

# アルゴリズム理解能力の分析II

— 学生の特性からみたアルゴリズム理解能力 —

森田 彦

学生にプログラミング教育を施す場合、多くの学生がアルゴリズムの理解に困難を感じるという点が障害となることが指摘されている。この問題を解消するためには、まず学生がどのようなところでつまづいているのか、という実態を把握した上で、そのつまづきの原因を明らかにすることが必要である。

本研究では以上の観点から、学生のアルゴリズム理解度を問うテストを実施し、その結果の分析から学生の理解度の実態について議論する。次に、それらテストができる学生がいかなる特性を持っているのかを、別個に行った「学習意欲」、「学習技能」そして「創造性」を測定するアンケート調査結果との相関分析から明らかにし、アルゴリズムの理解・設計にいかなる資質が必要とされるのかについて議論を行っている。

## §1 序

当社会情報学部では専門科目としてプログラミング言語 I (FORTRAN), II (COBOL) が開講されており、学生は少なくともいずれか一つを選択することになっている。これらプログラミング科目の教育の目的は特定のプログラミング言語の習得にあるのではない。我々は、文献<sup>1)</sup>にも指摘されているように、問題解決のアルゴリズムの設計を教えることに教育の目的あるいは意義があると考えている。そして、プログラミングはその教育手段であると位置づけている。教育手段としてみたとき、プログラミングは有効である。というのは、プログラミング言語を通じたアルゴリズムの実行により、その正当性あるいは誤謬を即座に検証できるという教育的効果がそこにあるからである。この観点から我々は、アルゴリズム設計に重点を置いたプログラミング教育を目指している。

ところが実際の教育・指導を通じて、次の点が問題点として表面化して来た。即ち、多くの学生にとって、提示されたアルゴリズムを理解することはできても、能動的にアルゴリズムを考える（設計する）ことは、我々の予想以上に困難のように見えるという点である。ここに、学生に課しているアルゴリズムは、決して複雑なものではなくかなり単純なものである。そこで我々は、まずどこで学生がつまづいているのかを明らかにすることが必要であると考え、学生がつまづきの原因についてこれまで分析を行ってきた。その結果、学生がつまづく原因として以下の2点が浮かび上がってきた<sup>2)</sup>。

### ① 問題内容に対する理解の不足

アルゴリズムを設計できないという事態の多くは、与えられた問題内容に対する理解不足が第一の要因である場合が多い。典型的な具体例で言うと、文系の学生に数学的な知識を要する問題を課しても問題そのものの内容

に対する理解不足あるいは嫌悪感から、アルゴリズムを考える段階にまで達することができない。逆に、数学的な内容を排した問題を与えると、論理手順という意味では少し複雑になってもアルゴリズムを設計できる学生が増加するという傾向が見られる。

## ② 問題の抽象化の困難さ

問題内容を理解できても、それで自動的にアルゴリズムを設計できる訳ではない。アルゴリズム設計の第一段階として、問題の要求内容の本質を把握し枝葉末節を取り去ること、つまり問題を抽象化したレベルで捉えることが要求される。この段階でつまずいていると思われる学生も多い。実際、問題内容は理解できていると思われるのにアルゴリズムを設計できない学生は、本質的ではないと思われる細部に拘泥している場合が多い。この「問題の抽象化」という段階がクリアできれば、自然にアルゴリズムの大まかな全体構造を構築でき、その後個々の細部をトップダウン的に詰めて行くことができると期待される。

上記2点のうち、後者についてはまだ分析が不十分であり、今後さらに分析を進める必要がある。そのためには、学生がどの段階でどの程度つまずいているのか、その現状を的確に把握することが肝要であると我々は考えた。そこで今回、数種類のテストを行い、その結果の分析から学生の理解度をできるだけ定量的に把握することを第一の目標に置いた。テストとしては、前期の半ばに行った理解度チェックテスト、それから前期末に行ったテストの2種類のテストを考察の対象とした。

次に、学生をつまずきの原因を明らかにするために、その逆であるが、今回はつまずかずにそれらテストができるためにはどのような資質が必要とされるのかを明らかにするという観点から分析してみることにした。具体的には、学生の「学習意欲」、「学習技能」そ

