

アルゴリズム理解能力の分析

森田 彦, 新國三千代, 原田 融

要 旨

アルゴリズムに対する理解能力を決めているのは、いかなる要因であるのか、という点を明らかにするために、プログラミング学習の理解度に関するアンケート調査の結果、および2種類のアルゴリズムテストの成績を基に、包括的分析を行った。

分析結果より、アルゴリズムを作成するには、少なくとも、問題の理解能力、問題の定式化能力そして論理的思考能力が必要とされることを指摘する。さらにデータは、初等的なアルゴリズムを学生が作成できない場合、その原因は問題の理解と定式化の過程にあることを示唆している。

最後に、アルゴリズムテストの成績が、問題の理解能力、問題の定式化能力そして論理的思考能力の総合的結果としてどのように理解されるか、という点について議論を行っている。

§1. 序

本学社会情報学部では1年次の学生に「情報処理」を課し、ワードプロセッサによる文書作成、表計算ソフトウェア使用方法の習得、そしてBASICプログラミングを課している。その成果および問題点についてはすでに指摘したところであるが、特にBASICプログラミングに困難を感じている学生が多いことが分かり、その原因を分析することが課題として残された。

その原因としてまず第一に考えられるのは、アルゴリズムを理解できないという点であろう。なぜなら、アルゴリズムを作成する過程が、プログラミング作成過程において最

も創造的な過程であり、したがって困難度が最も高いと考えられるからである。この意味で、我々は、コンピュータ・プログラミング教育において最も重要なことは、アルゴリズムをいかに理解させるかという点にあると考えている。

アルゴリズムを理解させる教育を考えるには、まず理解できない原因を明らかにすることが第一歩である。そのためには、アルゴリズムを作成する一連の過程において、いかなる要因が関連し、いかなる質の思考能力が必要とされるのか、を明らかにした上で、その中のどれが障害になっているのかを分析して行く必要がある、と我々は考えている。

本報告では、アルゴリズム作成の際に障害となっている要因は何かを明らかにするという観点から、現在2年次の学生を対象に行っ

た、アルゴリズムの理解能力の分析結果について報告する。ここに分析に用いたデータは、昨年度に実施した、「情報処理」の学習内容についての理解度アンケート調査の結果、および、今年度プログラミング言語として FORTRAN を履修した学生に実施した 2 種類のアルゴリズムテストの成績である。

以下、§2 では、「情報処理」アンケート調査結果における BASIC プログラミングの理解度の分析から、問題自体に対する理解度が BASIC プログラミングの理解度に反映されている可能性を示唆し、§3 において、FORTRAN の講義で行ったアルゴリズム理解度チェックテストの結果が、その考えを支持することを示す。さらに、問題の定式化が重要であることを指摘した上で、アルゴリズムの作成という作業が、少なくとも「問題の理解→問題の定式化→処理手順の論理的組み立て」という過程から成っていると考えられることを示す。次に、ある種の思考能力テストが、その定式化能力の一つの指標となり得ることを、アルゴリズム理解度チェックテストと思考能力テストとの相関を見ながら議論する。以上の結果を基に、§4 では、前期末に行ったアルゴリズムテストの成績を問題の定式化能力と論理的思考能力の総合的な結果として見ることができるか、という点について議論する。その際、論理的思考能力の指標として、論述テストの成績を用いているが、そのことの妥当性についても若干議論する。以上の分析のまとめは §5 に与えてある。

§2. 「情報処理」アンケートの分析

まず、昨年度実施した「情報処理」アンケートの回答結果を用い、BASIC プログラミングに関する学生の理解度を調べてみよう。このアンケート調査は 235 名の学生を対象に行い、その回収率は約 7 割であったが、詳細については文献 1) に譲る。

図 1 は、BASIC の学習内容について理解できたと答えた学生の割合を、テキストの各節毎、即ち各学習項目毎に示したものである。ここに実線および点線はそれぞれ、理系学生および文系学生の回答を意味している。本学においては、高校時代に理系志向、あるいは文系志向であった学生の割合が、ほぼ 4 : 5 になっているが(残りは、その他)、各々はそれぞれ相対的に数学が得意なグループとそうではないグループに対応していると捉えてよい。なぜなら、ここにおける理系、文系グループは、入試時に、数学の配点比率が高い甲型で受験したグループと、数学を選択しなくてもよい乙型で受験したグループに、ほぼ対応しているからである。このように、理系、文系グループで比較することの意図は、プログラミングの理解度さらにはアルゴリズムの理解度と、数学の習熟度との関連を見ることにある。さらに、テキストの各節毎、つまり学習項目毎に理解度を見るのは、学習内容と理解度との相関を見ることによって、より詳細な分析ができると期待したからである。各節の学習項目については、Appendix-1 に示してある。ここに注目すべきは、§3 と §4 である。前者では、2 次方程式の解や、最大公約数を求める問題等の数学的な内容を持った題材を多く扱っているのに対し、後者で扱っているのはグラフィックを用いた作図であり、ここには数学的な内容はほとんど含まれていない。従って、§3 における理解度には数学的な知識が反映されるが、§4 においては、数学的な色彩が弱まり、アルゴリズムの理解度がプログラミングの理解度としてより鮮明に反映されると期待される。そこで図を見てみると、全学習項目において、文系学生よりも理系学生の理解度の方が一様に高くなっていることが分かる。では、この結果は、数学の習熟度がアルゴリズムの理解度のよい指標になっていることを意味しているのだろうか。

