

夏休みの課題

山本昌志*

2005年7月14日

1 課題の概要

夏休み後、C言語を使った数値計算の学習を進める。ここでは、テイラー展開と線形代数をつかうことになる。そのために、それらの復習を夏休みの課題として課す。さらに、ソーティングのプログラムの作成をとおして、C言語のプログラムになれて欲しい。

また、余力のある者は、円周率の値を10万桁まで計算するプログラムを作成して欲しい。この円周率のプログラムは課題でないので、興味がある者はトライせよ。

2 数値計算の基礎

- [問 1] テイラー展開の公式を導き出せ。公式を書くだけではだめである。3~4年生の数学の教科書に書かれているはずである。
- [問 2] 問1の結果を利用して、関数 $f(x)$ を $x = a$ の周りでテイラー展開する式を示せ。
- [問 3] 問1の結果を利用して、関数 $f(x)$ を $x = 0$ の周りでテイラー展開する式を示せ。これをマクローリン展開と言う。
- [問 4] 問1の結果を利用して、関数 $f(x + \Delta x)$ を x の周りでテイラー展開する式を示せ。
- [問 5] 次の3つの関数を、 $x = 0$ の周りでテイラー展開、即ちマクローリン展開せよ。以下の3つの関係は、どうなっているか考察せよ。

$$e^x =$$

$$\sin x =$$

$$\cos x =$$

*独立行政法人 秋田工業高等専門学校 電気情報工学科

[問 6] 以下の連立方程式を行列とベクトルで表現せよ。

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1N}x_N &= b_1 \\a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2N}x_N &= b_2 \\a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3N}x_N &= b_3 \\&\vdots \\a_{M1}x_1 + a_{M2}x_2 + a_{M3}x_3 + \cdots + a_{MN}x_N &= b_M\end{aligned}$$

[問 7] 以下の連立方程式を行列とベクトルで表現せよ。

$$\begin{aligned}x_1 + 2x_2 - x_3 + 3x_4 &= 1 \\-2x_1 - 3x_2 - 5x_4 &= 2 \\2x_1 + 2x_2 + 3x_3 + 5x_4 &= 3 \\-x_1 + 3x_2 - 12x_3 &= 4\end{aligned}$$

[問 8] 前問の連立方程式の解と係数行列の逆行列を求めよ。解と逆行列は掃き出し法を使うこと。

3 C 言語の練習

ソートングとは、整列あるいは並び替えのことである¹。プログラミングでは、数値を大きい順、あるいは小さい順に並び替える技法のことを言う。数値の並び替えは非常に重要な技法で、実際のプログラムではいたるところで使われている。ソートングでもっと重要なことは、処理速度である。高速な処理を目指しているいろいろなアルゴリズムが考えられている。ここでは、ソートングの簡単なアルゴリズムを示し、C 言語のプログラムの学習を進める。

3.1 単純挿入法

ソートングの中でも、最も簡単な単純挿入法のプログラムを作成する。有名な C 言語の本「NUMERICAL RECIPES in C」によると、これは経験を積んだトランプ師が使う方法と同じであるということである。順序がバラバラのトランプを並び替える場合、

- まず、2 枚目のカードを拾い、1 枚目と順序関係が正しい位置に置く。
- 次に 3 枚目のカードを拾い、最初の 2 枚と順序関係の正しい位置にそれを挿入する。
- 同じことを繰り返す。即ち、 i 枚目のカードを拾い、最初の $i - 1$ 枚のカードの順序関係の正しい位置にそれを挿入する。

¹ここでの説明は、NUMERICAL RECIPES in C を参考にしている。

- 最後のカードを正しい位置に挿入したら、並び替えは完了である。

この単純挿入法を用いて、リスト 1 で作成された整数の配列 $a[0] \sim a[1023]$ を小さい順に並び替えよ。このリストの 19 行目以降に単純挿入法のプログラムを書く。ヒント 図 1 に単純挿入法のフローチャートを示す。