して「創造性」を測定する目的で坂元教授らによって考案されたアンケート<sup>3)</sup>を行い、このアンケート結果とテストの成績との相関分析から、アルゴリズムを理解し設計できる学生がどのような特徴を持っているのかを明らかにしようと考えたのである。

以下§2および§3では、それぞれ理解度チェックテストそして前期末のテストのねらいとその成績分布を示し、学生のアルゴリズムに対する理解度の現状について議論する。次に、§4において、アンケートの結果と理解度チェックテストおよび前期末テストの結果との相関分析を行い、これらのテストができる学生の特徴とは何かを調べる。最後に以上のまとめを§5で与える。

## §2 理解度チェックテストの結果

以下、今回の調査の対象となるのは、1994年度にプログラミング言語I・プログラミング演習I (FORTRAN) を履修した社会情報学部<sup>1)</sup>の学生であり、履修登録学生は218名である。この科目は講義と演習が対になっており、講義は新國が、演習は森田が担当している。

まず、6月の下旬に理解度チェックテストを行った。テストの問題は資料1に与えてある。このテストのねらいは、プログラミング言語で表されているアルゴリズムをどれだけ普遍性の高い形で理解できているかを把握する目的で行ったものである。具体的には、それは、問題2と問題3がどの程度できているかということでチェックできる。

ここに、問題2では、最初に「 $SUM=1+2+\dots$ の値が100を越えるのは最後に何を加えた時で、そのときのSUMの値がいくつになるか」を求め表示するプログラムを与えている。そして、以下設問(1)ではこれを「 $SUM=1+3+5+\dots$ の値が300を越えるのは最期に何を加えた時で、そのときのSUMの値がいくつになるか」を求め表示す

プログラミング演習I理解度チェックテスト (F21, F22, F23)クラス	学籍番号	氏名	得点
			/100

※ 答案用紙は2枚ある。2枚それぞれに、学籍番号および氏名を忘れずに記入すること。

【問題1】 以下は2個の整数の平均を求め表示するプログラムであるが、コンパイルすると5ヵ所に文法エラーが検出された。

```

1 C    2数の平均
2 C    平成6年6月22日 作成
3      PROGRAM HEIKIN
4      INTEGER A, B, C
5      A=4
6      B=3
7      (A+B)/2 C=(A+B)/2
8      WRITE(*,*) 'A=', A, 'B=', B, '平均=', C
9      STOP
      END
    
```

設問(1) 上のプログラムに挿入、削除あるいは訂正を施し、エラー箇所を修正せよ。ここに、挿入が必要な場合は挿入箇所に直接書き込み、削除する場合は、削除部分を＝で消すようにせよ。また訂正する場合は＝で該当箇所を消した上で、最寄りの空白に訂正した文を書き込むようにせよ。

設問(2) (1)の修正を施した上で、プログラムを実行すると、画面には [ A= 4 B= 3 平均= 3 ] と結果が表示され、今の場合、平均値が3.5になるはずのところ、このプログラムでは3となってしまう。そこで、平均値が正しく出るようにするにはどのような修正を上記のプログラムに施せばよいかを、以下の記述例にならって記せ。ただし、AおよびBは整数型の変数のままとする。また設問(1)のエラー修正は完了しているものとし、改めてその修正を記入する必要はない。

【記述例】 ○ 行目を \_\_\_\_\_ → \_\_\_\_\_ のように修正する。 ○ 行目を削除する。  
 ○ 行目の下に \_\_\_\_\_ を1行挿入する。等々。

- 4行目を INTEGER A,B,C → INTEGER A,B と修正する。
- 4行目の下に REAL C を1行挿入する。
- 7行目を C=(A+B)/2 → C=(A+B)/2.0 と修正する。

