

アルゴリズム的思考法の学習

— 記憶から思考へ —

飯田 千代[†] 飯田 周作[†] 清藤 武暢^{††} 佐藤 創[†]

アルゴリズム的思考法とは、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である。我々は、このアルゴリズム的思考法を、「情報を学ぶ上での基本的な能力」として認識しており、本格的な情報の学習を始める高等学校における重要なテーマであると考えている。本報告では、学生の学習状況を「躰きパターンによる分類」と「学習パターンによる分類」という 2 つの軸で把握することを提案する。躰きパターンによる分類は学生の理解度を示し、学習パターンによる分類は学生が「暗記型学習」と「思考型学習」のどちらで学習を進めているかを示す。暗記型学習には到達度に限界があり、いかに思考型学習に移行させるかが重要である。

A Pattern of Learning Algorithmic Thinking Style

- Toward learn by thinking rather than learn by memory -

Chiyo Iida[†] Shusaku Iida[†] Takenobu Seito^{††} Hajime Sato[†]

Algorithmic thinking style is a problem solving method, which breaks down a process into basic operations and line up them in a proper order. We consider it as a fundamental ability for students who are studying information and computer science. To teach students algorithmic thinking style, it is important to know each student's learning status. We propose two viewpoints: what kinds of topics do they feel difficult and how they try to lean. For the latter viewpoint, there are two possibilities: "learn by memory" and "learn by thinking". It is important to encourage students to switch from "learn by memory" to "learn by thinking".

1. はじめに

アルゴリズム的思考法とは、作業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である。我々は、このアルゴリズム的思考法を「情報を学ぶ上での基本的な能力」として認識しており、本格的な情報の学習を始める高等学校における重要なテーマであると考えている。

我々は、大学 1 年次生を対象として、「アルゴリズム的思考法」という講義で 2001 年からこのアルゴリズム的思考法の教育を続けている。この講義の目的は、プログラミングの基礎としてアルゴリズムを教えることではなく、情報を学ぶ上での基本的な能力としての思考法を教えることにある。つまり、よくあるアルゴリズムを列挙して、それらの有効性

[†] 専修大学ネットワーク情報学部
School of Network and Information, Senshu Univ.

^{††} 横浜国立大学環境情報研究院
Graduate School of Environment and Information Sciences, Yokohama National Univ.

を説明するのではなく、問題を自ら解決する手段を探る方法を体得してもらうことが目的である。

この講義の中で実施したテストの結果を分析した結果、学生が躓きやすい学習トピックがテスト結果から読み取れることが分かった[1]。[1]では、図1で示すような問題を例として取り上げた(「200X年問題」と命名する)。本稿でも引き続き、200X年問題を例として取り上げる。

問] 以下の仕様を満たすフローチャートを描きなさい。

[仕 様]

正の整数 n を入力として受けつけ、 x_0 から x_n までの各値が、

2, 0, 0, 7, 2, 0, 0, 7, 2, 0, ...

と 2, 0, 0, 7 がくり返し並ぶように配列 x を設定し、出力する。

図 1

さらに[1]では、この問題に対する解答をタイプ別に分類した。

1. 正解
2. 単純ミス
3. 繰り返しの理解不足
4. 配列の理解不足
5. 問題の読解ミス
6. 混乱
7. 白紙

これは言わば学生の表面的な分類である。つまり、その学生が書いた解答のみをもとにミスのタイプで分類している。本稿では、この分類を「躓きパターンによる分類」と呼ぶことにする。

1.1 学習パターンによる分類

先の報告を行った後、講義における学生の反応や、解答のタイプについて改めて検討を行うと、不正解の解答の中に、以下のような解答例が存在することが判明した。

1. 講義内で取り扱ったアルゴリズムの形をそのまま暗記し、思考ではなくパターンマッチによって問題を解こうとしているタイプ(暗記型学習)。
2. 正解には至らなかったものの、アルゴリズム的思考法を用いて問題を解こうとしているタイプ(思考型学習)。