この結論を得るためには、文系学生の理解度が低くなっていることの原因が他にある可能性を検討してみなければならない。一般に、理解できないという学生の中には、あまり熱心に実習に取り組んでいない学生が多いという傾向がある。その場合、理解度が低いことの原因は、アルゴリズムの理解度よりもむしろ学習態度に帰着されるべきであろう。もちろん、アルゴリズムの理解度の低さが学習態度に反映されている可能性は排除できないが、少なくとも、理系、文系グループでの理解度の差異を論ずる場合、学習態度という要因をあるレベルに揃えた上で比較を行った方がよいことは確かである。そこで、学習態度が良好なグループを選別して、その集団について理解度を見てみたのが図2である。

ここに、学習態度良好グループとは、実習への出席率が8割以上で事前にテキストに目を通して実習に臨んでおり、さらに1週間に30分以上、つまり一度は実習時間外に情報処理実習室で自主学習を行ったという条件を満たすグループである。これらは、実習に当たって我々が学生に奨励した学習態度である。図より、§3については、文系学生の理解度が下がり、理系、文系学生間の理解度の差が広がっているのに対し、§4になると両者の差が縮まるといふ顕著な傾向が見てとれる。このことは、数学的な理解力の差を除けば、アルゴリズム

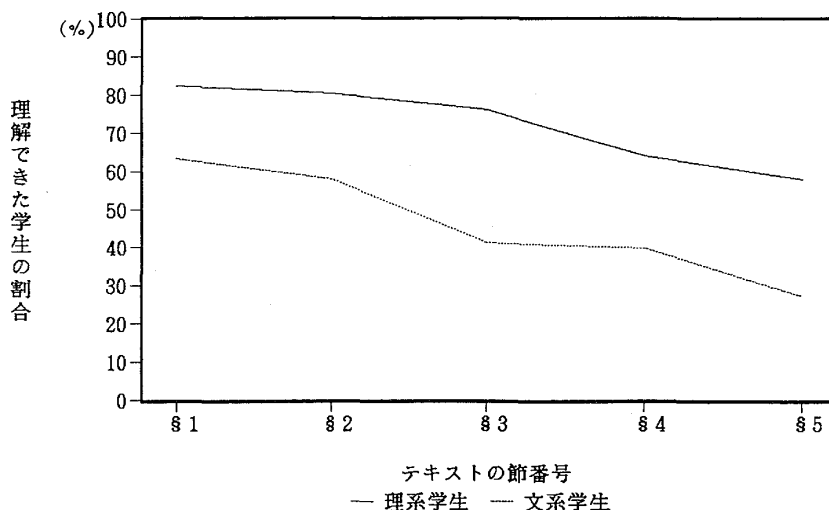


図1 BASIC 理解度

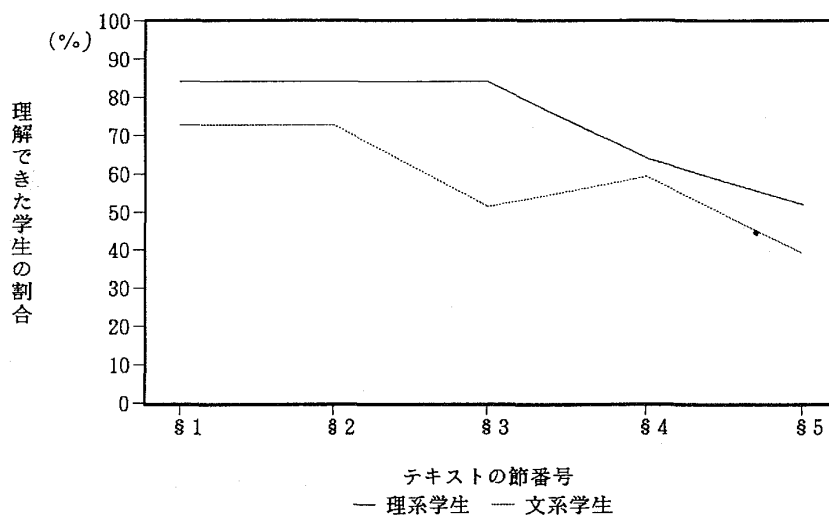


図2 BASIC 理解度 (学習態度良好グループ)

の理解度自体は、理系、文系でほとんど差がないことを示唆している。またこの結果は、プログラミングの理解度には、問題内容に対する理解度が大きく反映されている可能性があることも示している。従って、アルゴリズムの理解能力を分析するためには、まず問題内容の理解に支障がないように配慮する必要がある。このことを教訓として、以下問題内容の理解度に支障がないように工夫した上で、さらにアルゴリズム理解能力の分析を進めることにする。

§ 3. アルゴリズム理解度チェック テストの分析

前節で述べた指摘を検証し、さらにアルゴリズム理解能力の分析を進めるために、今年度、FORTRAN を専攻した 2 年次の学生に対して、前期の中間に図 3 に示すアルゴリズム理解度チェックテストを行った。

§ 2 で得られた教訓を生かして、問題としては、特に数学的な知識を必要しない題材を選び、問題自体の理解に支障がないように配慮した。実施方法としては、まず、問題 (1) (テスト 1) を課し、その後ヒントとして画面設計の仕方を与えて改めて問題 (2) (テスト 2) として課す、という形態を採った。この解答例は、Appendix-2 に示してある。