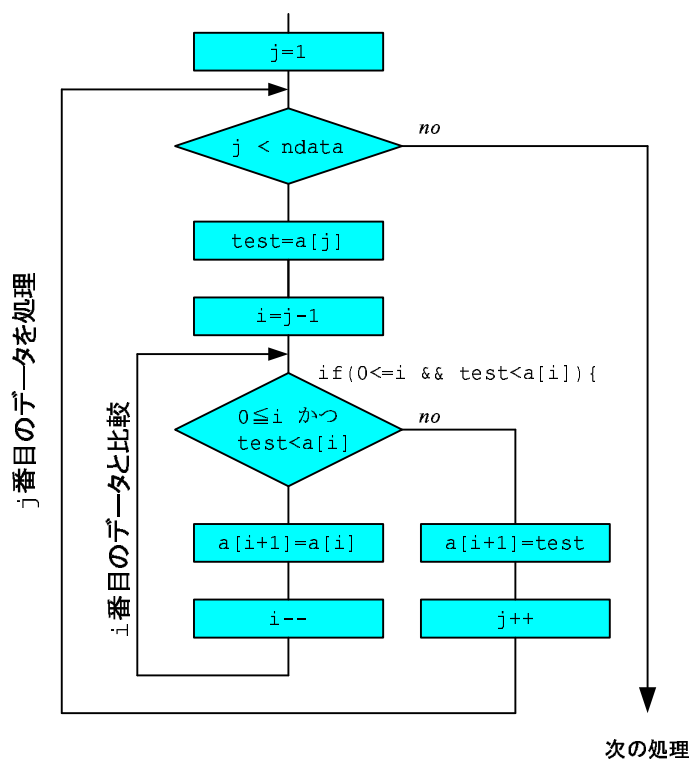


図 1: 単純挿入法のフローチャート。ndata はデータ数で、 $a[0] \sim a[ndata-1]$ の配列に格納されている整数を小さい順 (昇順) に並べる。

リスト 1: 単純挿入法のプログラム

```

1 #include <stdio.h>
2 #include <stdlib.h> /* 乱数発生のため */
3 #include <time.h> /* 時刻の関数を使うため */
4
5 int main(void){
6     int a[1024], i, j, ndata, test;
7
8     ndata=1024;
9
10    srand((unsigned int)time(NULL)); /* 起動毎に異なる乱数を発生させるため */
11
12    for(i=0; i<ndata; i++){
13        a[i]=rand(); /* 配列 a[i] に乱数の整数を設定 */
14    }
15

```

```

16
17
18 /* これ以降に単純ソートと昇順に並んだ出力のプログラムを書く */
19
20
21
22
23
24
25
26
27     return 0;
28 }

```

3.1.1 shellソート

Shell ソート²は 1959 年に D.L.Shell が考案した方法で、単純挿入法を改良したものとなっている。単純挿入法は、隣同士を比較しましたが、Shell ソートでは、大きな h 飛ばしで比較する。ソートに時間のかかる大きな数や小さな数は、一気に右や左に移動する。 h 飛ばしで比較すると、

$$\begin{aligned}
 a[1] &\leq a[h+1] \leq a[2*h+1] \leq a[3*h+1] \leq a[4*h+1] \leq a[5*h+1] \cdots \\
 a[2] &\leq a[h+2] \leq a[2*h+2] \leq a[3*h+2] \leq a[4*h+2] \leq a[5*h+2] \cdots \\
 a[3] &\leq a[h+3] \leq a[2*h+3] \leq a[3*h+3] \leq a[4*h+3] \leq a[5*h+3] \cdots \\
 &\vdots \\
 a[h] &\leq a[h+h] \leq a[2*h+h] \leq a[3*h+h] \leq a[4*h+h] \leq a[5*h+h] \cdots
 \end{aligned}$$

と並び替えます。この並び替えには単純挿入法を使います。

そうして、とび幅 h をどんどん小さく、最後は $h = 1$ にすると並び替えは完了です。この h の選び方にコツがあって、小さいほうから 1, 4, 13, 40, 121, ... と

$$h_{i+1} = 3h_i + 1 \qquad h_1 = 1$$

とするのが良いらしい。良いというのは早いということである。最初に行う一番大きな h は、データの個数の半分以下にする。

Shell ソートの手順は、次の通りである。

1. 最初の飛び幅 h を決める。データの個数の半分以下で最大の h_i を最初の飛び幅とする。
2. $i = 1, 2, 3, \dots, h$ に対して、 $a[i], a[h+i], a[2*h+i], a[3*h+i], \dots$ を並び替える。
3. 次の $i++$ して、並び替える。
4. 次の $h=(h-1)/3$ にして、再度、並び替えを実行する。

リスト 1 の 19 行目以降に shell ソートのプログラムを書き、配列を小さい順 (昇順) に並び替えよ。

²この辺の説明は、www.rkmath.rikkyo.ac.jp/kida/shellsort.htm を参考にしている。

4 円周率の計算

夏休み明けに円周率 10 万桁計算のプログラミングコンテストを行う。これは、必須ではないが、興味のある者はプログラムを作成せよ。この部分は、山形大学の新聞久ーさんの「プログラミング演習 III」を参考にさせてもらっている。最初に、ヒントとして、長桁計算のプログラムを示す。

4.0.1 長桁計算

円周率を 10 万桁まで求めようとすると、長桁の計算が必要になる。しかし、C 言語の倍精度実数は 20 桁程度しか有効数字がない。そこで、長桁計算を考えることにする。

