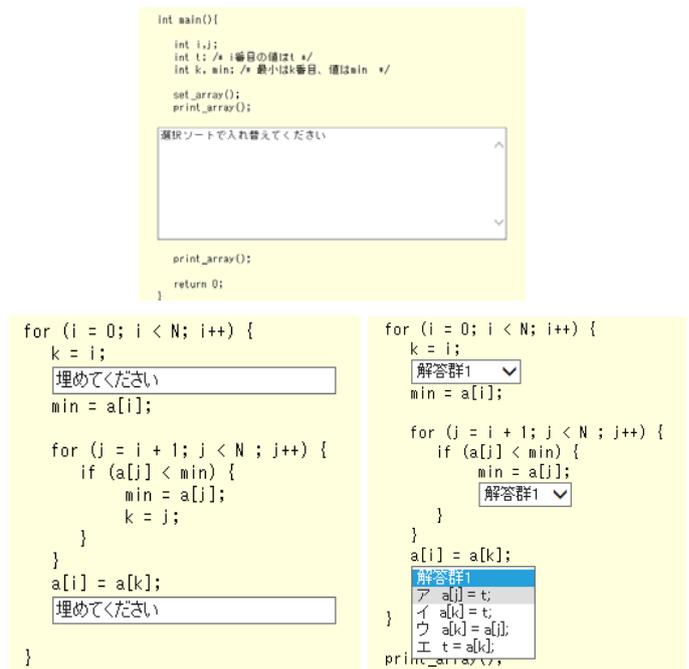


図2 選択ソート: 生成型ヒントと選択型ヒントページ例

図2は、選択ソートの段階的ヒントページの例である。上から右下へと解答に近いヒントとなる。

2013年度の運用について、参加者41名がどのタイプのヒントから回答を送信開始したのかを確認したところ40名が図2右下の選択型からとなった。この結果には、部分的な仮説生成を行わないまま解答に近いヒントへ進んでしまうという問題と、あわせて、解答後に十分な振り返りやまとめが行われているかどうかという問題が含まれている。前者については、手続きをコードに結び付けるため、自然言語による疑似コード表現や、部分コードや類似コードを利用したプロンプト開発などの追加ヒントが考えられる。後者についてはコードの読み直しや、実行過程のトレースなどの活動が考えられる。

参考文献

- (1) Anderson, J. R., Farrell, R. & Sauers, R.: "Learning to program in LISP", *Cognitive Science*, 8, 87-129 (1984).
- (2) Hume, G. D., Michael, J. A., Rovick, A. A., & Evans, M. W.: "Hinting as a tactic in one-on-one tutoring", *The journal of the learning sciences*, 5, pp.23-47 (1996).
- (3) 木村泉: "さなげ山通信 16 アルゴの恐怖 — 歴史, 総合, 心理", *bit*, Vol. 31, No.4 pp.93-101 共立出版 (1999).
- (4) 土屋孝文, 杉山康太, 常富康平, 渡邊裕介: "自己説明に基づく基本アルゴリズムの学習支援環境とその運用", 2013PCカンファレンス論文集, pp. 39-40 (2013).