図 4 に、テスト 1 の成績結果を理系、文系学生各々について示してある。ここに、成績は A から D まで 4 ランクに分類されており、B までが、アルゴリズムの基本構造は理解しているという意味で、合格レベルに達しているグループ、C 以下は、理解が不十分なグループである。そこで (A+B) グループ、即ちアルゴリズムを理解しているグループが全体に占める割合に注目すると、予想通り、理系、文系とでほとんど差がない (いずれも約 60%) ことが分かる。テスト 2 についてもこの傾向は全く同様である。この結果は、問題内容が理解できれば、アルゴリズムの理解度は両者で大差がない、という § 2 で述べた指摘を支持してい

図 3 アルゴリズム理解度チェックテスト

(1) 【前判定反復型問題】
150 円の入場切符を売る自動販売機の模型を作るプログラムの PAD 図を作成せよ。
入れるお金は入力データで与えよ。
実際にこの模型にお金を入れたつもりでうまく機能するよう出力するメッセージや値を考えよ。

(2) (ヒント)
【画面設計】

```

お金を入れて下さい=20
-----
お金がたりません。追加して下さい=100
お金がたりません。追加して下さい=50
-----
あなたの入れた金額の合計は170円です
おつりは 20円です
切符をお取り下さい
    
```

図 4 テスト 1 成績 (理系 vs 文系)

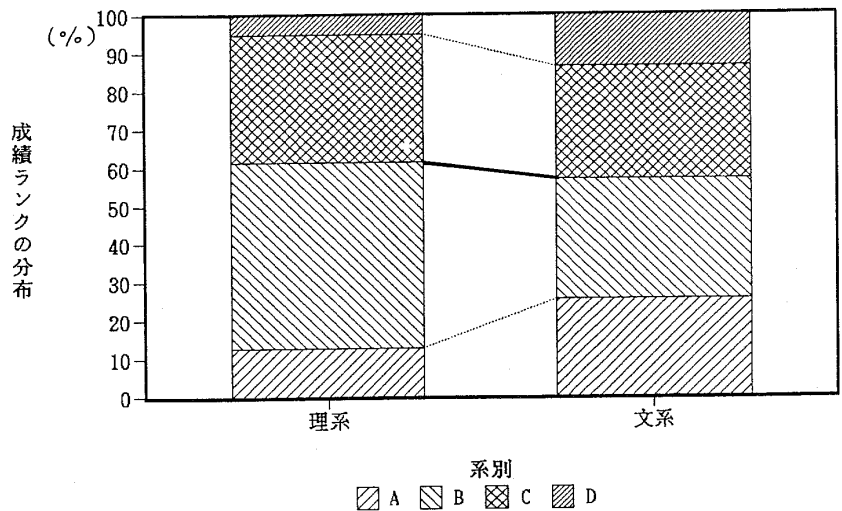
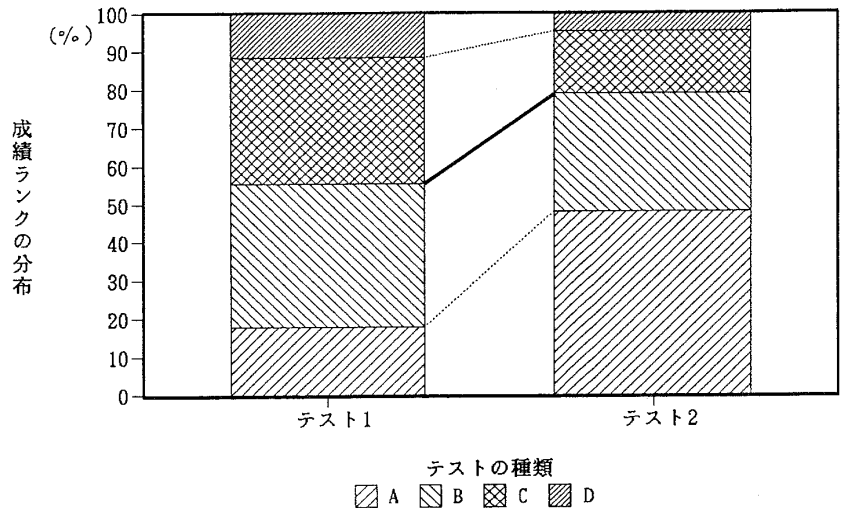


図 5 テスト 1 vs テスト 2



る。

次に、テスト1とテスト2の成績結果を比較した図5に注目しよう。この結果を理解するためには、テストが行われた状況について今少し説明する必要がある。テスト1の場合は、150円の切符を販売する自動販売機という以外には何も指示しておらず、どのような仕様にするかは学生の判断に委ねてある。そのため、おつりを出す場合に、硬貨の数を最小にするように500円、100円……をどの組み合わせで出したらよいか、まで考えていた学生もいれば、150円丁度を入れる場合のみ、つまりおつりのことは何も考えていない学生もいた。また出力するメッセージをどのような表現にすればよいかに頭を悩ます学生もいた、というように学生の対応は千差万別であった。このように考察範囲の自由度が広いということは、学生にとってかなり苦痛であったと見え、図5の結果を見ると、アルゴリズムの基本構造が単純であるにもかかわらず合格ラインに達したのは約55%に過ぎない（この数字が理系、文系それぞれのグループの値よりも小さくなっているのは、全体にはそれ以外の学生も1割弱含まれているためである）。次に、少し時間をおいて、画面設計を与え、つまり考察する範囲を具体的に設定した上でアルゴリズムを考えさせ、これをテスト2とした。すると、図5に見るように8割の学生が合格点に達することができた。

