

筆算の認識を深めるプログラミングの在り方

算数科の検定教科書の分析を中心として

How Programming Should be Used for Deepen One's Knowledge of calculation by writing

星野 将直 (Masanao HOSHINO)

1. 筆算とプログラミング的思考

小学校プログラミング教育の手引¹⁾(第三版)(令和2年2月文部科学省)によると、「プログラミング的思考」とは、「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」であると説明されている。

そこで、手引きに例示されている「正三角形をプログラミングする授業」で「プログラミング思考」を具体化すると次のようになる。

Scratch アプリ上で、猫が移動して正三角形を作図させることを実現するために、辺(等辺)ー内角(60°)ー辺(等辺)ー内角(60°)ー辺(等辺)ー内角(60°)というように要素の組み合わせで、正三角形が構成されていることを把握する。そのために、辺には「○歩進む」という記号、角には「○° 回転(外角)する」という記号を図1の左側のように割りあてると、猫が正三角形の周上を動くプログラムになる。ここで、スタートボタンをクリックすると猫が移動し正三角形を作図する。

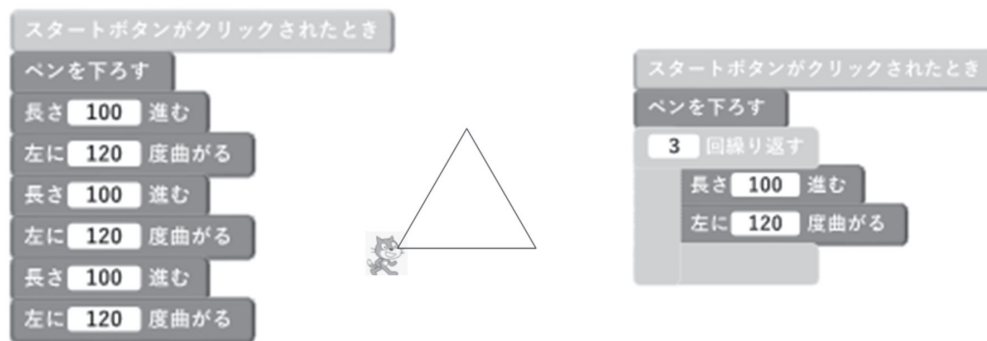


図1 Scratch アプリ上で正三角形を作図するプログラム 図は Scratch(MIT メディアラボ)による

さらに、図1の左側のプログラムの繰り返し部分に着目すると、同じ記号が3回繰り返されている。そのため「反復処理」を用いることができ、図1の右側のようにプログラムを改善できる。このようにして、猫が移動し正三角形の作図をする効率的なプログラムができる。

この過程は、現実の世界での一連の活動(問題)について、記号化をすることから始まる。この記号化された世界のルールに基づいて問題を解決し、その解を現実の世界にあてはめ、現実の問題についてさらに改善を図ることである。

また、この過程は中教審答申*で示された「事象を数理的に捉え、数学の問題を見いだし、問題を自立的、協働的に解決し、解決過程を振り返って概念を形成したり体系化したりする過程」である「数学的活動」とほぼ同義であると捉えることができる。(図2参照)

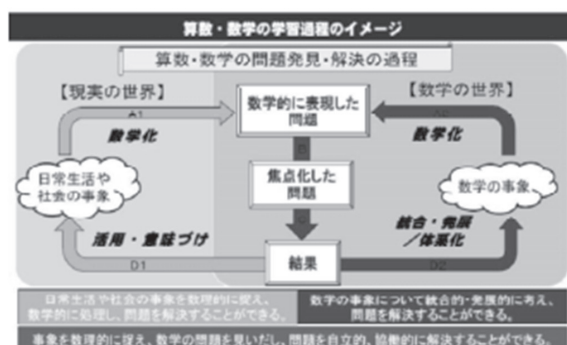


図2 算数・数学の学習過程のイメージ

※平成28年8月26日算数・数学ワーキンググループの審議のとりまとめより

ただし、記号化の対象が「プログラム言語」であったり「フローチャート」であったりすることが通常行われる数学的活動の数理化と異なる点である。

前論文²⁾においては、「プログラミング的思考」を算数科の学習に導入することによって、従来の立場より新たな教材価値が生じたり認識を深めたりすることができるかについて検討した。結果、第5年単元「正多角形」において、「ベクトルの見方」「一筆書きができるグラフ」という正多角形の新しい図形の見方ができること、さらには外角の見方による論理的思考をも高めることができることを示した。そのため、第5年単元「正多角形」において、従来までの教材価値をふまえた学習に加え、「プログラミング的思考」を扱うことは有効であると結論した。

さて、本論文では、整数の筆算の学習でフローチャートを用いた授業について検討する。このときに用いられるフローチャートは、コンピュータを用いなくてもよいので、いわゆるアンプラグドプログラミングと呼ばれている。小学校にプログラミング教育が導入されるようになってから、アンプラグドプログラミングであるフローチャートを用いた筆算の実践が多く行われている。このことはWeb検索の数や実践が掲載されている書籍の発行数からもわかる。

従来から算数教育では筆算の学習でアルゴリズムを意識した授業が行われていた。これが最近になってからは、フローチャートを用いることによってプログラミング的思考を促す授業になっている。おそらく、次に示す「小学校段階におけるプログラミング教育の在