暗記型学習タイプの特徴は、回答のフローチャートに問題とは全く関連がない処理が現れることである。一方、思考型学習タイプは、回答から学生の意図が読み取れる。暗記型と思考型学習タイプでは問題に対処する方法が全く異なり、この差異がアルゴリズム的思考法の学習に大きな影響を与えているのではないかと考えた。

本報告では、暗記型学習パターンと思考型学習パターンの学生がどのような特徴を持っているのかを分析し、さらに[1]で行った分析との関連を調べる。それらの分類を踏まえて、アルゴリズム的思考法の教授法についての考察を行う。

1.2 本報告の構成

2章では[1]の躓きパターンによる分類をさらに進化させ、3章では記憶型学習と思考型学習の違いを見る。4章では、具体的な例に対してこれらの分類を適応し分析を行う。5章で、これらの結果を考察する。

2. 分類

[1]では、200X年問題に対する解答から「躓きパターンによる分類」として学生を7つのタイプに分類した。今回はそれらの分類と、我々が講義中に直面した事例や、学生に個別に行ったインタビューから得た実感などをもとに、学生の躓きの根本的な原因に目を向け、それをもとにより本質的と思われる分類を提案する。

- a. 経験不足
- b. 繰り返し処理の理解不足
- c. 変数の理解不足
- d. (ルールに対して) 無神経

タイプ a の経験不足というグループは、前回の分類で 1 の正解および 2 の単純ミスに属した学生と、それ以外にタイプ 6 の混乱のグループに属した学生のほんの一部からなる。このグループは、実はあま

り問題がないと考えている。経験不足と名付けた理由は、単に彼らはビギナーであり、このまま経験をつめば特に躓くことなくスキルアップしていくであろうグループである。(前分類のタイプ1の学生は、200X年問題に関しては正解であったが、他の問題全てで満点を取ったわけではない。)混乱した解答を書く学生の中に、なぜこのグループに含まれる者がいるのか、という疑問があるだろう。前分類はタイプ6の混乱とされながら、新分類でこのグループに分類された学生の解答は、一見混乱したようなフローチャートでも、注意深く見ると何らかしらの方向性をもって記述され、試行錯誤の結果收拾がつかなくなった、というようなものであった。收拾をつけられずにギブアップした理由には、制限時間の問題が多分にあると思われる。

このような解答を書く学生は、自分のアイデアをチャートに表し、またそれを検算し、修正を加えるという作業が行えるという意味で、なぜ自分がそのような解答を書いたか全く説明ができない真に混乱した学生とは分けて考えるべきである。

タイプbの繰り返し処理の理解不足というグループは、前回の分類のタイプ3の繰り返しの理解不足と、タイプ4の配列の理解不足と分類された学生の一部から構成される。このグループの特徴は、メインの処理に関しては問題を解決するためのアイデアを自ら考案し、そのことが読み取れる程度に整合性のある処理を部分的には書くことができる。しかし、ループをうまく構成できない。一目で無限ループするとわかるような処理を書いたり、ループカウンターとして使用するはずの変数を下手にいじってしまい、ループの処理が意図する回数行われなかったりする。

このグループに属する学生は、検算能力が弱いといえるのではないだろうか。講義中には表を用いた検算のやり方を説明している。表を用いて自らが書いたフローチャートを検算すれば、どこで意図しない処理が行われるのか、一目瞭然のはずである。タイプaとタイプbの大きな学生の違いはこの検算能力であるといってもよい。与えられた問題を解決す

るための良いフローチャートを得るためには、その過程で試行錯誤する必要がある。試行錯誤に必要な能力は、まず何かしらアイデアを思いつきそれを記述できることと、そしてそのアイデアを自ら検証し、問題があれば修正できることである。タイプbの学生も、前半部分のアイデアの記述はできているといえる。しかし、後半部分の検証ができないため、とりあえず頭に浮かんだアイデアを記述したというところで作業が止まってしまう。(このタイプの学生の解答用紙は比較的きれいである。一方、タイプaの学生で、特に前回の分類で混乱とされていた学生の解答用紙は、フローチャートを何回も書き直した後や、検算のためのメモ書きが多数見られる。)