人間は時間と紙さえあれば、いくらでも長い桁の計算ができる。ただの計算なので、人間ができてコンピューターができないわけがない。人間と同じことをコンピューターにやらせれば良いのである。人間と同じように、コンピューターに長桁の筆算の計算をさせる。紙の代わりにデータは配列に記憶させるだけである。

これを最初から考えるのは、初心者には少し難しいので、リスト 2 にプログラムを載せておく。このプログラムでは、非常に大きな桁数の 2 つの整数を入力して、その和と差を計算することができる。

このプログラムでは、負の値は 10 の補数を用いている。もしある数が負の整数であれば、その絶対値の各桁を 10 から差し引いて、最後に 1 を加えている。この辺のところは、3 年生の電子計算機の授業で教えたはずである。ただ、そのときは 2 進法を使っていたので、2 の補数だったが、考え方は同じである。

リスト 2: 長桁の整数の和と差

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdlib.h>
4 #define N 32768
5 #define RADIX 10
6
7 void lf_scan(int n[]);
8 void lf_plus();
9 void lf_minus();
10 void lf_print(int n[]);
11 void lf_complement(int n[]);
12 void prt_bit(int n[]);
13 int a[N], b[N], Acc[N];
14
15 /*=====*/
16 /*  main                                     */
17 /*=====*/
18 int main(void){
19
20     lf_scan(a);
21     lf_scan(b);
22     lf_plus();
23     lf_print(Acc);
24     lf_minus();
25     lf_print(Acc);
26
27     return 0;
28 }
29
30
```

```

31  /*=====*/
32  /*  lf_scan                               */
33  /*=====*/
34  void lf_scan (int n[]) {
35      unsigned char key_in[N];
36      int i, l, flag=0;
37
38      scanf("%s", key_in);
39
40      l=strlen(key_in);
41
42      if (key_in[0] == '-') {
43          flag=1;
44          for (i=1; i<l; i++) {
45              key_in[i-1]=key_in[i];
46          }
47          l--;
48      }
49
50      for (i=0; i<l; i++) {
51          n[i]=(unsigned int)key_in[l-1-i]-48;
52      }
53
54      if (flag==1) lf_complement(n);
55  }
56
57  /*=====*/
58  /*  lf_plus                               */
59  /*=====*/
60  void lf_plus () {
61      int i;
62
63      for (i=0; i<N; i++) Acc[i] = a[i]+b[i];
64
65      for (i=0; i<N-1; i++) {
66          if (Acc[i] > 9) Acc[i+1]++;
67          Acc[i]%=RADIX;
68      }
69
70      Acc[N-1]%=RADIX;
71  }
72
73
74  /*=====*/
75  /*  lf_minus                             */
76  /*=====*/
77  void lf_minus () {
78
79      lf_complement(b);
80      lf_plus ();
81
82  }
83
84  /*=====*/
85  /*  lf_print                             */
86  /*=====*/
87  void lf_print (int n[]) {
88      int i, j, flag=0;
89
90      i=N-1;
91
92      if (n[i]>4) {

```

```

93     flag=1;
94     lf_complement(n);
95     printf("-");
96 }
97
98 while(n[i]==0 && i>0) i--;
99 for(j=i; j>=0; j--)printf("%d", n[j]);
100
101 if(flag==1)lf_complement(n);
102
103 printf("\n");
104
105 }
106 }
107
108 /*=====*/
109 /*  complement                               */
110 /*=====*/
111 void lf_complement(int n[]){
112     int i;
113
114     for(i=0; i<N; i++) n[i]=9-n[i];
115
116     n[0]++;
117
118     i=0;
119     while(n[i]==10 && i < N){
120         n[i]=0;
121         n[i+1]++;
122         i++;
123     }
124 }

```

4.0.2 円周率の計算

足し算と引き算の長桁の計算方法は分かった。わり算も同じである。後は自分で考えて、マチンの公式

$$\pi = 16 \arctan\left(\frac{1}{5}\right) - 4 \arctan\left(\frac{1}{239}\right)$$

とテイラー展開

$$\begin{aligned} \arctan(x) &= x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} - \frac{x^{11}}{11} + \frac{x^{13}}{13} - \frac{x^{15}}{15} + \dots \\ &= \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^{2n-1}}{2n-1} \end{aligned}$$

を用いて、10万桁を計算するプログラムを作成せよ。

4.1 課題提出要領

提出方法は、次の通りとする。

期限 8月25日(木) AM 8:55
用紙 A4
提出場所 山本研究室の入口のポスト
表紙 表紙を1枚つけて、以下の項目を分かりやすく記述すること。
授業科目名「計算機応用」
課題名「課題 夏休みの宿題」
5E 学籍番号 氏名
提出日
内容 ソースプログラム(プリントアウトでも、手書きでもOKとする)