り方について（議論の取りまとめ）」³⁾がこの流れの契機の1つとなったと考える。

私たちが計算するときには、プログラミングで表現しなくても、人間の文明が生み出した遺産である「筆算」で計算することができる。小学校で筆算を学習するということは、計算の手続を一つ一つのステップに分解し、記憶し反復し、それぞれの過程を確実にこなしていくということであり、これは、プログラミングの一つ一つの要素に対応する。つまり、筆算の学習は、プログラミング的思考の素地（そじ）を体験していることであり、プログラミングを用いずに計算を行うことが、プログラミング的思考につながっていく。

また、このことと関連するが小学校学習指導要領(平成 29 年改訂文部科学省)⁴⁾および算数科解説編⁵⁾には筆算の学習でプログラミング的思考を促すことは示されていないため、算数科では「小学校プログラミング教育の手引」にある「※B 領域 学習指導要領に例示されていないが、学習指導要領に示される各教科等の内容を指導する中で実施するもの」として扱うことになる。

そこで、筆算の学習でフローチャートを用いることの教材価値についての検討をはじめめるわけであるが、前論文²⁾と同様の進め方で議論する。それは筆算のアルゴリズムについて記号化した表現であるフローチャートを用いることによって、筆算の認識をどのように深めることができるのかを議論することである。この結果として、小学校学習指導要領の目標である教科の学習での生きて働く知識・技能の習得に主眼をおいたとき、算数科でプログラミング思考を導入する意義についての示唆になるからである。

2. プログラミングによって筆算の認識をどのように深めるのか

ここでは、筆算の認識を深めるため(目的)のフローチャート(手段)というように捉え、次のように A の立場と比較して B の立場ではどのような教材価値が生じるか検討する。

A：従来の(フローチャートを用いずに)アルゴリズムを意識させた場合の筆算の学習

B：フローチャートを用いてアルゴリズムを意識させた場合の筆算の学習

(1) 算数の検定教科書 6 社の筆算の分析

最初に、算数の検定教科書 6 社⁶⁾⁷⁾⁸⁾⁹⁾¹⁰⁾¹¹⁾では、プログラミング教育が導入された現行教科書(令和 2 年度発行)と、プログラミング教育が導入されていない旧教科書(平成 26 年度発行)での、筆算でのプログラム教育やプログラミング的思考の扱われ方について、4 学年単元「わり算の筆算」をもとに調べてみる。なお、除法筆算のみを取り上げたのは、加法・減法・乗法と比較して、除法は「たてる→かける→おろす→ひく」を繰り返すというアルゴリズムが明確だからである。違いを比較する観点は、単元の導入、除法筆算でのアルゴリズムの提示、プログラミング教育との関連(フローチャートの提示等)、その他(現行と主な旧版の違い)などである。

表1 第4学年単元「1桁でわるわり算の筆算」の単元構成とプログラミングとの関連

教科書会社	単元導入	除法筆算でのアルゴリズム提示	プログラミング教育との関連	単元構成など	その他
啓林館	(旧)買い物場面 $60 \div 3$, $120 \div 3$, $600 \div 3$ (現)色紙分配 $72 \div 3$	$72 \div 3$ でアルゴリズム→ $53 \div 4$ (旧)・ $93 \div 4$ (現)でアルゴリズム	プログラミング教育関連は教科書に「P」マーク(現)	現行と旧とも単元構成に変化なし。	現行では「たてる・かける・おろす・ひく」が2回記述されている。※アルゴリズム中に「十の位を計算」・「一の位の計算」という表記がない(旧・新)
教育出版	(旧)色紙分配 $60 \div 3$, $63 \div 3$ (現)色紙分配 $63 \div 3$	$72 \div 3$ でアルゴリズム(旧・現)	特になし(旧・現同じ)	現行と旧とも単元構成に変化なし。	現行ではわり算の見積もりが導入時に掲載されている。
大日本図書	(旧)色紙分配 $63 \div 3$ (現)色紙分配 $60 \div 3$, $63 \div 3$, $72 \div 3$	$72 \div 3$ でアルゴリズム(旧・現)	巻末の特設ページでプログラミング関連がありフローチャートが示されている(現)。	現行と旧とも単元構成に変化なし。	旧と比べて現行は単元導入 $72 \div 3$ の問題の解決に時間をかけている。
学校図書	(旧)キャラメル分配 $48 \div 9$ (現)色紙分配 $27 \div 3$, $36 \div 3$, $48 \div 3$	$48 \div 9$ でアルゴリズム→ $72 \div 3$ でアルゴリズム→ $74 \div 3$ (旧)・ $83 \div 5$ (現)でアルゴリズム	特になし(旧・現同じ) プログラミング教育関連というよりアルゴリズムの認識過程を重視している。	現行と旧とも単元構成に変化なし。	余りのあるわり算でのアルゴリズムが提示され、余りのないわり算でもアルゴリズムが提示され、再度余りのあるわり算でもアルゴリズムが提示されている(旧・現)。
東京書籍	(旧)色紙分配 $32 \div 4$ や $24 \div 4$ (現)数のあてはめ $24 \div 4$	$72 \div 3$ でアルゴリズム(旧・現)	特になし(旧・現同じ)	現行と旧とも単元構成に変化なし。	旧も現も「たてる・かける・おろす・ひく」を2回記述されている。
日本文教出版	(旧・現)日常場面から取り出す $36 \div 9$, $13 \div 4$, $42 \div 2$, $60 \div 3$	$42 \div 3$ (旧), $72 \div 3$ (現)でアルゴリズム	特になし(旧・現同じ)	現行と旧とも単元構成に変化なし。	旧も現も「たてる・かける・おろす・ひく」を2回記述されている。