タイプcの変数の理解不足というグループは、アルゴリズムを学ぶ上でかなりの努力を要すると思われる。おそらく中学校で習う文字式、方程式でつまずいて、数式中に文字が現れる様な形状のものに抵抗を感じている。いわゆる数学アレルギーをもつ学生はこのグループに属することが多いだろう。彼らは全般に数学的な知識に興味がないため、この手のことは今ある知識だけでなんとか切り抜けようとする。その結果、文字式の文字と変数の違いが理解できないため、変数の保持する値が変化するということが信じられない。また配列に至っては、高校で習う数列を連想させ、さらに添え字も変数となった x_i 等の表現は、彼らのアレルギー反応を助長する。

前分類のうち、タイプ4やタイプ6の一部、タイプ7は新分類ではこのグループに属する。

タイプdに属する学生はルールを守ることに関心が薄いといえる。このグループに属する学生の書いた解答でまず目につくのは、シンタクスのミスが非常に多いということである。代入演算子である「←」と等価演算子である「=」をしょっちゅう混同する。また、入出力では平行四辺形、分岐ではひし形、代入では長方形と決めている処理の書き方のルールも、入出力や分岐を長方形で書いたりする。それもひとつのフローチャート内にひし形の分岐と

長方形の分岐が混在したりしている。意味的な部分で多いミスは、フローの合流地点の誤りである。特に多いのがループ処理の合流地点であるが、例えばループカウンターの初期化処理をループに含めてしまったり、条件分岐で分岐した片方はカウンターのインクリメント処理を飛ばしてしまったりする。「ケアレスミス、うっかりミス」といって片づけてしまえばそれまでだが、通常は誰しもが十分に気がつかってミスがないように努めるはずの期末考査等でも随所にそのようなミスがみられるのである。これは、たまたまその時に注意がおよばなかっただけではないだろう。

日常生活においては、そのような「些細な違い」に気を配らなくてもうまくいくことが多い。むしろ、そのようなことは気にとめない方がスムーズなことのほうが多いのも事実である。例えば、日常的に良く目にするトイレの男女識別サインは無数の種類があるが、その形（丸い、角張っている等）や色（赤、青等）を見てどちらが女性用でどちらが男性用であるか、ほとんどの場合は区別可能である。このようなものに「厳密さ」を求めることはナンセンスである。

しかし、「アルゴリズム的思考法」の講義の中で最も学生に伝えたいことのひとつとして、厳密な議論をするためには、その構文や意味が厳密に定義された言語を使用する必要がある、ということがある。自然言語には曖昧な部分があり、そのような用途には向かない。フローチャートは図式言語であるため、可読性が高く、初心者が扱うのにちょうどよい形式言語であるといえるが、その可読性の高さ（見た目の柔軟さ）が逆に徒となって、自分が記述しているものの意味を十分に理解しないままに、「それっぽいチャート」を書いてしまう学生がいる。その構文や意味が曖昧なままフローチャートを用いては本末転倒である。フローチャートには方言が多いため、確かに代入演算子として「=」を用いているものもある。しかし、我々のフローチャートでは「←」が代入演算子であると決めた。「=」は等価演算子として用いている。「 $x \leftarrow x+1$ 」のつもりで「 $x = x+1$ 」と書いたら、(気持ち的には理解できるが)それは構文エラー

である。また、ループを回る毎にカウンターを初期化しては、それは無限ループとなる。残念なことに、このことの重要性を理解しない学生は少なくない。彼らはこれを「ちょっとしたミス」と捉える。

本論文では、[1]で提案した「躓きによる分類」を、全ての学生がきっちりこの4タイプに分類できるというわけではなく、どれにもあてはまらない学生（そもそもやる気が全くない等）、もしくは複合タイプであるような学生も存在する。

