アルゴリズム基礎及び演習レポート

ソート手順比較

7JFC1121 佐藤圭一

【目次】

7JFC1121 佐藤圭一

§１：プログラムの構成

今回は、ソートプログラム・手順測定プログラム・時間計測プログラムの３本で構成した。

プログラムの汎用性､及び簡略化の為､補助プログラムとして BATUTY･pfm686 を使用している。

データファイルは前回（挿入法）のデータを流用した。

§１．１：プログラム概要

SORT.PAS	ソートプログラム 挿入法・QuickSort の昇順・降順ソートを実現している
SORT_O.PAS	手順測定プログラム SORT.PAS に Order カウント機能を付加している
STOP_W.BAT	時間計測プログラム(Interface)
STOP_W2.BAT	時間計測プログラム(Main) 100 万回各ソートを繰り返し演算させ、実行時間を計測する

§１．２：補助プログラム概要

BATUTY	バッチファイル拡張機能提供ソフト
Pfm686	コンピュータパフォーマンス計測ツール

7JFC1121 佐藤圭一

§２：プログラムソース・補助プログラム詳細

§２．１：[SORT.PAS]

```
program sort; {7JFC1121  佐藤圭一}

{変数の宣言}

type item     = record
                  id          : integer;
                  math        : integer;
                  english : integer
       end;

type ar_int = array[0..51,0..2] of integer;

var   readfile                                  : file of item ;
       data                                       : item              ;
       filename                                  : string[12]     ;
       i_data,o_data                          : ar_int           ;
       flag,flag2,left,right              : integer          ;
       g1,g2,lp,temp                         : longint          ;

{クイックソート手続き}

procedure quick_sort(var o_data              : ar_int     ;
                         flag,left,right             :integer );
  var i,j,k,w,some,pivot : integer;

  begin
    some    := (left + right ) div 2;
    pivot := o_data[some,flag];
    i         := left;
    j         := right;

    repeat
      begin

        while o_data[i,flag] < pivot do
          i := i+1;

        while o_data[j,flag] > pivot do
          j := j-1;

        if (i <= j) then
          begin
            for k := 0 to 2 do
              begin
                w                 := o_data[i,k];
                o_data[i,k] := o_data[j,k];
                o_data[j,k] := w
              end;
            i := i+1;
            j := j-1
          end
      end;
```

```
    until i > j;
    if (j > left) then
       quick_sort(o_data,flag,left,j);

    if (i < right) then
       quick_sort(o_data,flag,i,right)

  end;
```

{降順クイックソート手続き}

```
procedure quick_sort_rev(var o_data              : ar_int     ;
                             flag,left,right     :integer );
  var i,j,k,w,some,pivot : integer;

  begin
    some   := (left + right ) div 2;
    pivot := o_data[some,flag] ;
    i       := left;
    j       := right;

    repeat
      begin

        while o_data[i,flag] > pivot do
          i := i+1;

        while o_data[j,flag] < pivot do
          j := j-1;

        if (i <= j) then
          begin
            for k := 0 to 2 do
              begin
                w                := o_data[i,k];
                o_data[i,k] := o_data[j,k];
                o_data[j,k] := w
              end;
            i := i+1;
            j := j-1
          end
      end;
    until i > j;
    if (j > left) then
      quick_sort_rev(o_data,flag,left,j);

    if (i < right) then
      quick_sort_rev(o_data,flag,i,right)

  end;
```

{ソート手続き}

```
procedure sort_ins(var o_data : ar_int    ;
                           flag           : integer);
  var i,j,k,w,x,y,z : integer;
  begin
    for i := 2 to 50 do
      begin
        x := o_data[i,0];
        y := o_data[i,1];
```

```
            z := o_data[i,2];
            w := o_data[i,flag];
            o_data[0,flag] := w;
            j := i - 1;

            while w < o_data[j,flag] do
               begin
                  o_data[j+1,0] := o_data[j,0];
                  o_data[j+1,1] := o_data[j,1];
                  o_data[j+1,2] := o_data[j,2];
                  j := j-1
               end;
            o_data[j+1,0] := x;
            o_data[j+1,1] := y;
            o_data[j+1,2] := z;
            o_data[j+1,flag] := w
         end
   end;
```

{**降順ソート手続き**}

```
procedure sort_ins_rev(var o_data    : ar_int    ;
                                flag              : integer);
   var i,j,k,w,x,y,z : integer;
   begin
      for i := 49 downto 1 do
         begin
            x := o_data[i,0];
            y := o_data[i,1];
            z := o_data[i,2];
            w := o_data[i,flag];
            o_data[51,flag] := w;
            j := i + 1;

            while w < o_data[j,flag] do
               begin
                  o_data[j-1,0] := o_data[j,0];
                  o_data[j-1,1] := o_data[j,1];
                  o_data[j-1,2] := o_data[j,2];
                  j := j+1
               end;
            o_data[j-1,0] := x;
            o_data[j-1,1] := y;
            o_data[j-1,2] := z;
            o_data[j-1,flag] := w
         end
   end;
```