表 1 より 3 点示す。1 つは、除法筆算でのアルゴリズムについて、プログラミング教育関連として扱っていることがわかるのは、啓林館と大日本図書の 2 社である。啓林館は旧版より紙面を増やし、「たてる・かける・おろす・ひく」を 2 回記述し除法アルゴリズムについて説明し、「P」というマークを示しプログラミング教育との関連を示している。大日本図書は、現行で巻末にプログラミングに関する特設ページがあり除法アルゴリズムとフローチャートを扱っている。ただし、フローチャートについては、除法筆算のアルゴリズムではなく 3 桁の加法筆算のアルゴリズムの表現である。

2 つは、アルゴリズムの提示回数について、学校図書は、旧版と現行とも導入時に余りがあるわり算でアルゴリズムを示し、次の余りのないわり算と余りのあるわり算のともアルゴリズムを示している。つまり、アルゴリズムを 3 回示している。啓林館は、最初は余りのないわり算でアルゴリズムを示し、次に余りのあるわり算でアルゴリズムを示しているので、アルゴリズムを 2 回示している。また、「十の位を計算」・「一の位の計算」という表記はない。他社は、余りのないわり算でアルゴリズム 1 回のみ示している。

3 つは、各社とも単元構成に大きな変更(単元構成, 教材変更・配列)はない。変更がほとんどないということは、プログラミング教育の導入の有無にかかわらず、除法筆算とアルゴリズムの認識について解釈がほとんど変わっていないということになる。ただし、6 社とも導入時に $72 \div 3$ での分け方について教科書の紙面を多く使い問題解決的な活動を行っている。これは除法筆算アルゴリズムが単なる暗記の対象とならないように、意味理解を図るための工夫だと考えられる。この点からは、現行では余りのないわり算でのアルゴリズムの学習が重要視されるようになったと言える。

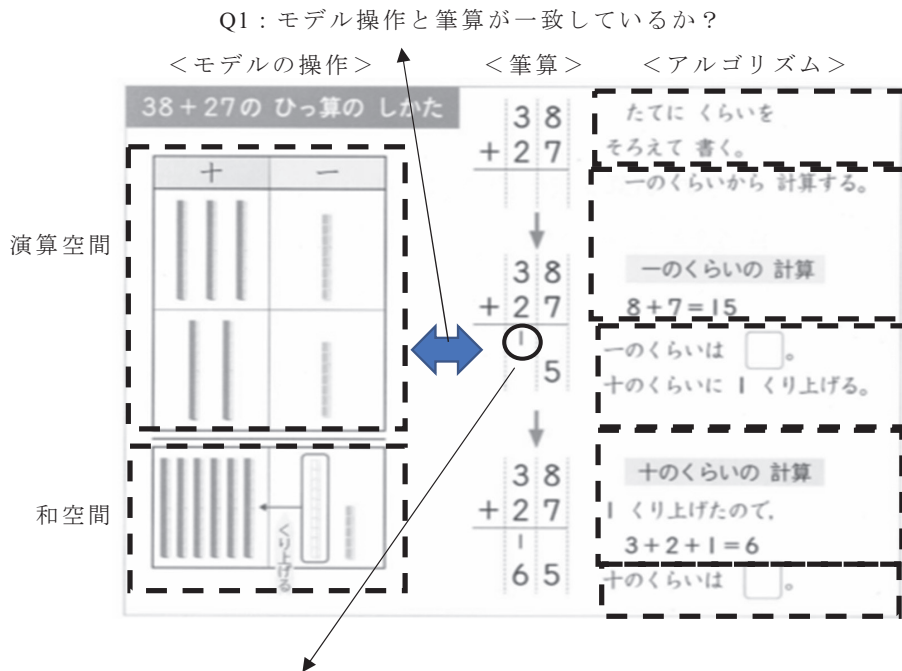
次に観点を变えて、現行の 6 社の加法・減法・乗法の筆算ページの内容について調べてみる。いずれも、計算手順が多くなる繰り上がり・繰り下がりのある筆算を対象とした。筆算の認識のためのモデル(ブロック, 数え棒, お金)の操作と筆算の式操作の対応という枠組みをもとにする。図 3 のように $38 + 27$ という演算を計算する空間(演算空間)と、65 という答を入れる和(差)空間の 2 つに分けた上で Q1・Q2・Q3 の 3 点を調べる。

Q1: 演算空間での操作, 和(差)空間での操作と区別したとき。モデル操作と筆算の式操作は対応しているか?(どちらの空間で操作しているか明確でない場合もある。)

Q2: 繰り上がりの数, 繰り下がりの数が, 筆算の演算空間に記述されているか, 和(差)空間に記述されているか?

加えて、筆算の手順であるアルゴリズムの文言について調べてみる。加法・減法, 乗法の筆算の導入時の繰り上がり・下がりのない場合では、図 4 のようにアルゴリズムの文言の 1 番目に「位を縦にそろえて書く」という文言が全 6 社にある。そこで、加法・減法, 乗法の筆算で繰り上がり・下がりがある場合には、その 1 番目の「位を縦にそろえて書く」というアルゴリズムの文言があるかどうかを調べた。

Q3: 繰り上がり・下がりのある筆算で「位を縦にそろえて書く」とあるか?



