

「アルゴリズム」教科(C言語分)の概要と 例プログラムでのデータの表示

北九州職業能力開発短期大学校 関谷 順太

A Report of "Algorithms" Syllabus with C and Data Display by Example Programs

Jyunta SEKIYA

要約

当短大の情報システム系1年生の「アルゴリズム」の教科（C言語による10月までの分）の概要と教材の工夫として例題プログラムにおけるデータの表示を報告する。

情報システム系のカリキュラムの検討を「あり方検討委員会」の答申により行って、平成4年度から「アルゴリズム」教科をI期から導入を含めて行うことにした。10月までの学習範囲としては、基本的な算法とその評価とした。

処理手順の表現法としては、流れ図、日本語での文章表現、C言語でのプログラミングとした。

テストデータがメモリで変化する様子の表示を例題プログラムに追加し、処理の確認が自動的に行えるようにプログラム教材を準備した。これにより、コンピュータを利用して、プログラムの動作の把握が容易に行えるようになった。

1 はじめに

情報システム系の1年生（平成4年度48名）の「アルゴリズム」と「構造化プログラミング実習」教科の目的は、基本的な処理手順の理解と表現、評価ができることである。具体的には、アルゴリズム(処理手順)の表現物である流れ図や日本語での文章(処理手順書)を読み、構成し、書くことと、計算量と個別のアルゴリズムの特徴の評価ができることをめざす。さらに、このアルゴリズムを実行可能なプログラムとして計算機を使って作成・テストし、実行時間の測定などができ、評価できることである。

この前に学習していること（既習事項）について説明する。図1のシラバス(授業計画表)を参照のこと。入学後すぐに、導入教育・コンピュータ入門としてキー操作、エディタ、MS-DOS、日本語ワープロについての実習を、「計算機命令実習」などの4教科合わせて12時間ほど行っている。次にC言語でのプログラミングの基礎的な学習は、「ソフトウェア工学」との3教科一緒で導入教育に引き続き既に11回ほど行っている。

この授業で使用するテキストは「C—データ構造と

プログラム」⁽¹⁾であり、処理手順はC言語での例題プログラムによっている。そこには、考え方とともに、具体的なデータ例での説明が付いている。例題のCソース・プログラムにコメント等を付加したものは、予め著者がハードディスクに入力しており、学生は各自のフロッピーにコピーして使うのを原則とした。(演習で作るプログラムは除く。)

授業の進め方は、始めに学習内容と処理手順の概要を説明し、その後は例題プログラムの機能と使い方の説明をして、実際のデータを準備して、それを使っての操作を行い、データの動きを確認していく方法をとった。そして、レポートとしての提出物をプログラム仕様(入力、処理、出力)、関数毎の処理手順書、考察をエディタで入力したもの、実行結果のリスト、流れ図(手書き)とした。

この講義と実習を組み合わせたやり方によって、実習のないクラスでの授業での単なる文章やプログラムリストでの想像のみでなく、テストデータでのコンピュータ実験ができた。そしてVで述べるが、コンピュータによりデータ処理の確認が出来るような例題プログラムを用意していた。時間的にはコンピュータ