2.1 記憶型学習と思考型学習

前節では、躓きの原因は何かということに焦点を当てて学生の分類を行ったが、1.1で述べたとおり、「学習の仕方」という視点で学生を見た時に、明らかに2タイプの学生がいることがわかった。あるグループの学生は、講義中に説明したフローチャートを途中までそのまま書き、そのまま書き直しをすることで問題を解こうとしていた。しかし、問題に対する思考が十分でないために、このグループの学生達はほとんど正解していない。また、個別のインタビュー等で多くの学生と一対一で向きあい、問題を解く過程を観察することにより、その事実はさらに明らかとなった。記憶に頼る記憶型学習は、学習初期の単純な問題に対してはある程度有効である。しかし、今回取り上げた例のような難しくはないのだが、これまでに扱ったことのないような問題に直面するとそれ以上理解が進まなくなってしまう。4章では、その具体的な例を見ていくことにする。

3. 事例

ここではいくつかの事例を紹介する。

3.1 試行錯誤

図2は一見ひどく混乱しており、とても正しいフローチャートであるとは思えない。ところがよくよく流れを辿ってみると、これは正しく動作する。お

そらくこのフローチャートを書いた学生は、最初におおよその処理の流れを思いつきそれをフローチャートに表したが、自らテストしてみるとうまくいかないことがわかり、少しずつ修正を重ねてなんとか期待通りの動作をするフローチャートを得たのであろう。この全く洗練されていないフローチャートに対して高評価を与えることはできないが、このような解答を書く学生は、実はあまり問題がない学生といえる。つまり、まだ未熟ではあるが向いている方向は間違っていない。

洗練されたシンプルなフローチャートを書くためには、それなりの訓練を必要とする。また、初めて向き合う問題に対しては、試行錯誤して自分なりに良いと思えるフローチャートを導くのが普通である。この学生は、制限時間の問題もあって今回は試行錯誤の結果のなんとかつじつまをあわせた解答を書かざるを得なかったのかもしれない。しかしこの「なんとかつじつまをあわせる」能力は、アルゴリズム的思考において重要である。この学生の場合は、十分な時間が与えられれば、より洗練された解答を自ら導き出す可能性は十分にあるといえる。

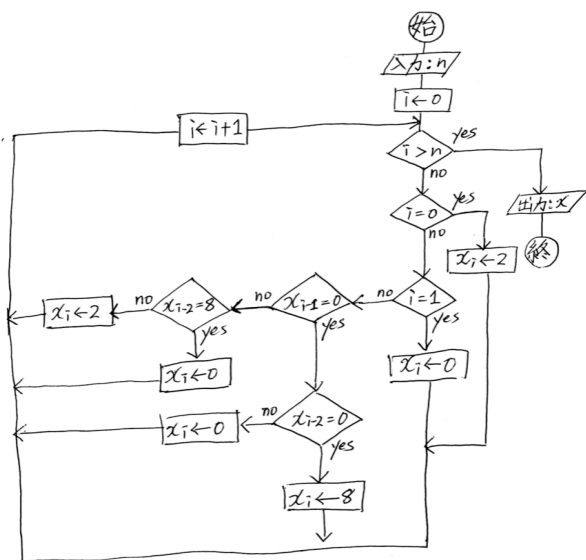


図 2

3.2 記憶のツギハギ

問題文をみて、それに含まれるキーワードから過去に解いた（模範解答を得た）問題を思い出し、そ

れらの組み合わせで問題に対処しようとしている。一見この行動は真つ当で何も問題ないように思える。しかし、我々は図_のようなフローチャートは大変問題であると考えている。

まず、このフローチャートは二つのループをもっている。最初のループは、ループの継続条件が中途半端な状態で書かれているが、もしここが「 $i \leq n$ 」のような構文的に完全な状態であったとしても、この第一のループはループカウンター i をインクリメントするのみで単にスピンする。このような全く意味のない処理は、思考型の学習をしていれば絶対に書き得ない。曖昧に記憶に頼ってフローチャートをなんとか構成しようとする学生の典型的な解答パターンであるといえる。講義中には、最初に配列を初期化するような問題をいくつか扱った。それらに対するフローチャートは、基本的に図_のように二つのループから構成されており、最初のループは配列の初期化で、もう一方のループでメインの処理を行うようになっている。この解答を書いた学生は、おそらくそれらのフローチャートを丸暗記し、問題文中の「配列」という言葉に条件反射のように反応して、とにかくこれらの二つのループを書いたのであろう。それも、チャートの構成要素であるひとつひとつの処理の意味は全く考えずに、その形のみを暗記しているため、曖昧な記憶からは非常に中途半端なフローチャートを再構成することしかできない。