Q2: 繰り上がりの数が演算空間にあるか? 和空間にあるか?

図3 筆算の意味理解の分析の観点(1) 図は一松信ほか(2020)『小学校算数2上』学校図書,p39より

<繰り上がり・下がりなし>

<繰り上がり・下がりあり>

Q3: 繰り上がり・下がりのある場合で

「位を縦にそろえる」とあるか?

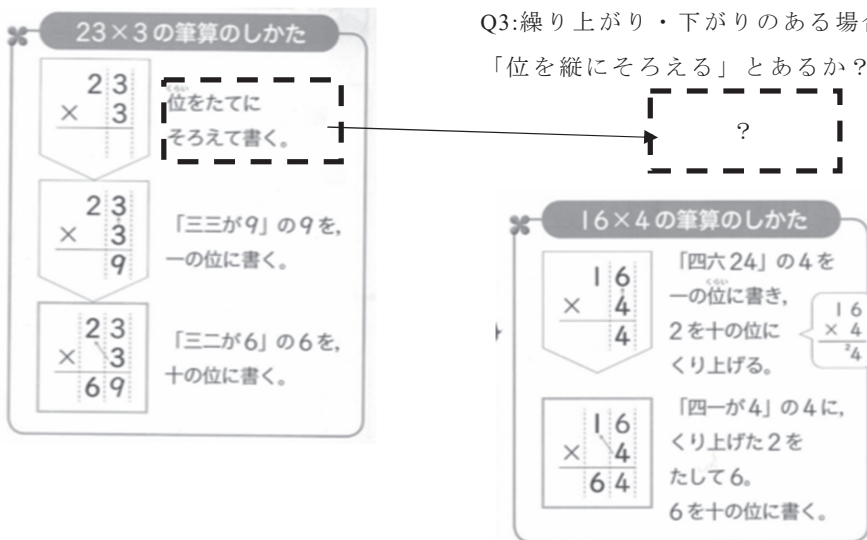


図4 筆算の意味理解の分析の観点(2) 図は小山正孝・飯田慎司ほか(2020)『小学算数2上』日本文教出版,p9・p11より

表 2 繰り上がり・下がりあり加法・減法・乗法のアルゴリズムとモデル操作について

教科書会社	繰り上がりがある加法筆算 (2位数)+(2位数)	繰り下がりがある減法筆算 (2位数)-(2位数)	繰り上がりがある乗法筆算 (2位数)×(1位数)
	分析の観点 Q1: モデル操作と筆算の式操作は対応しているか? Q2: 繰り上がり・下がりの数が、筆算の演算空間・和(差)空間にどちらに記述されているか? Q3: 繰り上がり・下がりのある筆算にある、アルゴリズムの文言の1番目に、「位を縦にそろえて書く」とあるか?		
啓林館	モデル操作と式操作は一致 繰り上がりの数は演算空間 「位をそろえる」あり	モデル操作と式操作は一致 繰り下がりの数は演算空間 「位をそろえる」あり	モデル操作と式操作は一致 繰り上がりの数は和空間 「位をそろえる」なし
教育出版	モデル操作と式操作は不一致 繰り上がりの数は演算空間 「位をそろえる」あり	モデル操作と式操作は一致 繰り下がりの数は演算空間 「位をそろえる」あり	モデル操作と式操作は一致 繰り上がりの数は和空間 「位をそろえる」あり ※「一の位の計算」・「十の位の計算」という表記あり
大日本図書	モデル操作と式操作は不一致 繰り上がりの数は演算空間 「位をそろえる」あり	モデル操作と式操作は一致 繰り下がりの数は演算空間 「位をそろえる」あり	モデル操作はなく式操作のみ提示 繰り上がりの数は和空間 「位をそろえる」なし
学校図書	モデル操作と式操作は一致 繰り上がりの数は和空間 「位をそろえる」あり	モデル操作と式操作は一致 繰り下がりの数は演算空間 「位をそろえる」あり	モデル操作はなく式操作のみ提示 繰り上がりの数は和空間 「位をそろえる」なし
東京書籍	モデル操作と式操作は不一致 繰り上がりの数は演算空間 「位をそろえる」あり	モデル操作と式操作は一致 繰り下がりの数は演算空間 「位をそろえる」あり	モデル操作はなく式操作のみ提示 繰り上がりの数は和空間 「位をそろえる」なし
日本文教出版	モデル操作と式操作は不一致 繰り上がりの数は演算空間 「位をそろえる」あり	モデル操作と式操作は一致 繰り下がりの数は演算空間 「位をそろえる」あり	モデル操作はなく式操作のみ提示 繰り上がりの数は和空間 「位をそろえる」なし ※「一の位の計算」・「十の位の計算」という表記あり