<p>目的：計算機を用いて、情報を処理するための、基本的なデータ構造とその処理方法（アルゴリズム）について学ぶ。I期は、配列とソート、組み合わせ問題までとする。各種のデータ構造とアルゴリズムの得失（時間計算量、領域計算量）を検討する。 目標：アルゴリズムの言語表現と、C言語による、線形や2分探索・再帰・配列・ソートの各種のプログラムが理解できること。 評価：定期試験、小テスト、レポート、出席の総合評価とする。</p>			
<p>テキスト（1, 2, 3, 4）と参考書 1. Leendert Ammeraal 著/小山裕徳訳, Cデータ構造とプログラム, オーム社, 1990 2. 河西朝雄著, C言語, ナツメ社, 1990 3. 河西朝雄著, MS-DOS, ナツメ社, 1990 4. 馬場勇著, プログラミングの基礎知識, 技術評論社, 5. 石畑 清著, アルゴリズムとデータ構造, 若波書店, 1989</p>			
<p>実績と反省 1)期の前半を、関連教科と一連の授業として、導入教育と言語Cの学習とした。（導入教育については、別紙を参照のこと。） C言語は、テキストの例題と演習問題を実際に入力、実行した。流れ図はテンプレートを使ってリストに追加した。 しかしながら、豊富な演算子やポインタ、標準関数、ファイル操作については、不十分であった。 2)期の後半を、C言語でのアルゴリズムとデータ構造のテキストの前半を学習した。 テキストの例題の動作確認によって学習した。流れ図のほかに処理手順を文章で書いたが、制御構造の理解が不十分であった。</p>			
No.(月日)	内 容	詳 細 内 容	実 績
1回 2回	導入教育(6回) 計算機入門 エディタ,	VTR, 入門ディスクとタッチタイプの紹介, MS-DOS 日本語ワープロ, 導入教育のまとめ(レポート)	別紙参照
3回(1)	1章 Cの基本的なきまり	1.1 C言語とは 1.2 簡単なプログラム 1.3 単語	1)5.08 1章
4回(2)	2章 コンピュータとの入出力	1.4 データ型と変数の型宣言 1.5 プログラミング書式 2.1 scanf, printfの概要 2.2 printf 2.3 出力桁数の指定	2)5.12 2.1 ~2.4 3)5.15 2.5 ~2.7
5回(2)	3章 制御構造	2.4 scanf 2.5 文字とアスキー・コード 2.6 1文字単位の入出力 3.1 if文 3.2 if else文 3.3 複雑な条件式	4)5.19 3.1 ~3.3 5)5.22 3.4 ~3.5 演習
6回(2)	4章 配列	3.4 for文 3.5 while文 4.1 配列 4.2 配列の初期化 4.3 データ代入 4.4 2次元配列	6)5.26 4章
7回(2)	5章 ポインタ 6章 関数	5.1 ポインタとは 5.2 文字列 5.3 配列 5.7 アドレス計算 6.1 関数とは 6.2 局所的変数 6.3 値による呼び出し	7)5.29 5.1 ~5.3 8)6.2 5.4 ~6.1
8回(2) 6月12日	12章 標準ライブラリ 13章 ファイル操作	6.4 参照による呼び出し 6.5 配列を渡す 6.8 関数プロトタイプ 11.1 プリプロセッサ 11.1 引き数を持たないマクロ 12.1 ANSI関数 13.1 ファイル処理の概要 13.2 ファイル処理関数	9)6.5 6.1 ~6.4 10)6.9 6.5 ~演習(ハノイの塔) 11)6.12 まとめ
9回 6月16日	1章 プログラミング・スタイル, 繰り返しと再帰	1.1 時間計算量(time complexity) 1.2/3 番兵を使った線形探索・比較 1.4 広域の変数と副作用	d13comp.c d1e4.c
10回 6月23日		1.5 再帰とは 1.6 再帰の除去 1.7 最大公約数を求めるユークリッドのアルゴリズム	d014.c(再帰) d016.c d024.c, d1e1_1.c, 2.c
11回 6月30日		1.8 多項式の計算—ホーナーの方法 1.9 基数の変換 1.10 指数が整数の中乗	d025.c, d027.c d032.c, d1e11.c
12回 7月7日	2章 配列とファイルの操作	2.1 直接ソート 2.2 クイック・ソート	d044.c, d046.c d049.c
13回 ①② 7月23日		演習：前回の復習として、ポインタを配列表現に変える。	d2e2.c
14回 ③④ 7月23日		2.3 可変長文字列のソート 2.4 ファイルのソート	d056.c (5) d060.c, d062.c, d065.c
15回 ①② 7月24日		2.5 2分探索 2.6 ハッシュ法	d072.c d076.c
16回 ③ 7月24日		演習問題(素因数分解)	d2e8.c (7)
17回 8月25日	3章 組み合わせ問題	3.1 多重度が変数のループ	d085.c, d087.c d3e2.c
18回 9月1日		3.2 順列 3.3 組み合わせ	d091.c, d3e2.c d095.c, d3e4.c(チーム)
19回 9月8日		3.4 ナップザック問題 3.5 ダイナミックプログラミング	d098.c d102.c
20回 9月22日	定期試験	1. 良いプログラムの条件 2. フィボナッチ関数の比較 3. cosI関数の証明 (記述式) 4. ソート法(直接, クイック)の比較 5. 2分探索 6. ハッシュ法	$\mu=55, \sigma=11$ で23名には、レポートを要求した。