【問題2】 今、SUM=1+2+3+・・・の値を求めることを考える。以下は、SUMの値が100を超えるのは最後に何を加えた時で、そのときのSUMの値がいくつになるかを求め表示するプログラムである。以下の設問に答えよ。

```

PROGRAM P2
INTEGER I, SUM
I=0
SUM=0
100 CONTINUE
IF (SUM.LB. 100 300) THEN
    I=I+1 2
    SUM=SUM+I
GOTO 100
ENDIF
WRITE(*,*) '最後に加えた数=', I, 'そのときの値=', SUM
END
    
```

平成6年6月22日4講時実施 NO-2

プログラミング演習Ⅰ理解度チェックテスト (F21, F22, F23) クラス	学籍番号	氏 名	得 点
			/100

設問(1) 上のプログラムに適当な修正(文の挿入、削除、訂正等)を施し、 $SUM=1+3+5+7+\dots$ が300を越えるのは最後に何を加えた時で、そのときのSUMの値はいくつになるかを求め表示するプログラムに変更せよ。修正は、直接上のプログラムに施せ。修正(文の挿入、削除、訂正等)に当たっては、問題1の設問(1)と同じ要領で行うこと。

設問(2) 上のプログラムを参考にして、 $SEKI=2 \times 4 \times 6 \times 8 \times \dots$ の値が1500を越えるのは、最後に何をかけた時で、さらにそのときのSEKIの値がいくつであるかを表示するプログラムを以下に作成せよ。

```
PROGRAM P2A
INTEGER I, SEKI
```

```
I = 0
SEKI = 1
100 CONTINUE
IF (SEKI .LE. 1500) THEN
    I = I + 2
    SEKI = SEKI * I
    GOTO 100
ENDIF
```

```
WRITE(*,*) '最後にかけた数=', I, 'そのときの値=', SEKI
END
```

【問題3】 奇数Nの値を読み込み、 $SUM=1^2+3^2+5^2+\dots+N^2$ の値を計算して、NおよびSUMの値を表示するプログラムを以下に作成せよ(Nとしては奇数のみを入力することを前提にプログラムを作成してよい)。

```
PROGRAM P3
```

```
INTEGER I, N, SUM
READ(*,*) N
I = 1
SUM = 0
100 CONTINUE
IF (I .LE. N) THEN
    SUM = SUM + I**2
    I = I + 2
    GOTO 100
ENDIF
```

```
WRITE(*,*) 'N=', N, '和=', SUM
END
```

るように変更し、すなわち奇数の和となるように変更し、設問(2)では「SEKI=2×4×6×・・・」というように偶数の積となるように変更することを求めている。さらに問題3においては、「SUM=1<sup>2</sup>+3<sup>2</sup>+5<sup>2</sup>+・・・+N」の値を求めるプログラムの作成を求める。これらのプログラムのアルゴリズムは、いずれもある条件が成り立っている間はある演算を繰り返して行い、という意味で抽象的なレベルでは同一である。従って、そのような抽象的なレベルでアルゴリズムを理解できていれば、問題2および3は解答できるはずであると考えたのである。

成績分布は図1に与えている。ここに、成績は0～3までの4段階に分けてつけられており、「3」が完全にできている場合である。「2」は1カ所ミスがある場合であるが、これはケアレスミスの類であるので、ここでは2以上であればアルゴリズムを理解しているとみなした。図より、「2」以上、つまりアルゴリズムを的確に理解していると思われる学生数は受験者194名中113(名)→96(名)→52(名)と推移している。割合で言うと、58.5(%)→49.5(%)→26.7(%)と減少して行っている。この結果によると、抽象的なレベルでアルゴリズムを捉えている学生は、全体の3割弱程度であることになる。また、問題2までできている学生の内、問題3ができている学生は約半数である。アルゴリズムを受動的に理解することはできてもそれを応用

することができない、という我々が実際の指導に際して直面する実態がこの数値に現れている。

### §3 前期末テストの結果

前期末テストはプログラミング言語の講義担当の新國が行った。そのテスト問題は資料2に与えてある。

ここに注目したい点は問題2である。この問題は日本語で記述されたアルゴリズムをPAD (Problem Analysis Diagram) によって表すことを要求する問題である。上の理解度チェックテストでは、FORTRAN 言語で表現されたアルゴリズムをどこまで理解しているのかを確かめる問題であったが、ここではそれが日本語で表現されており、その分だけ抵抗が少ないと思われる。つまりプログラミング言語の知識に関係なくアルゴリズムの理解度を確かめることができると考えられる。そのことを念頭において図2の結果をみてみよう。