以上の経過を考慮すると、テスト1の段階で、アルゴリズムを書けなかった学生の多くは、操作手順の論理構造が分からなかったと

いうよりも、どこまで仕様を工夫すればよいのかという考察範囲の限定化ができなかったものと考えられる。今、問題に応じて適当な仕様を設定できる能力を、問題定式化能力と呼ぶことにしよう。すると上の結果は、アルゴリズムを作成するには、§2で指摘した問題内容の理解能力（今の場合、多くの学生がクリアーしていると考えられる）に加えて問題の定式化能力が必要とされることを意味している。このことはまた、問題の定式化がなされれば、多くの学生がアルゴリズムを作成できる、という可能性を示唆している。

ここまでの考察から、アルゴリズムを考案するという作業は、少なくとも、[問題内容の理解→問題の定式化→処理手順の論理的組み立て]という過程から成っていると考えられる。一般にアルゴリズムの理解能力というと、論理的な思考能力のみに注目しがちであるが、以上の分析結果は、初等的なプログラミングの場合、問題の理解能力および定式化能力が相対的により重要であることを示唆している。

さて、次に問題の定式化能力について、もう少し分析を進めてみよう。問題の定式化を行うには、その要求内容に応じた柔軟な発想が要求される。その柔軟な発想を阻害しているのは、すぐに暗記に頼ろうとする今の学生の気質にあるのではないかと、我々は分析している。そこで、暗記に頼らず柔軟に問題に対処できる能力を、問題の定式化能力の一つの指標と考え、この能力を調べるテストを行ってみた。以下これを仮に思考能力テスト

②
$$\begin{array}{|c|} \hline \square \\ \hline \square \\ \hline \end{array} + \square = \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} + \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array} + \square = \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \end{array} + \begin{array}{|c|c|} \hline \square & \square \\ \hline \square & \square \\ \hline \end{array}$$

という規則が成り立つとき、下式の空欄に入る図形を解答欄に記入して下さい。

$$\begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \square & \square & \square \\ \hline \square & & \\ \hline \end{array} + \square = \square + \square + \square$$

図6 思考能力テスト

と呼ぶことにする。図6はその思考能力テストの一部である。

これは、与えられた図形の合成規則を読み取り、それを一般の場合に適用するという問題であり、約65%の学生がこれに正解することができた。もし、このテスト結果が、問題の定式化能力の指標になり得るとすれば、アルゴリズム理解度テストの成績との間に強い相関があるはずである。そこで、思考能力テストに正しく答えた(正答)グループと、答えが誤っていた(誤答)グループそれぞれについて、アルゴリズム理解度チェックテストの成績結果を調べてみたのが、図7である。

(a)より、テスト1については、正答グループの方がアルゴリズム理解度チェックテストの成績が良い、つまり合格ラインに達した(A+B)グループの割合がかなり高くなっていることが分かる。そして両者の成績の差はテスト2では小さくなる。これは、テスト2ではヒントが与えられており、定式化能力の比重が小さくなっているためであると考えられる。これらの結果は、思考能力テストが問題定式化能力の一つの指標になっていることを示唆している。