図1 「アルゴリズム」(C言語I期分)のシラバス

の操作実習が伴うために、予定よりも時間が掛かる面があった。しかし、それ以上に、自分からの能動的な働きによってしか、テスト・実行ができないこと、データ処理の確認が容易にできたこと、日本語での処理手順書の作成などにより、基本的な処理手順とそれらを実現したプログラム例題の理解が良くできたと思うので、報告する。

II 「アルゴリズム」教科の位置づけと学習範囲

1 「アルゴリズム」教科の位置づけ

職業訓練短期大学校におけるカリキュラムについては、「あり方検討委員会」の標準カリキュラム作成専門部会で検討され、それに従って当系・科でも見直しを行った。情報処理科での変更の主なものは、次の2点である。

- 1) 「計算機命令実習」が追加された。
- 2) 「デジタル工学」、「同実習」が追加された。

情報システム系でのカリキュラムの検討をして、図2の教科目の系統図⁽²⁾を作り、分担してシラバスの作成と検討会を持ち、授業の計画と実施をした。

なお、「アルゴリズム」教科（C言語によるもので10月まで）については、それまでの「データ構造」（II、III期で4単位で実習は付いていなかったもの）が、これに引き継いだことになった。開講時期がI期から変わったことにより、導入教育や、C言語の教育を始めに追加している。

これらの検討での参考資料としては、次のものも参照したので紹介する。

- 1) 情報処理学会での「大学等における情報処理教育検討委員会」の調査研究^{(3),(4),(5)}

これによると、アメリカでのACMカリキュラム'78等を参考にして、日本の大学での情報処理教育の検討がなされている。

- 2) 文部省と回り持ちの幹事大学による情報処理教育研究集会報告書^{(6),(7),(8)}

第1回を九州工業大学で昭和63年の10月に開催して以来第5回を1992年北海道大学で行っている。いろいろな大学や短大、高専での事例が紹介されている。

- 3) 初級情報処理技術者育成指針⁽⁹⁾

情報処理技術者試験を対象とする技術者養成のための標準のカリキュラムとして、初級のほかに中級、上級もある。主として、専修学校のカリキュラム等に使われている。

- 4) 幾つかの大学での情報処理教育の事例の紹介が、

情報処理学会誌にある。^{(10),(11),(12)}

2 「アルゴリズム」教科(C言語)での学習の範囲

学習時間や時期によるわけで、教科書によっても違おうし、標準的なものが有るわけではない。「あり方」での範囲を参考にして、使用するテキストは以前からのもの⁽¹⁾を使うことにして、図1のシラバスとした。なお、II期のはじめに16こま、続きを行っている（線形リスト、2分木、B-木、トライ、グラフ、インタプリタとコンパイラの基礎）。その後は、COBOLでのプログラミングに切り替わっている。

なお、この教科でのアルゴリズム概要に関する授業（第9回）だけでなく、その都度、学習するそれぞれのアルゴリズムの比較・評価を行っている。

III アルゴリズム（処理手順）の表現法

処理手順の図による表現法は、流れ図のほかに幾つかの方法がJISには参考表としてある。また、教科書などでは、日本語での表現、PASCALやC言語などのプログラミング言語での表現などがある。それらの表現法について概要を説明する。

1 流れ図

図記号が分かりやすく、現在の「JIS X0121 情報処理用流れ図等記号」⁽¹³⁾はループ端の記号を持ち構造化プログラミングに対応している。情報処理技術者試験の2種では、流れ図の問題が出題されている。