表 2 より 2 点示す。1 つは、初発である繰り上がりのある加法筆算の学習では、モデル操作と筆算の式操作を対応させることが、繰り下がりのある減法筆算と繰り上がりがある乗法筆算にもつながるため、概念形成上重要である。この過程が明確なのは 2 社である。学校図書は、和空間で一の位の数を足す操作をしている。このことはモデル操作からは自然な動きである。すると、筆算での繰り上がりの補助数字 1 は和空間に置くことがこどもにとってわかりやすいはずである。または、啓林館のように、モデル操作の前に演算空間で和が十より大きくなると見積もり、和空間での操作をする前に繰り上がりの補助数字 1 を演算空間に置くことも考えられる。これらの過程での見方は繰り下がりのある場合につながる。それは、繰り下がりのある減法筆算は、取るというモデル操作が演算空間で行われるため、繰り下がり補助数字 1 は演算空間に置くことになる。よって減法筆算は 6 社ともモデル操作と筆算の式操作が対応している。そして、乗法筆算は、モデル操作と筆算の式操作の対応を示しているのが啓林館と教育出版の 2 社である。2 位数×1 位数の乗法筆算は、2 位数×2 位数につなげるために部分積の和について、最初はモデル操作をもとに示すことが必要である。それは、2 位数×1 位数は部分積の和を省略できるという筆算のよさが認識でき、これをもとに部分積の和は省略しない 2 位数×2 位数の計算に発展するからである。

2 つは、繰り上がりのない乗法筆算と繰り上がりのある乗法筆算において、「位を縦にそろえる」と「一の位の計算」「十の位の計算」というアルゴリズムの 1 番目と 2 番目の文言があったのは、教育出版だけであった。筆算のアルゴリズムの 1 番目の「位を縦にそろえる」という文言については、整数は十進位取り記数法表現のため、縦にそろえることで位に着目した方が計算しやすくなるという、整数の加法・減法・乗法筆算における一貫した原理である。ただし、整数が小数に拡張されると、乗法の場合のみ右側にそろえると調整をする必要がある。しかし、この調整も整数還元の見方をすれば容易に理解できる。そのため、1 番目の「位を縦にそろえる」という文言は重要である。また、アルゴリズムの 2 番目の文言である「一の位の計算」・「十の位の計算」という表記についても、積の一の位を求めること、積の十の位を求めることと、いう意味であり、加減筆算にある「一の位の計算」「十の位の計算」と整合しているのでこの文言も必要である。

以上、算数科の検定教科書 6 社の旧版と現行の除法筆算および現行の加法・減法・乗法筆算での指導の実態をふまえ、A の立場：従来のアルゴリズムを意識した筆算の教材価値について再考し、A の立場を具現した上での B の立場：筆算の学習でアルゴリズムを記号化しフローチャートを用いた筆算の教材価値について述べる。

(2)筆算の教材価値の再考

A：の立場での筆算学習の教材価値は概ね次の A1 と A2 の 2 つであると考える。

A1：既習の見方をもとに筆算をつくることができること(加法→減法→乗法→除法)

アルゴリズムとは手順でもあり考え方でもある。以下に、筆算のアルゴリズムを①②③④と表すことにする。すると、アルゴリズムの理解と付加・修正によって、加法→減法→乗法→除法と構成できることが教材価値である。

(加法) 図5にあるように、(2位数)と(2位数)の加法で横書き計算を行う場合には、それぞれの位同士の数を計算することに気がつく。整数は十進位取り記数法表現のため、縦にしてそろえて書けば十の位と一の位が簡単にそろえよう。このようにして「なぜ、2数を縦にそろえるのか(why①)」について納得した上で、繰り上がりのない場合で筆算という新しい形式を初発に学習する。Ⅰ. の繰り上がりのない筆算を原型として、繰り上がりの原理「下の位10を上への位1にすること」が追加され、Ⅱ. の筆算ができる。次に、繰り上がりの原理を2回行うことによって、Ⅲ. の筆算ができる。このように既習の見方をもとにして、加法筆算をつくることができる。ただし、「なぜ、一の位から計算するのか(why②③)」についても重要であり、この加法筆算アルゴリズムの合理性を納得させておく必要がある。

$$45 + 23 = (40 + 5) + (20 + 3)$$

$$= (40 + 20) + (5 + 3) = (5 + 3) + (40 + 20) \quad \text{位同士の計算}$$



「なぜ、2数を縦にそろえるのか(why①)」

Ⅰ. 繰り上がりのない筆算(繰り上がり 0回)		
$\begin{array}{r} 45 \\ + 23 \\ \hline 68 \end{array}$	①位を縦にそろえる ②一の位・十の位の計算 ③計算終了	(初発に学習する筆算の原型)



「なぜ、一の位から計算するのか? (why②③)」

Ⅱ. 一の位または十の位が繰り上がる筆算(繰り上がり 1回)

$\begin{array}{r} 45 \\ + 27 \\ \hline 72 \end{array}$	①位を縦にそろえる ②一の位の計算 ③十の位の計算 ④計算終了	$\begin{array}{r} 45 \\ + 83 \\ \hline 128 \end{array}$	①位を縦にそろえる ②一の位の計算 ③十の位の計算 ④計算終了
	③十の位に繰り上げる ④十の位の計算		③十の位の計算 ④百の位に繰り上げる



Ⅲ. 一と十の位の両方が繰り上がる筆算(繰り上がり 2回)

$\begin{array}{r} 45 \\ + 87 \\ \hline 132 \end{array}$	①位を縦にそろえる ②一の位の計算 ③十の位の計算 ④百の位に繰り上げる ④計算終了
	③十の位に繰り上げる ④十の位の計算 ④百の位に繰り上げる