ここに、問題2の成績は5段階に分けており、アルゴリズムの構造が分かっていると思われるのは「4」および「5」のグループであり、「3」以下は構造に関わる誤りを犯しており、アルゴリズムの理解に問題があるグループとみなすことができる。そこで「4」および「5」の合計を求めると、受験者212名中79名で、割合は全体の37.3(%)となって

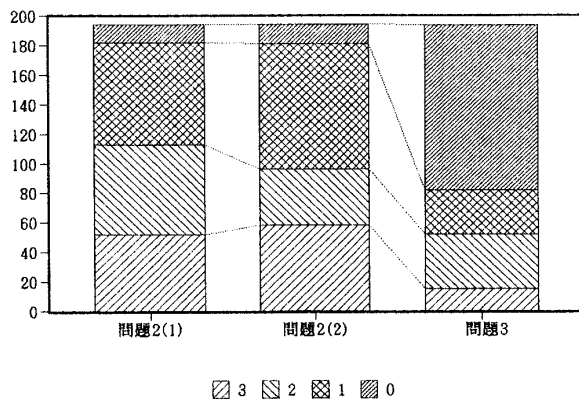


図1 理解度チェックテストの結果

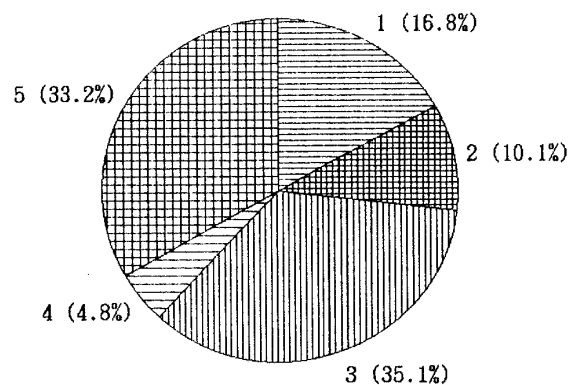
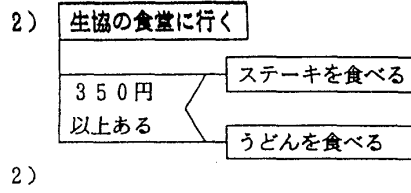
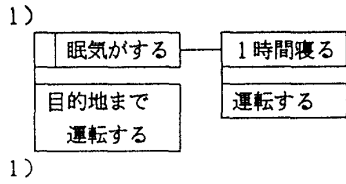


図2 前期末テストの結果

1. 下記のPADを日本語で記述せよ。



2. 次の記述を1)～2)の順にPAD(日本語を用いて)で表現せよ。

箱Aの中に白と黒の玉、箱Bには黒の玉がそれぞれ何個か入っているとす。「箱Aの中から玉を同時に二つずつ取り出し、もし同色ならば二つとも捨て、箱Bに入っている黒玉を一つ箱Aに入れる。もし同色でなければ、黒玉を捨て、白玉を箱Aにもどす。この操作を、箱Aの中に残る玉が2個以上ある間続ける。」

- 1) 上記「 」部分中の下線部分
- 2) 上記「 」全体

3. 1)から3)の問いの空欄をうめよ。

```

1) INTEGER N1, N2
   N1=-1
   N2=N1+4
   N1=N1+N2
   N2=N1+N2
   WRITE(*,*) N1, N2
   END
  
```

↓ 実行結果

N1  N2

```

2) INTEGER K, N
   WRITE(*,*) '二つ自然数を入れて下さい:'
   READ(*,*) K, N
   IF (K.GE.N) THEN
     K=K+N
   ELSE
     N=K+N
   ENDIF
   WRITE(*,*) K, N
   END
  
```

↓

入力データを  
5, 2  
とした時の  
実行結果は?