{**ファイル読み込み手続き**}

```
procedure file_load;

   begin
      assign(readfile,filename);
      reset(readfile);
      with data do
         for temp := 1 to 50 do
            begin
               read(readfile,data);
               o_data[temp,0] := id;
               o_data[temp,1] := math;
               o_data[temp,2] := english;
               i_data[temp,0] := id;
```

```
            i_data[temp,1] := math;
            i_data[temp,2] := english;
          end;
    close(readfile)
  end;


{ソート呼出・繰り返し手続き}
procedure sort_data;
  begin
    if ( lp >= 20) then
      begin
        g1 := lp div 20;
        g2 := g1;
        writeln('進行状況:_____ [「~」一個:',g1,'回 全部で',lp,'回]');
        write    ('       (-_-/')
      end
      else
        g2 := 999;

    for temp := 1 to lp do
      begin
        if (g2 < temp) then
          begin
            write('~');
            g2 := g2 + g1;
          end;
        o_data := i_data;
        left := 1;
        right := 50;
        case flag2 of
          1 : sort_ins(o_data,flag);
          2 : sort_ins_rev(o_data,flag);
          3 : quick_sort(o_data,flag,left,right);
          4 : quick_sort_rev(o_data,flag,left,right)
        end
      end;
      writeln('~(^^/ ｵﾜｯﾀﾖ
  end;


{画面表示手続き}

procedure data_out;

  begin
    writeln('＊＊＊処理結果＊＊＊');
    write('[ ',filename,' ]の内容を、');
    if flag = 0 then
      write('学生証番号');
    if flag = 1 then
      write('数学');
    if flag = 2 then
      write('英語');
    write('順に');
    case flag2 of
      1 : write('挿入法で昇順');
      2 : write('挿入法で降順');
      3 : write('Quick Sort で昇順');
      4 : write('Quick Sort で降順')
    end;
    writeln('ソートしました。');
    writeln('ID:学生証番号  MT:数学  EN:英語');
    writeln('1..10_____11..20_____21..30_____31..40_____41..50____');
    writeln('ID     MT EN ID     MT EN ID     MT EN ID     MT EN ID     MT EN');
```

```
    for temp := 1 to 10 do
      begin
        flag := temp;
        write   (o_data[flag,0],' ',o_data[flag,1],' ',o_data[flag,2],' ');
        flag := temp+10;
        write   (o_data[flag,0],' ',o_data[flag,1],' ',o_data[flag,2],' ');
        flag := temp+20;
        write   (o_data[flag,0],' ',o_data[flag,1],' ',o_data[flag,2],' ');
        flag := temp+30;
        write   (o_data[flag,0],' ',o_data[flag,1],' ',o_data[flag,2],' ');
        flag := temp+40;
        writeln(o_data[flag,0],' ',o_data[flag,1],' ',o_data[flag,2],' ');
      end
  end;
```

{**メインルーチン**}

```
begin
```

{**データファイルの名前を取得**}

```
  writeln('＊＊＊ソートの実践・比較プログラム＊＊＊');
  writeln('Created By eucalyptus. 1998');
  write   ('データファイル名を入力して下さい:');
  readln (filename);
```

{**ファイル読み込み**}

```
  file_load;
```

{**処理の選択**}

```
  writeln('ファイルの読み込み完了。処理を選択して下さい。');
  write('[0:学番でソート(笑) 1:数学でソート  2:英語でソート]:');
  readln(flag);
  write('[1:挿入法昇順  2:挿入法降順  3:quickSort 昇順  4:Quicksort 降順]:');
  readln(flag2);
  write('繰り返し回数を入力して下さい:');
  readln(lp);
```

{**ソート**}

```
  sort_data;
```

{**処理結果出力**}

```
  data_out

end.
```

§２．２：[SORT_O.PAS]

```
program sort_o; {7JFC1121  佐藤圭一}

{変数の宣言}

type item     = record
                    id          : integer;
                    math        : integer;
                    english : integer
        end;

type ar_int = array[0..51,0..2] of integer;

var   readfile                                  : file of item ;
        data                                      : item              ;
        filename                                  : string[12]     ;
        i_data,o_data                             : ar_int           ;
        flag,flag2,left,right                 : integer          ;
        temp,lp,g1,g2,order                       : longint          ;

{クイックソート手続き}

procedure quick_sort(var o_data          : ar_int   ;
                              flag,left,right :integer   ;
                              var order           : longint);
   var i,j,k,w,some,pivot : integer;

   begin
      some    := (left + right ) div 2;
      pivot := o_data[some,flag];
      i         := left;
      j         := right;
      order := order + 6;


      repeat
         begin

            while o_data[i,flag] < pivot do
               begin
                  i := i+1;
                  order := order + 2
               end;

            while o_data[j,flag] > pivot do
               begin
                 j := j-1;
                  order := order + 2
               end;

            if (i <= j) then
               begin
                  for k := 0 to 2 do
                     begin
                        w                  := o_data[i,k];
                        o_data[i,k] := o_data[j,k];
                        o_data[j,k] := w;
                        order := order + 4
                     end;
                  i := i+1;
                  j := j-1;
```

```
                order := order + 2
              end;
          order := order + 1
          end;
       order := order + 1;
       until i > j;
       order := order + 1;
       if (j > left) then
          begin
             order := order + 1;
             quick_sort(o_data,flag,left,j,order)
          end;

       order := order + 1;
       if (i < right) then
          begin
             order := order + 1;
             quick_sort(o_data,flag,i,right,order)
          end

    end;
```