図5 アルゴリズムの付加・修正による(2位数)+(2位数)の加法筆算の構成

(減法) 図 5 の加法筆算をもとに、減法筆算をつくることができる。このときの共通原理は、「①位を縦にそろえること」と「②一の位から計算すること」である。違いは、繰り上がる場合は「下の位 10 を上の位 1 にすること」、繰り下がる場合には「上の位 1 を下の位 10 にすること」であり加法と減法が逆演算の関係であると再認識できる。

(乗法) 加法筆算と減法筆算をもとに、乗法筆算をつくる。横書きで計算し、乗数を被乗数の十の位と一の位に分けて、それぞれの位の数にかけるため、手順①にあるように 2 数を縦にそろえることで位が見えやすくなる。手順②については、部分積の和の式があれば、乗数 3 を $3 \times 20 = 60$, $3 \times 4 = 12$ と十の位からかけても計算できる。しかし、筆算は簡単に計算するためのツールであるので、部分積の和の式を省略した場合は、一の位から計算しないと繰り上がりに対応できないため、乗数 3 を被乗数の一の位からかけるのである。さらに、「なぜ乗数 3 をかけるのか」については後述するが、位に分けたことで答えの一の位を確定し、順に十の位、百の位、と順に確定できるシステムということを強調する。

$$12 \times 4 = (10 + 2) \times 4$$

$$= 10 \times 4 + 2 \times 4 (= 2 \times 4 + 10 \times 4) \quad \text{乗数 4 を 12 の十の位と一の位の数にかける}$$

↓ 「なぜ、2 数を縦にそろえるのか(why①)」調整的に保存

I. 繰り上がりのない筆算(繰り上がり 0 回)

1 2	①位を縦にそろえる
× 4	②一の位・十の位の計算
4 8	③計算終了

↓ 「なぜ、乗数 3 を被乗数 24 の一の位にかけるのか?(why②③)」

II. 一の位または十の位が繰り下がる筆算(繰り上がり 1 回)

2 4	①位を縦にそろえる	6 2	①位を縦にそろえる
× 3	②一の位の計算	× 3	②一の位の計算
7 2	③十の位の計算 十の位に繰り上げる	1 8 6	③十の位の計算 百の位に繰り上げる
	④計算終了		④計算終了



III. 一と十の位の両方に繰り上がる筆算(繰り上がり 2 回)

5 3	①位を縦にそろえる
× 7	②一の位の計算
3 7 1	③十の位の計算 百の位に繰り上げる
	④計算終了

図 6 アルゴリズムの付加・修正による(2 位数)×(1 位数)の乗法筆算の構成

(除法)除法筆算は、加法筆算→減法筆算→乗法筆算と整合的に構成できたことに比べると、手順は大きく変更する。手順①は「わる数を左、わられる数を右にかき、答(商)は被除数の上に位ごとに書く。」となる。手順②③は一の位からでなく「上の位から計算」となり、この理由を納得させる必要がある。さらに、商の上位の位から求める計算毎に「たてる→かける→ひく→おろす」という手順を繰り返す。さらに、余りが除数より小さくなるときにこの計算が終わるといふ終了の手順④が新たに加わるのである。

$$96 \div 3 = (90+6) \div 3$$

$$= 90 \div 3 + 6 \div 3$$

↓ 「なぜ、わる数を左、わられる数を右にかくのか(why①)」

I. $96 \div 3$ のような余りのない筆算(繰り下げ 0 回)

$$\begin{array}{r} 32 \\ 3 \overline{) 96} \end{array}$$

①わる数を左、わられる数を右にかく
②一の位・十の位の計算
③計算終わり

↓ 「なぜ、被除数 72 の上(十)の位から除数 3 でわるのか? (why②③)」

II. $72 \div 3$ のような余りのない筆算(繰り下げ 1 回)

$$\begin{array}{r} 24 \\ 3 \overline{) 72} \\ \underline{6} \\ 12 \\ \underline{12} \\ 0 \end{array}$$

①わる数を左、わられる数を右にかく
②十の位の計算 たてる→かける→ひく
③一の位の計算 たてる→かける→ひく
④計算終了

余り < 除数なので(われない)おろす

① 2

↓

III. $74 \div 3$ のような余りのある筆算(繰り下げ 1 回)

$$\begin{array}{r} 2 \\ 3 \overline{) 74} \\ \underline{6} \\ 14 \\ \underline{12} \\ 2 \end{array}$$

①わる数を左、わられる数を右にかく
②十の位の計算 たてる→かける→ひく
③一の位の計算 たてる→かける→ひく
④計算終了 余りが 0 でなくても終了

余り < 除数なので(われない)おろす
余り < 除数(われない)

図 7 アルゴリズムの付加・修正による(2位数)×(1位数)の乗法筆算の構成

A2: 論理的思考力を高めることができる

そもそも何のために筆算を行うのかという筆算の意義の理解があり、アルゴリズムの論理性を意識した学習につながる事が大切である。すでにつくられたアルゴリズムを、暗記して早く正確にということだけを目的として計算するためではない。例えば、繰り上が

りのある加法の筆算は図 8 のように、部分和が書かれているのであれば、一の位から計算しても十の位から計算してもできるのであるが、加法筆算は通常は部分和をかかないため、一の位から計算することによって、二度手間を避け計算過程をより簡単にするという論理性がある。また、繰り上がりの数について筆算のどの箇所に書くとよいのか、モデル操作による納得と、既習である繰り上がりのない加法筆算との整合的な納得が必要である。このように筆算の学習では論理的な思考力を高めることができる。



図 8 筆算の意義の理解について(1 段で書くとしたらどちらがわかりやすいか)