§ 4. 前期テストの分析

FORTTRANの前期末の試験として、図8に示すアルゴリズムテストを行った。そこで、

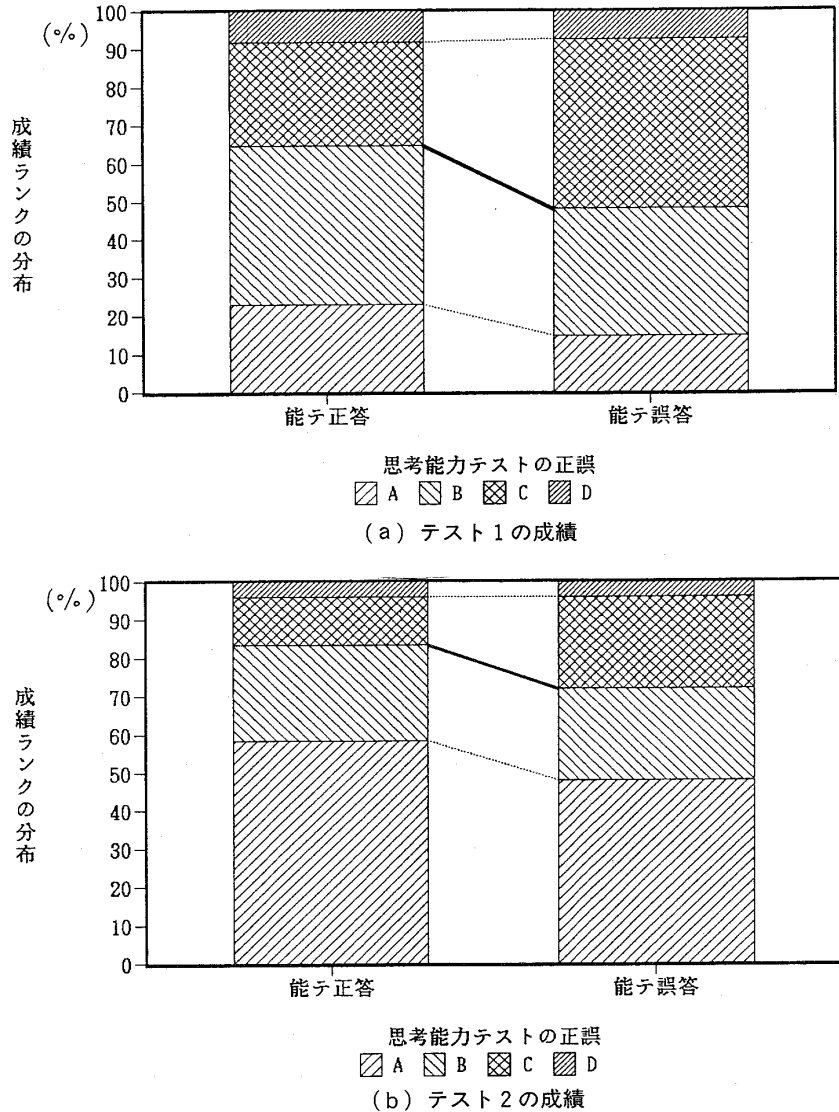


図7 アルゴリズム理解度テストと思考能力テストとの相関

前節までの分析結果から得られた、アルゴリズム理解能力に対する我々の考え方を、この成績結果を用いて検証してみよう。なおテストの解答例は Appendix-3 に示してある。

§3で示したアルゴリズム理解度チェックテストと同様に、問題としては、数学的な知識を特に必要としないものを用意し、問題自体の理解でつまづくことがないように配慮した。従って、§3の指摘が正しければ、その成績結果は、問題の定式化能力と強い相関があると期待される。その結果を図9に示す。

今、アルゴリズムの理解度が合格ラインに達している(A+B)グループの占める割合

1992年度前期 プログラミング言語Iテスト 7月10日(金)3講時目

学籍番号 _____ 氏名 _____

1) と 2) および別紙のアンケートに答えよ。

1) ある日の気温と湿度 (%) を入力して、不快指数を求め、不快指数の値によって下記通りメッセージを表示するプログラムのPAD図を書け。入力を促すメッセージも表示せよ。分岐(選択)箱内は日本語で記述してよい。気温、湿度、不快指数とも小数点がつく。不快指数 = $0.81 \times \text{気温} + (0.99 \times \text{気温} - 14.3) \times \text{湿度} \times 0.01 + 46.3$ である。

不快指数	表示メッセージ
80未満	'汗はでない'
80以上~85未満	'暑くてあせが出る'
85以上	'暑くてたまらない'

2) ある期間について不快指数を調べたい。最初に、調べる日数を入力し、入力した日数だけ1)を繰り返し、最後に「暑くてたまらない」日何日あったかを表示するようにPAD図を書き直せ。新たに加える変数名と型の定義も与えよ。

1) 変数名 _____ 型 _____ 何を表す変数名か _____ 2) 変数名 _____ 型 _____ 何を表す変数名か _____

図8 前期テスト

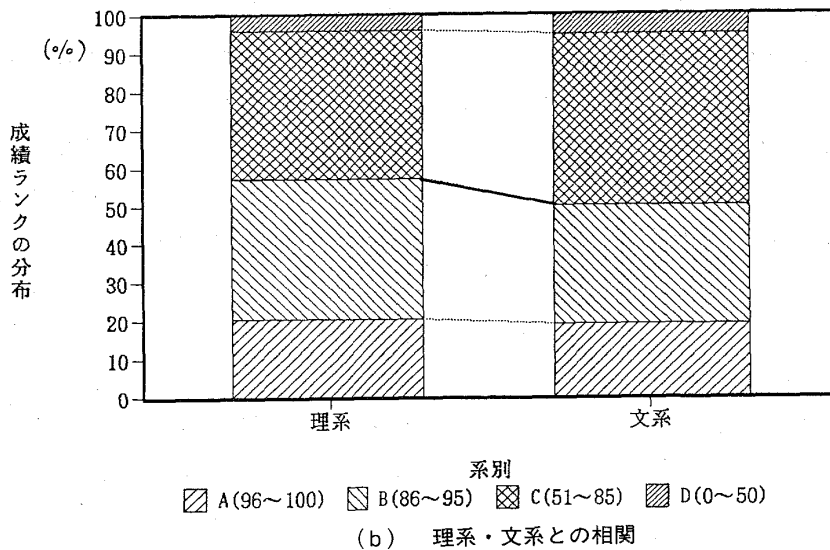
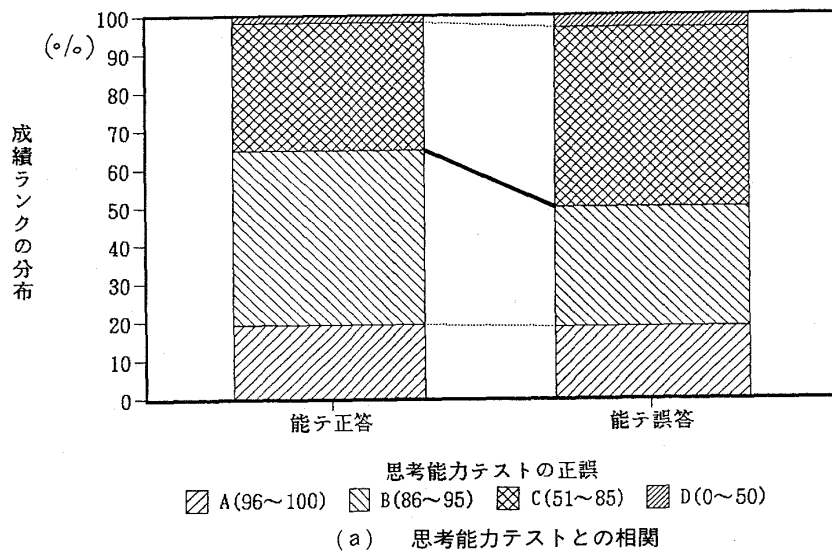