↓

K  N

```

3) INTEGER NO, I, N
   REAL W, H, S
   N=0
   WRITE(*,*) '人数を入れて下さい:'
   READ(*,*) NO
   DO 100 I=1, NO
     WRITE(*,*) '身長と体重:'
     READ(*,*) H, W
     S=(H-100)*0.9
     IF (W.GT.S) THEN
       WRITE(*,*) I, '番目の方は太りすぎです'
       N=N+1
     ELSE
       WRITE(*,*) I, '番目の方は太っていません'
     ENDIF
   100 CONTINUE
   WRITE(*,*) N ←(b)
   END
  
```

人数を入れて下さい: 5	
身長と体重:	150, 46.8
身長と体重:	170, 70.4
身長と体重:	160, 52.5
身長と体重:	180, 71.8
身長と体重:	190, 85.5

(入力データの画面)

①上記の右枠内のデータが入力されたとする。

(a) 部分は何回実行されるか?

回

②(a) 部分が繰り返し実行された結果、画面に表示される内容を順に書け。

③プログラム中の(b)で表示されるNの値は?

④このプログラムは何するプログラムか?日本語で記せ。

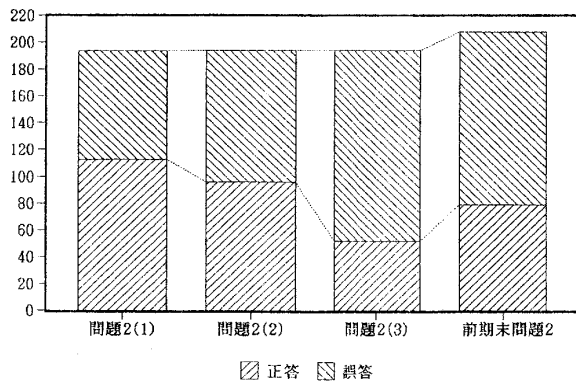


図3 理解度チェックテストと前期末テストの比較

いる。

ここで、この結果を理解度チェックテストと図3で比較してみる。ここで、各問題の難易度を考察してみよう。まず、理解度チェックテストの問題2の設問(2) (以下 [理2] と略記) では与えられたアルゴリズムを若干変更しなければならぬが、類似したアルゴリズムがすでに FORTRAN 言語で表現されているので、頭の中でアルゴリズムを一から設計しなくても与えられたプログラムの部分的修正で済む。これに対して、前期末テストの問題2 (以下 [前2] と略記) では、日本語で表現されたアルゴリズムを一旦頭の中で構築してから PAD で表現することが求められる。このように、前者では、アルゴリズムの表現形態が FORTRAN 言語→FORTRAN 言語と同一であるのに対し、後者では日本語→PAD と変更されているのである。このため後者においてはより一層シビアに、表現形態に拘泥することなくアルゴリズムを抽象的なレベルで理解することが求められることになる。その結果、[理2] より [前2] の正答率が低くなったものと考えられる。ところが、理解度チェックテストの問題3 (以下 [理3] と略記) になると、プログラムの修正の度合いが増し、結果として [前2] よりも難易度が増し、正答率が下がったのであろう。現在、これ以上の分析は進んでいないが、これらのデータはアルゴリズムを抽象的なレベルで理解することの困難さの定量的把握につながる

ものと考えられるので、今後さらに分析を進めて行きたい。

#### §4 テスト結果とアンケート結果との相関分析

前2節において、学生のアルゴリズムに対する理解度の実態について調べてきたが、ではアルゴリズムをよりよく理解している学生とはどのような学生であろうか。この点が明らかになれば、アルゴリズムの理解に必要な要素が明らかになり、逆にいうとできない学生がつかずく原因を調べる際のヒントを与えてくれるものと期待される。

そこで、坂元教授らによって提唱されている学生の「学習意欲」、「学習技能」および「創造性」を測定するアンケート調査<sup>3)</sup>を学生に対して実施し、その結果とテスト結果との相関分析から、テストがよくできている学生の特徴を分析してみることにした。アンケート項目は、本ワークショップの新國の報告において、表1～表3として与えられている。このアンケートでは各項目について、「非常に当たっている」から「全く当たっていない」まで5段階の選択枝のいずれかを選択する形式になっている。アンケート調査は、新國がプログラミング言語の講義時間中に行った。