図は一松信ほか(2020)『小学校算数 2 上』学校図書,p39 より

さらに、乗法筆算での乗数からかける理由について補足する。日本の場合、図 9 の 4 番目にある手順で行う。この手順は乗数からかけるので、式を見る視点が下から上、さらに被乗数は右から左となり、一番目の手順と比べれば視点移動がしにくいものとなっている。しかし、一番目の手順は概数として数の大きさがわかるのに対して、日本式の手順は一の位からの数が確定するため、正確な計算をするためという論理性がある。

			日本式 かけ算の筆算
① $\begin{array}{r} 34 \\ \times 56 \\ \hline 168 \\ 224 \\ \hline 1904 \end{array}$	② $\begin{array}{r} 34 \\ \times 56 \\ \hline 224 \\ 168 \\ \hline 1904 \end{array}$	③ $\begin{array}{r} 34 \\ \times 56 \\ \hline 170 \\ 204 \\ \hline 1904 \end{array}$	④ $\begin{array}{r} 34 \\ \times 56 \\ \hline 204 \\ 170 \\ \hline 1904 \end{array}$
被乗数からかけるので 視点は上→下に移動する	被乗数からかけるので 視点は上→下に移動する	乗数からかけるので 視点は下→上に移動する	乗数からかけるので 視点は下→上に移動する

計算を始める数を決め、その数の真下に計算結果を記す
計算は位の低い数から順に計算する

図 9 乗法筆算でのかけ算の順番について

次に、B の立場である「筆算の学習でアルゴリズムを記号化しフローチャートを用いた場合の教材価値」について議論する。アルゴリズムは考え方であるのに対して、フローチャートは図式であるので、可視化しやすいこと、処理の手順に問題点があれば発見・修正しやすいこと、問題点を複数で改善でき精度を上げることができるよさがある。しかし、手順の理由を納得しながらアルゴリズムの学習を先に行うため、フローチャートはアルゴリズムの見直しやその手順の論理的関係の再確認のためにあると考える。フローチャート作成活動を分類すると概ね 3 タイプあり、B1 と B2 の教材価値があると考えられる。

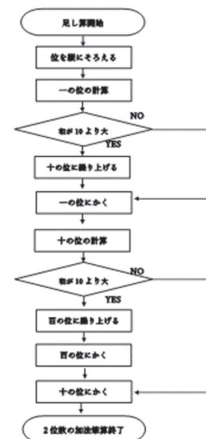
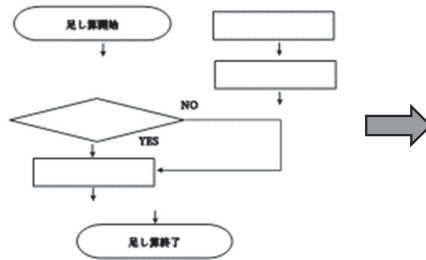
B1 筆算で使われていた考え方を整理したり手順の意味を発見したりすることができる

習得過程でなく習得後に筆算の考え方や手順の意味を再発見させる。例えば、減法筆算の習得後加法筆算と対比し、繰り上がり・下がりが逆の原理であることに気づく等である。

＜フローチャート図そのものを作成する活動＞

(例) 計算の仕方について図に整理しよう

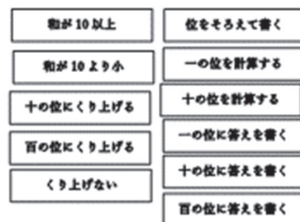
$$\begin{array}{r} 56 \\ + 78 \\ \hline 134 \end{array}$$



＜カードに手順がありそれらのカードを並べ替えてフローチャートとして作成する活動＞

(例) 計算の仕方についてカードを使ってまとめよう

$$\begin{array}{r} 56 \\ + 78 \\ \hline 134 \end{array}$$



＜筆算に文字が配列されており、それらの文字をフローチャートにあてはめる活動＞

(例) E, F, Gにはどんな数があてはまりますか？関係を整理しましょう

$$\begin{array}{r} A B \\ + C D \\ \hline E F G \end{array}$$

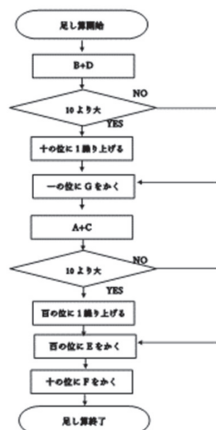
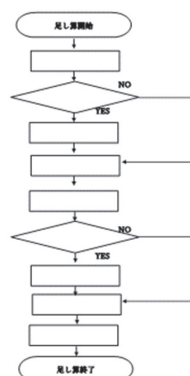


図 10 筆算のアルゴリズムをフローチャートに表現する活動

B2 一見複雑そうな筆算も単純化・抽象化して捉えることができる

加法筆算と減法筆算と乗法筆算は、一の位から順に上の位への計算を行うため、「順次処理」の考え方を学習することができる。さらに、繰り上がりあり・繰り下がりありのフローチャート作成の学習からは、「条件分岐処理」の考え方を学習できる。

除法筆算では、通常十進位取り記数法表現に依存して上位の位から順に計算を行う。これに対して除法の原理をもとに「反復処理」の考えを使えば、図 11 のように簡単に表現することができる。この意味では論理的な思考力を高めることができる。