図9 前期テストの成績

に注目する。すると、(a)より、問題定式化能力の指標と考えられる思考能力テストの成績と前期テストの成績にはやはり強い相関があることが分かる。確認のために(b)で理系、文系グループ毎の成績を比較してみたが、これまでの考察から予想されるように両者の差はほとんどない。これらの結果は、前節までの我々の指摘を指示している。

さて、今、問題の理解に支障がないとすれば、§3で指摘したように、前期テストの成績は、問題定式化能力と論理的思考能力との総合的な結果であると考えられる。この点をもう少し突っ込んで分析してみよう。問題定式化能力については、前節に示した思考能力テストがその指標になりそうである。そこで、論理的思考能力についての何らかの指標があれば、アルゴリズム理解能力に対する

我々の理解をさらに深めることができる。この点については多くの議論を費やさなければならないが、現時点では、我々は、論述能力がその有力な指標になり得るのではないかと考えている。なぜなら、論理的な文章を書くためには、論理的な思考能力が不可欠であるからである。そこで、論理的思考能力の指標として、本学部の伊藤助教授が実施した「情報社会学」の論述テストの成績に注目してみた。図10が「情報社会学」の試験問題である。

- I. 言語情報の特質を、シニフィアン/シニフィエ、コネクション/デネーション、コンテキスト/話し手、という3つの対概念を用いて説明しなさい。
- II. 講義では、人間の情報処理過程を3つのレベルから把握しましたが、この3つのレベルをくわしく説明しなさい。

図10 情報社会学試験問題

この成績は、文章の論理展開力に応じてAからDまでの4段階に分類されている。この内、(A+B)グループは、ほぼ論理的な文章を書いているグループである。従って、この(A+B)グループを論理的な思考を展開できるグループであると考え、この論述テストの成績と前期アルゴリズムテストとの間に相関はあるであろうか。その結果が図11である。

図より、論述能力に優れているグループ(情社(A, B))は、やはりアルゴリズムテストの成績が良く、両者には強い相関があることが分かる。しかし、情社(A, B)グループは熱心に講義を聴いたグループであるとも考えられ、従って、このデータのみでは、真面目に勉強する学生の成績が良いということを示しているに過ぎない、とも考えられる。

そこで、最後に図11のデータを前節の思考能力テストと絡めて分析してみよう。これまでの分析からすると、問題の定式化能力がな

ければ、論理的思考能力があってもアルゴリズムを作成できないことになる。そこで、思考能力テストについての、正答、誤答それぞれのグループについて情報社会学テストと前期(アルゴリズム)テストの相関を取って見たのが図12である。

(a)より、思考能力テストが誤答である場合は、即ち問題の定式化ができていない場合は、情社(A, B)と情社(C, D)とでアルゴリズムテストの成績に差はない。つまり、問題の定式化ができていなければ、論述能力が生かされない事が示唆される。一方、(b)のように思考能力テストに正答し、つまり問題の定式化ができたと考えられる場合には、論述能力に優れたグループの方が顕著にアルゴリズムテストの成績が良くなっている。これらの結果は、問題内容が理解された後、問題の定式化と論理的思考の両方が備わって始めてアルゴリズムを作成できる、というこれまでの我々の分析と矛盾しない。

§5. ま と め

本報告では、学生の理解度調査やアルゴリズムテストの成績結果を基に、どのような要因によってアルゴリズムの理解度が決まっているのか、についての分析を包括的に行った。

分析結果より、我々は、アルゴリズムを作成する作業は、少なくとも[問題の理解→問題の定式化→処理手順の論理的組み立て]、という過程を経てなされていると考えられることを指摘した。さらに分析結果は、初等的なアルゴリズムの作成の場合、問題の理解と定式化の過程が相対的に重要であり、ここでのつまずきがアルゴリズムの理解を阻害していることを示唆している。一般に学生がアルゴリズムを理解できないという場合、まず問題の理解能力、問題の定式化能力そして論理的思考能力のどこに問題があるのかを分析した上で、その指導にあたる必要があるであろう。