まず、[理3]のテスト結果とアンケート結果との相関関係を調べ、正の相関が強いアンケート項目をピックアップした。結果は表1にまとめてある。ここに最右欄の数値は相関係数であり、( )内の数値はP値である。ここでは、P値が5(%)未満のものを相関がある項目の候補としてピックアップしている。表より、相関係数は必ずしも大きな値ではなく、この分析のみから明確な結論を引き出すことは困難であるが、これらは他の色々な結果を分析・理解する際の基礎データになると我々は考えている。

さて、表中の項目は、「わかるまで・・・」、「ジックリ・・・」そして「納得がいくまで・・・」

表1 理解度チェックテスト問題3と正の相関がある項目

技能	わからなかったところはわかるまで読んでいます。	0.279(0.0002)
創造	一つの考えをジックリ追求する。	0.198(0.0084)
意欲	分からない時には、納得がいくまで調べます。	0.178(0.0200)
意欲	他人に頼らずに課題をやりとげます。	0.169(0.0272)
技能	わからなかったところを調べています。	0.164(0.0285)
技能	繰り返しておぼえています。	0.153(0.0410)

表2 前期末テスト問題2と正の相関がある項目

意欲	能率が上がる様に時々勉強の仕方を変えてみます。	0.172(0.0214)
意欲	失敗をした時、原因をつきとめようとします。	0.165(0.0265)
意欲	予想と結果が違う時は、その原因を調べます。	0.164(0.0283)
技能	わからなかったところはわかるまで読んでいます。	0.156(0.0329)
創造	応用力のある。	0.152(0.0386)
技能	はじめに予想を立てて解いています。	0.147(0.0442)

等という「根気強さ」あるいは「粘り強さ」を表す特性と「他人に頼らずに・・・」という自立性を表す特性がみとれる。以上をまとめると、[理3]を解ける学生の特性として「他人に頼らずに、ジックリと自分が納得のいくまで考え、根気強く物事の解決に当たる」という特徴が浮かび上がってくる。このように、プログラミング言語によって表現されたアルゴリズムを理解し、それを応用するためには、粘り強くジックリと考えることが必要とされるものと思われる。

次に、[前2]についても同様にアンケート結果との相関を調べた結果が表2である。表より、相関のある項目は、[理3]の場合と異なることが分かる。[前2]を解ける学生の場合には、「能率が上がる様に工夫したり、失敗した場合の原因を分析・調査する」という傾向がみとれる。大まかに言えば、[理3]では、根気や粘り強さが求められのに対して、[前2]では、分析や調査という特性が求められ

ていることになる。特に後者の場合、原因の究明ということがポイントになっていると考えられる。

我々はこの両者の違いを次のように理解している。前節でも述べたように、[前2]の場合、アルゴリズムが日本語で表現されていることから、その意味を理解するには困難はないが、論理構造を視覚的にPADで表現するためには自分の頭の中で一旦その処理手順の論理構造を構築することが求められる。ここに、「原因を究明する」という分析的思考が必要とされるのでないであろうか。一方、[理3]においては、プログラミング言語で表現されたアルゴリズムの理解に「根気や粘り強さ」が必要とされることに加え、[理3]がテストの最後の問題であり、かつ[理2]の理解を前提とした問題であるため、辛抱強く問題を解き進んで行く姿勢が必要とされた可能性が高い。また、すでに参考となるアルゴリズムがFORTRAN言語によって提示されており、これは日本語の文章による表現よりも論理構造を視覚的に把握しやすいため、[前2]程には、「原因を分析・究明する」という特性は必要とされなかったと考えられる。