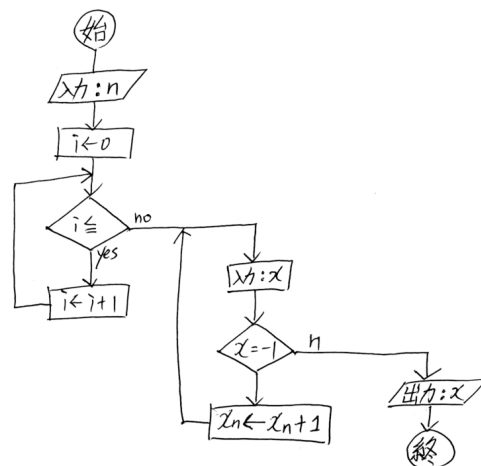


図 3

3.3 再利用!?

前分類のタイプ5には「読解ミス」と名付けていた。このグループに属する学生は、解答を見ただけでは、問題の意図を誤解したとしか思えないフローチャートを書いている。正解とはほど遠いにもかかわらず、似たような解答を書いている学生が多数存在したため、これらの学生達は何か問題を勘違いしたのではないかと考えたが、その原因は同じ問題用紙に載っていた別の問題にあることがわかった。その問題を図4に示す。

<p>問】空欄を埋めて、以下の仕様を満たすフローチャートを完成させなさい。</p> <p>[仕様]</p> <p>正の整数 n を入力として受けつけ、次の条件を満たす配列 x を設定し、出力する。</p> <p style="text-align: center;">$x_0=n, x_1=n-1, x_2=n-2, \dots, x_n=0$</p>

図4

この問題には、2カ所が虫食いとなったフローチャートが予め与えられており、学生はその空欄を適切な式で埋めるだけとなっている。

図1と図4の問題を良く見比べてみてほしい。この二つの問題は字面だけをみるとかなり似ているのである。もちろん、正解となるフローチャートはこの二つの問題では全く似ていない。しかし、前分類で読解ミスと分類した学生達は、おそらく問題の字面上の類似に気づき、それならばその解答となるフローチャートも似ているはずだという判断の下、図4の間で与えられた虫食いフローチャートを基に200X年問題に挑んでいる。そのため、フローチャート自体の形はすっきりとシンプルにまとまっているが、それは全く別の仕事をするものになっている。そしてこのタイプの学生の解答は、当然だがみな酷似している。