しかし、流れ図では記号の大きさに制限されて、枠内では短い表現となり、説明は右側にコメントとして分けて書くことが必要となる。

また、一般のテキスト・エディタでは流れ図の記号を書くことができない。

2 その他の記号表現図

「JIS X0128 プログラム構成要素及びその表記法」⁽¹⁴⁾の参考表で紹介してあるように、欧米各国や日本でも企業により、幾つかが使用されている。

当科でも、今までにPAD、YAC IIなどが一部の教科で使用されてきた。

これらの表現をコンピュータ上で（編集ソフトを使って）行えば、目的言語のソース・プログラムを生成するソフトがある。（PADETやYPSコンパイラ）

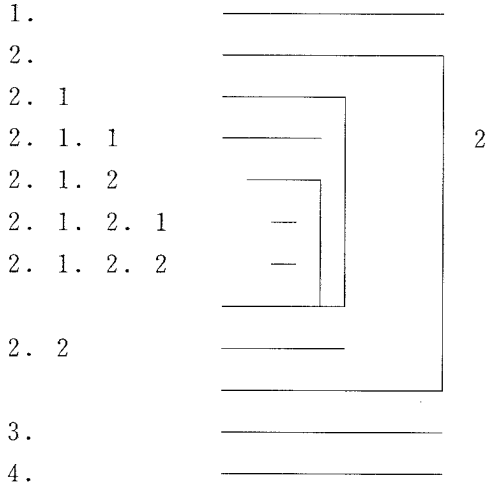
- 3) 日本語（自然言語）での文章表現（処理手順書）

情報システム系の系統図



図2 情報システム系の教科の系統図⁽²⁾

図を使わない表現として、制御範囲を明確にするために、見出し記号の付け方を次のように規定すると、処理の意味を説明して日本語で詳しく表現できる。この見出し記号の付け方は技術関係で多用される分類体系⁽¹⁵⁾である。



この処理手順書での要素は英数字であり、テキスト・エディタで編集できる。なお、テキストエディタは2つ以上の窓を上下に分けるなどして開けることにより、ソース・プログラムと対比しながら、これらの処理内容をコピー・入力・編集することができる。

4 プログラミング言語等での表現

処理手順の図を使わない表現としては、日本語と記号をもとにした簡潔な記法⁽¹⁶⁾や PASCAL⁽¹⁷⁾、C^{(1),(18)}などのプログラミング言語での表現が使われている。プログラミング言語で表現の場合、データの入出力などを行うメイン関数とそこでのモジュールの呼び出し等を追加して処理系に通せば、計算機でテストできる。実際にテストを行うには、テストデータの準備や結果の確認などが必要になるが、そして、それらのモジュールを実用にも可能である。

プログラミング言語で表現の場合、その言語についての文法の理解が必要となる。文法が分かれば、文章に較べて簡略した表現ができるので、見やすいとも言える。

情報システム系の1年生については系でのカリキュラム検討会で、図表現では一番基本的な流れ図を使うことに話し合って決めた。始めからいろいろな記号を使った表現図に戸惑うことをさけるためである。また、情報処理技術者試験の受験者のことも考えた。

この授業の中では、図記号での流れ図の他に、上記

の3処理手順書と、4のC言語での表現を組み合わせることにした。⁽¹⁹⁾

IV データの表示法

プログラムで使用するデータの表現の仕方として、変数表、データ構造図、領域図、レコードレイアウトや、プログラム言語での定義などがある。

単純なデータであれば、変数表や領域図やプログラム言語での定義で十分である。しかしポインタなどによる構造をもったデータの場合は、データ構造図により（動的な変化を含めて）2次的にデータの関係を表示することが必要になる。

「アルゴリズム」に関する教科書や参考書では、データの例をつけて、操作ごとに、どのようにデータ（の構造）が変化していくかを、図示している。図3は、2次のB木での追加の例である。

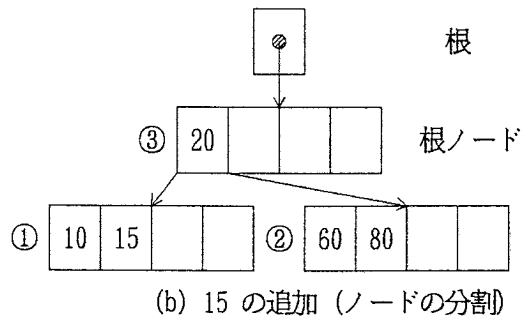
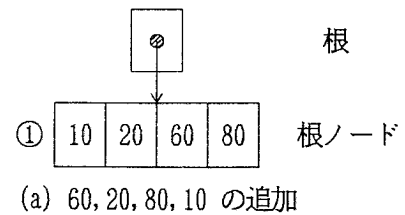


図3 2次のB木の成長（入力列とその結果）⁽¹⁾

授業での説明も、これらの図を白板に書いたり、テキストやプリントを参照したりして行っている。レポートでもそれらの手書きを要求していた。（実習の時間がなかった前のクラスの場合。）

しかしながら、実習でせっかくコンピュータを使うのだから、コンピュータでこれらを表示してやれば、プログラムでどんなことをしているのかを、わかりやすく見れるはずである。これを、次にのべる。