ところで、[理3]において「ジックリと自分が納得のいくまで考える」という特性が必要とされるのであれば、同じくアンケート項目「創造性」の2番目にある「熟考型」の項目とも正の相関があってしかるべきである。しかし、実際には正の相関はなく、一見するとこれは矛盾である。我々は、このことは、ただ熟考するというだけでは不十分であり、何か他の要素が伴って初めて正の相関が生ずるということを示唆するものと考えた。そこで、伴うべき要素つまり前提となる要素とは何かを以下の方法で調べた。すなわち、アンケートのある項目について、その傾向がある集団、ここでは選択枝の3, 4, 5を答えた集団のみを対象を限定して、その集団の中で、[理3]と熟考型との間に正の相関が現れるか



表3 熟考型と理解度チェックテスト問題3および前期末テスト問題2との相関

	全 体	見通しのきく (3+4+5)
理解度チェックテスト問題3	0.115(0.1290) 177	0.182(0.0354) 134
前期末テスト問題2	0.124(0.0929) 186	0.204(0.0147) 142

否かを、全てのアンケート項目について行ったのである。その結果、「創造性」の20番目「見通しのきく」という項目について3, 4, 5を答えた集団について考えると、熟考型との間に正の相関が現れることが分かった。その結果を表3に示す。このことは、「ジックリと自分が納得のいくまで考える」という特性は、ただやみくもに考えるのではなく、「見通しをもった上でよく考える」ことを意味しているものと考えられる。

同様の調査を[前2]についても行ったが、表3に示すようにやはり[理3]の場合と同じく、「見通しがきく」集団内では、熟考型との正の相関が現れている。つまり、[理3]と[前2]では必要とされる特性は表面的には異なっているが、「見通しをもってよく考える」という特性は共通の要素になっているのである。この点の分析を進めることにより、アルゴリズムの理解に必要とされる特性を把握することができると思われる。

## §5 まとめ

本研究では、学生がアルゴリズムの理解・設計においてつまづく原因を明らかにするためには、まず学生のアルゴリズムに対する理解度の実態を把握する必要があるとの観点から、理解度チェックテストおよび前期末テストを行い、その結果を整理した。これにより、学生の理解度の実態を定量的に示す基礎データがそろい、今後の解析に役立つものと思われる。

さらに、これらテストができる学生がどのような特性を持っているのかを調べるため、

学生の「学習意欲」、「学習技能」および「創造性」を測定するアンケート調査を行い、テスト結果とアンケート結果との相関分析を行った。その結果、プログラミング言語で表現されたアルゴリズムを理解し、更にそれを応用することができる学生には、「他人に頼らずに、ジックリと自分が納得のいくまで考え、根気強く物事の解決に当たる」という特徴があることが分かった。さらに、日本語で表現されたアルゴリズムをPADを用いて視覚的に表現することができる学生の特性には、「能率が上がる様に工夫したり、失敗した場合の原因を分析・調査する」という傾向があった。

両者の特性が異なるのは、それぞれに最も必要とされる資質が異なることを意味するが、両者に共通の要素として、「見通しを持って熟考する」という特徴があることが分析の結果分かった。これらの分析は、まだ緒に付いたばかりで、これ以上の分析に進むためには、坂東教授が指摘されたように、アンケート調査の因子分析を行い、見出した因子を基にテスト結果との相関を分析する必要がある。これについては今後分析を進める予定である。

**謝辞** 本研究において、アンケート調査のデータおよび前期末テストの結果の使用を快く承諾してくれた新國氏、またワークショップの事前に貴重な意見を寄せてくれた能登氏、そしてアンケート調査結果とテスト結果の相関分析の一部を行ってくれた社会情報学部4年の上梨君にここに深謝したい。また、本研究は、札幌学院大学社会情報学部・理系教員プロジェクトの財政的支援を受けて行われた。合わせてここに謝意を表したい。

## 文献

- 1) 情報処理学会：一般情報処理教育の実態に関する調査研究報告書，pp.96-102，(1992)。
- 2) 森田 彦，新國三千代，原田 融：アルゴリ

ズム理解能力の分析, 社会情報, 札幌学院大学  
社会情報学部紀要, Vol.2, No.2, pp.87-99,  
(1993).

- 3) 坂元 昂: 大学教育改善技法, 社会情報, 札幌学院大学社会情報学部紀要, Vol.2, No.2, pp.101-109, (1993).