このような行動をとる学生は、完全に記憶型学習のグループに含まれる。

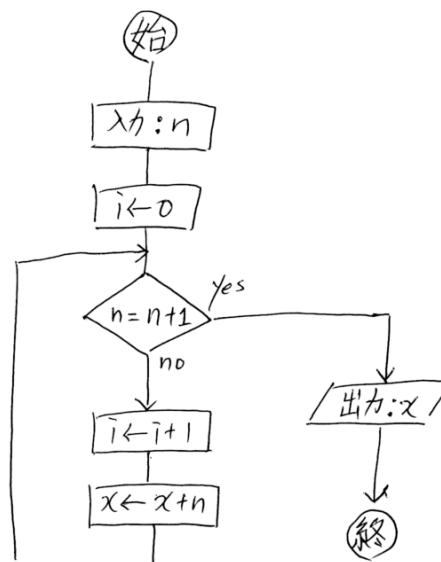


図5

4. 考察

本章では、4章で示した学生の回答を元に、アルゴリズム的思考法の指導を行う際の指針について考察を行う。我々は[1]において、学生がどのような学習上の問題を抱えているかについて、回答のパターンから7つの分類を行った(躓きのパターンによる分類)。この分類方法は、テストにおける学生の回答パターンによる分類であった。今回は、この躓きのパターンによる分類を発展させて、学生がどのような概念で躓いているのか測定できるようにした。しかし、この躓きのパターンによる分類だけでは、どのような学習方法を指示することによって改善され得るのかについては十分な情報を与えない。そのため、新たに学生の学習パターンが「記憶型学習」と「思考型学習」に分かれることに着目した新しい分類方法を導入した。この分類は、学生がどのような意識を持って学習に望んでいるのかを判別する能力がある。この躓きのパターンによる分類と学習パターンによる分類という2つの判別方法を用いることによって、学生がどのような理由によりどこで躓いているのかをより正確に把握することが可能となる。アルゴリズム的思考法の学習には、記憶と思考の

双方が必要であることは言うまでもない。フローチャートの基本的な記述法を記憶することから始まり、例題をまねながら徐々に自ら思考して新たな問題に対処する能力を得ていく。しかし今回の分析により、記憶に頼りすぎてなかなか思考の段階へと移れない学生が多く見受けられることが分かった。記憶に頼る学習方法は、学習の初期段階では効果を挙げるものの、徐々に対応できない問題が増えてくる。今回、このような結果が分かったのは、我々の要求した課題が「典型的なアルゴリズム」ではなかったからだと考えている。典型的なアルゴリズムの問題とは、並べ替えや探索と言った類のアルゴリズムで、多くの教科書に正解が示されている問題を指す。それに対して、我々が用いている「200X年問題」は、唯一の正解というものが無い問題である。様々な回答が予想され、許容される。もちろん効率の良いやり方や、より簡素な表現という尺度を使っての優劣は論じることができるが、そういうゴールを目指すことは我々の目的ではない。学生は、そのような問題に直面したとき初めて、自分なりの考え方を形にしなければならなくなる。そしてその時、記憶型学習は限界を迎える。

指導者にとっての問題はここからスタートする。

「問題を与えて解答を1つ示す」という教授方法では、記憶型の学生を思考型に転換させることは難しい。思考型の学習方法へ転換させるには、まず様々な解が存在し得る事を示し、例えば学生の回答が正答でなくても、彼らの考え方の延長線上に答えがあるという実感を持たせる必要がある。これは、言い換えるならば「思考の体験」と呼べよう。学生に対しては、正解を暗記することが重要なのではなく、どのように考えればよいのかを体得することが重要であることを繰り返し説明する必要がある。

これらの議論を踏まえて、我々は以下の3点を提案する。

- 学生の学習段階を判断するために7分類は有効だが、学生の学習行動を理解するには今回提案した記憶型学習・思考型学習の区別が重要である。これらの分類を適切に使い分けることにより、学生の理解の度合いと質を把握することが可能と

なる。

- 思考型学習を行わせるためには、正答が1つに定まるような問題よりも、様々な思考が可能であるような問題を使うべきである。
- マス教育を行う中で思考法を教授するためには、個人個人の考え方を尊重する工夫をしなければならぬ。そのためにはできるだけ多くの解答を示し、時間をかけて思考の体験をさせる必要がある。

5. おわりに

今回の研究では、学生の学習パターンによる分類、すなわち、記憶型学習と暗記型学習の区別を導入した。今後は、[1]で提案した躓きパターンによる分類と、今回提案した学習パターンによる分類の2つの軸を使ったより具体的な学生への指導方法と学習教材の開発を行う予定である。我々は特に、「思考の体験」を与えられるような教材が必要であると考えている。これは、[1]で提案した「体験による学習」を補強するものである。

謝辞

本研究は、平成20年度文部科学省科学研究費補助金(課題番号19300285)「高等学校情報化における科学的ミニマムエッセンシャルズのための教育プログラムの開発」の助成を受けています。

参考文献

- [1] 飯田周作, 飯田千代, 清藤武暢, 佐藤創, 「アルゴリズム的思考法の教育」, 情報処理学会研究報告2008-CE-93, pp.57-64, 2008.