V プログラムでのデータ（構造）の表示と利用

処理手順を実際にプログラム化して、その動作をテストデータを順番に処理していく中で見るには、主なデータの変更（代入）ごとに、データ群をプログラムで構造的に表示してやれば、コンピュータでの動作が追跡できることになる。

これを普通デバッグ用のチェックプリントと呼んでいるが、それをもっと積極的な意味で分かりやすく使うことにする。つまり、主要なデータやその構造（関係）を表示するモジュール（関数）を追加し、そのモジュール呼び出しを、主要な処理毎に行うことで、デー

タの変化—つまりプログラムの一連の処理結果—の表示が実現できる。

データ型には、数値や文字とそれらのグループ（構造体）とそれぞれへのポインタ型がある。この表示では、それぞれの変数名とその値をグループごとに表示している。出力した画面（リスト）でポインタ変数の値をたどって実際に指し示す先とを学習者が結べば、データ構造図として完成することになる。（ポインタの線を引く作業をコンピュータでは行っていないので、半製品といえる。）

なお、レポート用のプリンタ出力は、コントロールPで行う方が、リダイレクションで行うよりもよい。理由は、入力データが同時に分かるからである。

```
<B木を作る幾つかの整数を入力してください。終わりは、/ 60 20 80 10 /
次数2のB木の {削除(D), 挿入(I), 探索(S)} 整数, 終了(Q), 表示(P):p

--- B-木のプリント root:  c
p:  c, p->key: 10 20 60 80, p->ptr:  0  0  0  0  0
---
次数2のB木の {削除(D), 挿入(I), 探索(S)} 整数, 終了(Q), 表示(P):i15
次数2のB木の {削除(D), 挿入(I), 探索(S)} 整数, 終了(Q), 表示(P):p

--- B-木のプリント root:  4c
      p: 2c, p->key: 60 80, p->ptr:  0  0  0
      p:  c, p->key: 10 15, p->ptr:  0  0  0
p: 4c, p->key: 20, p->ptr:  c  2c
---
```

図4 B木の実行例

```
printbt(NODE *p)/* B木のプリント関数(再帰) 90度 反時計方向に回転したもの*/
{
    int i;
    static indent =0;

    if(indent == 0) printf("\n--- B-木のプリント root:%4x\n", p);
    if (p != NULL) {
        indent += 6;
        for (i = p->cnt; i >= 0; i--) printbt(p->ptr[i]);/*子ノードのプリント */
                                                /* 再帰による */
        for (i=6; i< indent; i++) printf(" "); /* 字下げ */
        printf("p:%4x, p->key:", p); /* p:ノードの番地, データの見出し */
        for (i=1; i<= p->cnt; i++) printf(" %3d", p->key[i-1]); /* データ表示 */
        printf(", p->ptr:"); /* 子の番地の見出し */
        for (i=0; i<= p->cnt; i++) printf(" %4x", p->ptr[i]); /* ptr表示 */
        printf("\n");

        indent -= 6;
        if(indent == 0) printf("---\n"); /* B-木の構造図の終わり */
    }
}
```

図5 B木のプリント関数のソース・プログラムリスト

図4は、2次のB木の追加処理での実行例である。このデータは図3と対応しており、各ノードが1行となっている。図5はこのB木を表示する関数リストである。

かった。」との学生が多かった。じっくり考える時間が少なくて、消化不良を起こしたと考えられる。学習内容の絞りこみが必要であると思う。

3) 教材用プログラムにデータ表示の関数を追加し、

このVSLインタプリタ・プログラムは、入力ファイル(VSLプログラム)を読み込み、計算をして、出力します。

入力ファイル名は:d239.dat

VSLプログラムは、ファイルから { } を読み込みます。

```

{9 |      因子 9
- |      式 -
3 |      因子 3
+ |      式 +
1 |      因子 1
}
計算結果は: 7

{9 |      因子 9
- |      式 -
( |      因子 (
3 |      因子 3
+ |      式 +
1 |      因子 1
)}
計算結果は: 5

{2 |      因子 2
* |      項 *
3 |      因子 3
- |      式 -
4 |      因子 4
* |      項 *
5 |      因子 5
}
計算結果は: -14
    
```