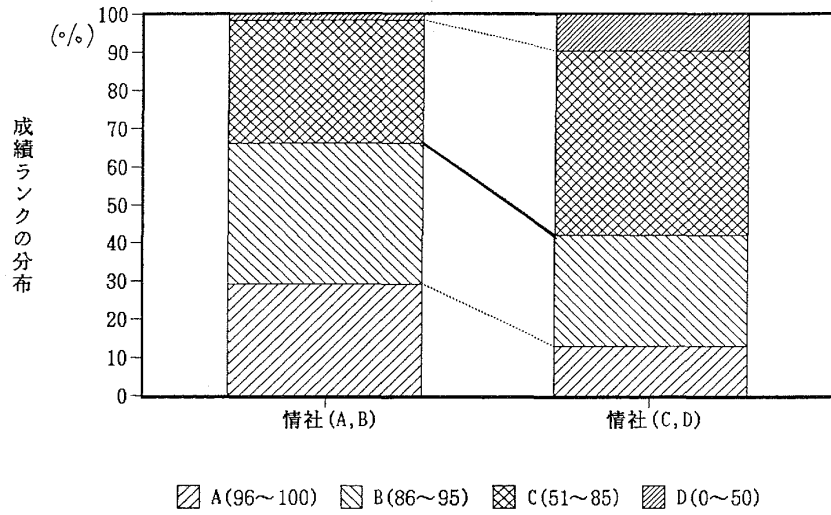
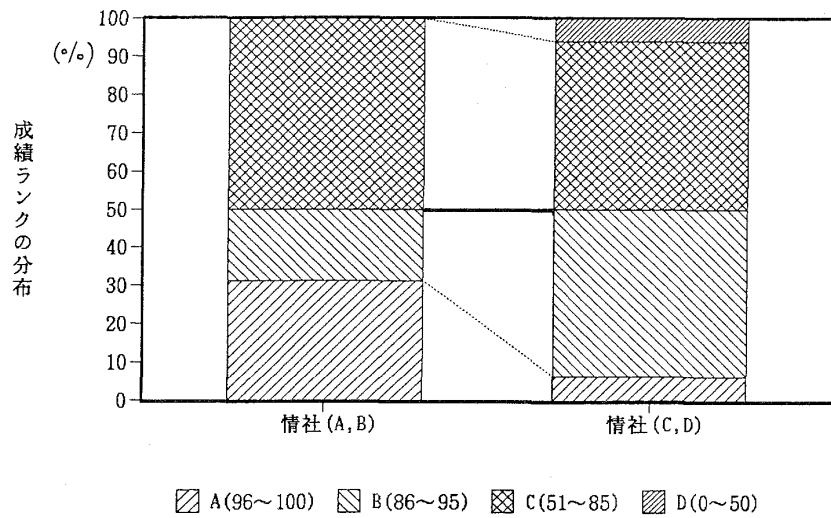
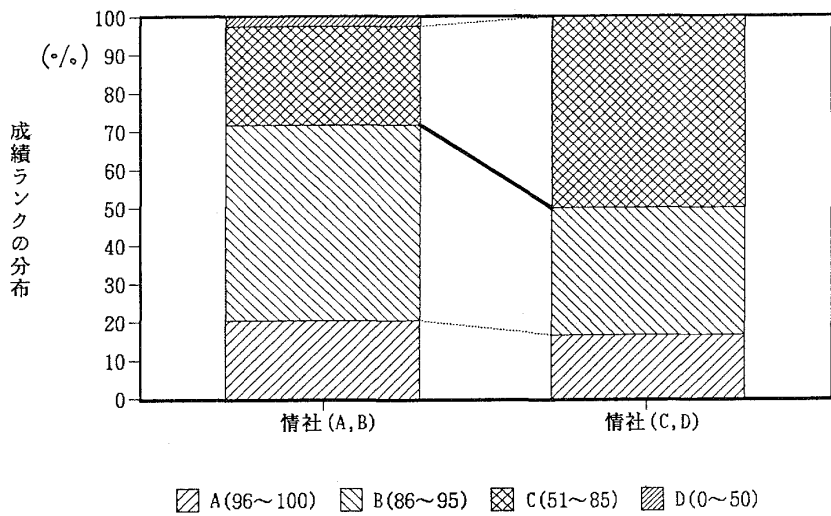


図11 前期テストと情報社会学テストとの相関



(a) 思考能力テスト誤答グループ



(b) 思考能力テスト正答グループ

図12 前期テストと情報社会学テストとの相関

今回の分析はまだ preliminary なものであるが、今後さらに分析を詰め、検証を進めて行きたいと考えている。

謝 辞

今回の報告のテーマについて貴重な助言を与えてくれた田中一教授、おおび分析に当たって「情報社会学」の成績結果を快く提供し、協力を惜しまなかった伊藤守助教授にここに深謝したい。

文 献

- (1) 森田 彦, 新國 三千代: 社会情報学部における情報処理基礎教育, 社会情報, 札幌学院大学社会情報学部紀要, Vol.1, No.2, pp. 35-47 (1992).

Appendix-1 「BASIC」学習項目

III. BASIC

§ 1. プログラム作成・実行の手順

- ① フロッピィディスクの初期化
- ② BASICの起動・終了
- ③ プログラムの入力・実行
- ④ プログラムの保存・呼び出し
- ⑤ プログラムの修正方法
- ⑥ プログラムの印刷

§ 2. BASICの基本命令

- ① PRINT文
- ② 代入文と変数
- ③ INPUT文
- ④ FOR~NEXT文
- ⑤ IF文(条件判断)
- ⑥ GOTO文

【練習問題I】—「IF 条件式 GOTO 行番号」の使用練習

- ⑦ エディットモードの使い方

§ 3. 基本命令の応用

- ① 2次方程式の解

【練習問題II】—3つ以上の選択肢がある場合のIF文の練習

- ② 「九九」の表

【練習問題III】—PRINT USING文を用いて数表を作る練習

- ③ 関数表(組み込み関数)

【練習問題IV】—組み込み関数の使用練習

- ④ 並べ替え(ソート)