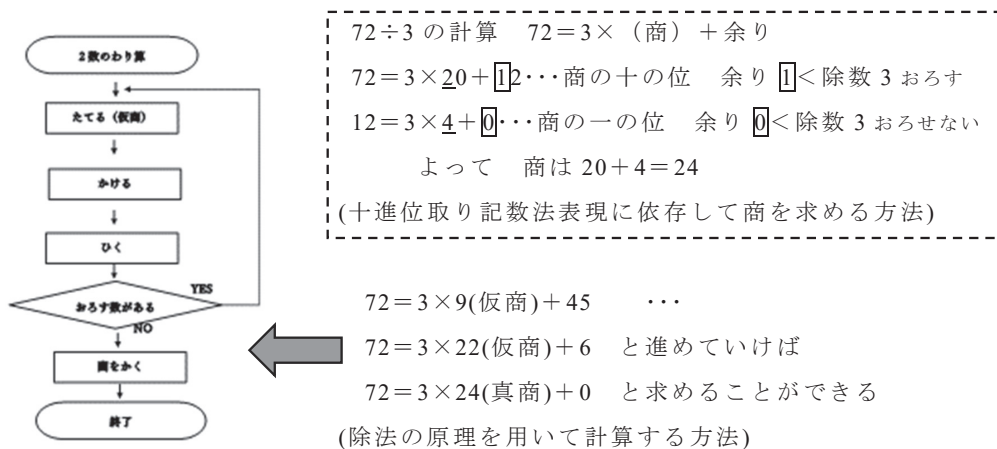


図 11 除法筆算のフローチャート

しかし、プログラムやフローチャートそのものの学習でないため、筆算の認識においては、フローチャートがどう役立つかと考える必要がある。すると、図 11 からは、こどもにとって一見複雑な手順だったと感じた除法筆算について、実は「たてる(仮商)→かける→ひく→おろす」という考えを繰り返し使っているだけであると気づくことができるのである。

3. まとめと課題

筆算とプログラミング的思考との関係を考察するとき、アルゴリズムとフローチャートの関係をどう捉えるかが大切である。アルゴリズムは考え方を含んだ手順であるので、汎用性をもつ。そのため、2位数+2位数の加法筆算を原型として、減法筆算、乗法筆算、さらに、付加・修正が大きくなるが除法筆算のアルゴリズムをつくることができる。

その過程に図式であるフローチャートが、筆算の認識をどのように高めることができるかについてまとめる。

1 点目は、フローチャートを使う場面は筆算の習得後がよいと考える。単元内で扱うとすれば、図 10 の 3 つめの活動を行えば論理的な確かめができる。例えば、文字を使いできあがった筆算を与えて、計算順と起こりうる場合を図示する課題を設定すれば、フローチャート図という形式を意識しなくても流れ図的な表現ができる。その中に「条件分岐」の考え方を発見させることもできる。

2点目は、筆算アルゴリズムと対応してフローチャートをしっかり作成させるとすれば、2位数+2位数の繰り上がりのない加法筆算のフローチャート作成を原型として、繰り上がりのある加法筆算のフローチャート作成というように系統的に学習させることも考えられる。その際に、筆算とは、十進位取り記数法表現に依存するため複雑な計算が簡単になった手順であると見直しができることを期待したい。

課題としては、いくつかのフローチャートを用いた実践例に、図形の包摂関係の理解においてフローチャートを用いたものがある。図形概念形成に役立ったとあるが、図形概念について弁別区分型の認識から包摂分類型の認識はフローチャートだけでは簡単にはできないと考える。これは次の課題としたい。

【引用文献】

- 1) 文部科学省(2020)『小学校プログラミング教育の手引(第三版)』
- 2) 星野将直(2021)『正多角形の認識を高めるプログラミングの在り方』愛知淑徳大学学び舎 17
- 3) 小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ)
https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm(最終閲覧 2023.1.25)
- 4) 文部科学省(2018)『小学校学習指導要領 解説算数編』日本文教出版
- 5) 文部科学省(2018)『小学校学習指導要領 総則編』日本文教出版
- 6) 一松信ほか(2020)『小学校算数 2上・3下・4年上』学校図書
- 7) 坪田耕三・金本良通ほか(2020)『小学算数 2上・3下・4上』教育出版
- 8) 橋本吉彦・相馬一彦ほか(2020)『たのしい算数 2・3・4年』大日本図書
- 9) 小山正孝・飯田慎司ほか(2020)『小学算数 2上・3下・4年上』日本文教出版
- 10) 藤井齊亮・真島秀行ほか(2020)『新しい算数 2上・3下・4上』東京書籍
- 11) 清水静海・船越俊介ほか(2020)『わくわく算数 2上・3下・4上』啓林館

【参考文献】

- ・ 清水匠・中川一史(2019)『小学校プログラミング教育におけるフローチャートづくりとICT活用に関する考察』日本STEM教育学会 2019年3月拡大研究会予稿集
- ・ 大日本図書(2020)『スタート!プログラミング教育』茨城プログラミング教育研究会
- ・ 鈴木はるか他(2018)『プログラミング教育を取り入れた授業実践』熊本大学PCカンファレンス
- ・ 杉能道明(2013)『筆算指導の今日的意義についての考察』岡山大学・数学教育学会誌パピルス第20号
- ・ 佐藤茂太郎(2020)『小学校第4学年算数教科書におけるわり算の筆算の説明の特徴』教育総合研究第4号