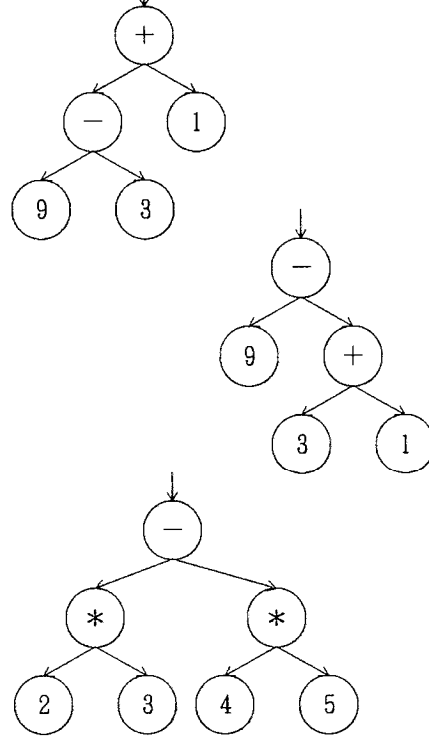


図6 整数演算インタプリタの実行例

図6は、整数値の算術演算を行うインタプリタの処理例である。関数が再帰的に呼ばれているのが分かる。木を反時計回りに90度回転して上下対称となっている。

VI まとめ

- 1) 北九州職業訓練短期大学の情報システム系での「アルゴリズム」教科（C言語による10月までの分）の平成4年度の概要を紹介した。
- 2) II期の11月から行ったCOBOLでのプログラミング作成との比較では、「C言語によるものが分かりにく

それを処理ごとに呼び出してデータ構造の変化をプリントした。これにより、処理途中での一連のデータ処理結果の確認ができ、計算機によるテスト状況把握を容易に行うことが出来た。

4) 今後の課題として、もっと分かりやすい標準的なデータの表示を工夫したい。

学生が興味を持って学習できるような授業とその中でひとりひとりに知識や技能が身に付いていくやり方を作っていきたいと思う。

最後に、指導・協力をして頂いている当系および当短大の職員の皆様に感謝します。

[参考文献]

- (1) Leendert Ammeraal 著：小山裕徳訳：C—データ構造とプログラム、オーム社、1990
- (2) 北九州職業訓練短期大学校：「平成4年度履修案内及び授業要目」平成4年、pp. 38
- (3) 野口正一・中森真理雄：大学等における情報処理教育の諸問題、情報処理、Vol. 31、No. 10、pp. 1373-1389
- (4) 野口正一他：大学等における情報系専門教育の改善への提言、情報処理、Vol. 32、No. 10、pp. 1079-1092
- (5) 情報処理学会大学等における情報処理教育検討委員会：大学等における情報処理教育のための調査研究報告書、情報処理学会、1991年3月
- (6) 九州工業大学情報工学部編：昭和63年度情報処理教育研究集会報告書、1989年3月
- (7) 東北大学情報処理教育センター編：平成元年度情報処理教育研究集会報告書、1990年3月
- (8) 京都大学情報処理教育センター編：平成2年度情報処理教育研究集会報告書、1991年
- (9) 情報処理技術者育成指針作成委員会：初級情報処理技術者育成指針、日本情報処理開発協会情報処理研修センター、1986年3月
- (10) 安村通晃・有澤誠・斎藤信男：コンピュータリテラシー教育の一事例、情報処理、Vol. 32、No. 12、pp. 1310-1317
- (11) 小谷善行・高橋延匡：情報処理専門教育の一事例、情報処理、Vol. 33、No. 2、pp. 161-168
- (12) 村岡洋一：情報学科カリキュラムの一例、情報処理、Vol. 33、No. 2、pp. 169-175
- (13) 日本工業規格 JIS X 0121-1986 情報処理用流れ図・プログラム網図・システム資源図記号、日本規格協会、1986
- (14) 日本工業規格 JIS X 0128-1988 プログラム構成要素及びその表記法、日本規格協会、1988
- (15) 篠田義明：コミュニケーション技術—実用的文章の書き方—中公新書807、中央公論社、1986、pp. 170
- (16) 土居範久・寛捷彦：プログラミングの考え方、岩波書店、1987
- (17) 石畑清：アルゴリズムとデータ構造、岩波書店、1989
- (18) 近藤嘉雪：Cプログラマのためのアルゴリズムとデータ構造、ソフトバンク、1992
- (19) 関谷順太：スクリーン・エディタによるプログラム資料作成、職業能力開発報文誌、第5巻、第1号、pp. 129-133