【練習問題V】—3数のソートプログラムの作成

【練習問題VI】—4数のソートプログラムの作成

- ⑤ 最大公約数

【練習問題VII】—2数の最小公倍数を求めるプログラムの作成

§ 4. グラフィック機能の基本

- ① 画面の座標
- ② PSET文(点を打つ)
- ③ LINE文(線を引く)
- ④ CIRCLE文(円を描く)
- ⑤ PAINT文(色をぬる)
- ⑥ アニメーション

§ 5. 一歩進んだ BASIC 命令

- ① 配列

- ② WHILE~WEND文

【練習問題VIII】—WHILE~WEND文の理解度確認

- ③ READ文とDATA文

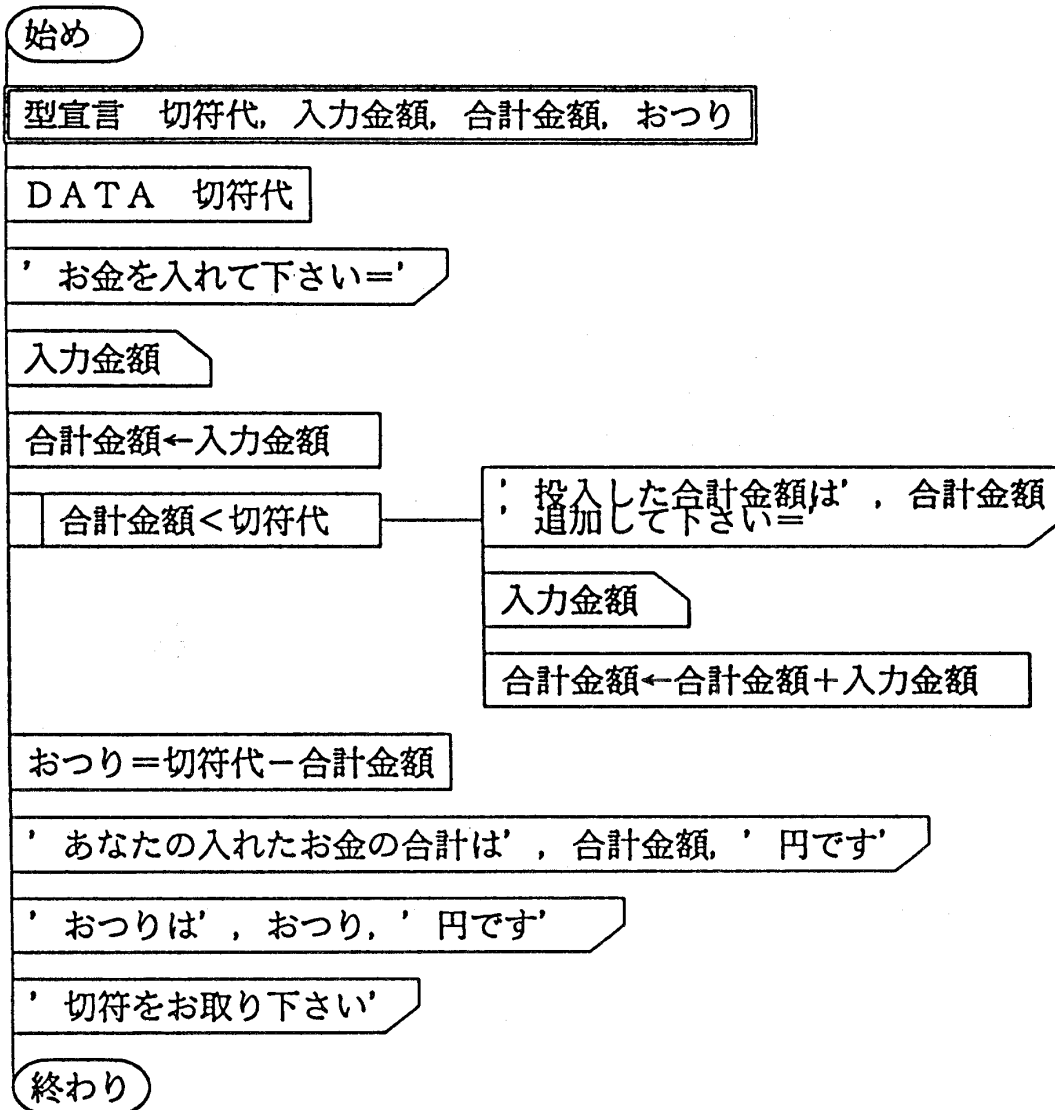
- ④ サブルーチン

【練習問題IX】—配列の使用練習

【練習問題X】—N個のデータのソートプログラムを配列を用いて作成

Appendix-2 アルゴリズム理解度チェックテスト・解答

変数の定義	変数名	型
切符代	切符代 (KIPPU)	INTEGER
入力する金額を入れる変数	入力金額 (NYUKIN)	INTEGER
合計した金額を入れる変数	合計金額 (GOUKEI)	INTEGER
おつりを入れる変数	おつり (OTURI)	INTEGER



データの与え方 (例)

- 1) 最初に150円入れる
- 2) 最初に150円より大きい金額を入れる
- 3) 最初に150円より小さい金額を入れて、次に適宜おつりがないように追加する
- 4) 3) でおつりがでるような追加のしかたをする

Appendix-3 前期テスト・解答例

1) 変数名 型 何を表す変数か

K	REAL	気温
S	REAL	温度
F	REAL	不快指数

2) 変数名 型 何を表す変数か

N	INTEGER	調べる日数
I	INTEGER	制御変数
A	INTEGER	暑くてたまらない日を求める変数