{降順クイックソート手続き}

```
procedure quick_sort_rev(var o_data    :ar_int    ;
                             flag,left,right :integer ;
                             var order             :longint);
    var i,j,k,w,some,pivot : integer;

    begin
       some    := (left + right ) div 2;
       pivot := o_data[some,flag] ;
       i         := left;
       j         := right;
       order := order + 6;

       repeat
          begin

             while o_data[i,flag] > pivot do
                begin
                   i := i+1;
                   order := order + 2
                end;

             while o_data[j,flag] < pivot do
                begin
                   j := j-1;
                   order := order + 2
                end;

             if (i <= j) then
                begin
                   for k := 0 to 2 do
                      begin
                         w                  := o_data[i,k];
                         o_data[i,k] := o_data[j,k];
                         o_data[j,k] := w;
                         order := order + 4
                      end;
                   i := i+1;
                   j := j-1;
                   order := order + 2
                end;
```


7JFC1121 佐藤圭一

```
        order := order + 1
        end;
    order := order + 1;
    until i > j;
    order := order + 1;
    if (j > left) then
        begin
            quick_sort_rev(o_data,flag,left,j,order);
            order := order + 1
        end;

    order := order + 1;
    if (i < right) then
        begin
            quick_sort_rev(o_data,flag,i,right,order);
            order := order + 1
        end
  end;
```

{**ソート手続き**}

```
procedure sort_ins(var o_data : ar_int    ;
                           flag           : integer ;
                           var order      : longint);
  var i,j,k,w,x,y,z : integer;
  begin
    for i := 2 to 50 do
        begin
            x := o_data[i,0];
            y := o_data[i,1];
            z := o_data[i,2];
            w := o_data[i,flag];
            o_data[0,flag] := w;
            j := i - 1;
            order := order + 7;

            while w < o_data[j,flag] do
                begin
                    o_data[j+1,0] := o_data[j,0];
                    o_data[j+1,1] := o_data[j,1];
                    o_data[j+1,2] := o_data[j,2];
                    j := j-1;
                    order := order + 5;

                end;
            o_data[j+1,0] := x;
            o_data[j+1,1] := y;
            o_data[j+1,2] := z;
            o_data[j+1,flag] := w;
            order := order + 5;

        end
  end;
```

{**降順ソート手続き**}

```
procedure sort_ins_rev(var o_data : ar_int    ;
                               flag           : integer ;
                               var order      : longint);
  var i,j,k,w,x,y,z : integer;
  begin
```

```
      for i := 49 downto 1 do
         begin
            x := o_data[i,0];
            y := o_data[i,1];
            z := o_data[i,2];
            w := o_data[i,flag];
            o_data[51,flag] := w;
            j := i + 1;
            order := order + 7;

            while w < o_data[j,flag] do
               begin
                  o_data[j-1,0] := o_data[j,0];
                  o_data[j-1,1] := o_data[j,1];
                  o_data[j-1,2] := o_data[j,2];
                  j := j+1;
                  order := order + 5
               end;
            o_data[j-1,0] := x;
            o_data[j-1,1] := y;
            o_data[j-1,2] := z;
            o_data[j-1,flag] := w;
            order := order + 4
         end
   end;
```

{ファイル読み込み手続き}

```
procedure file_load;

   begin
      assign(readfile,filename);
      reset(readfile);
      with data do
         for temp := 1 to 50 do
            begin
               read(readfile,data);
               o_data[temp,0] := id;
               o_data[temp,1] := math;
               o_data[temp,2] := english;
               i_data[temp,0] := id;
               i_data[temp,1] := math;
               i_data[temp,2] := english;
            end;
      close(readfile)
   end;
```

{ソート呼出・繰り返し手続き}

```
procedure sort_data;
   begin
      order := 0;
      if ( lp >= 20) then
         begin
            g1 := lp div 20;
            g2 := g1;
            writeln('進行状況:_____ [「~」一個:',g1,'回 全部で',lp,'回]');
            write    ('       (-_-/')
         end
         else
            g2 := 999;

      for temp := 1 to lp do
         begin
```

```
            if (g2 < temp) then
               begin
                  write('~');
                  g2 := g2 + g1;
               end;
            o_data := i_data;
            left := 1;
            right := 50;
            case flag2 of
               1 : sort_ins(o_data,flag,order);
               2 : sort_ins_rev(o_data,flag,order);
               3 : quick_sort(o_data,flag,left,right,order);
               4 : quick_sort_rev(o_data,flag,left,right,order)
            end
         end;
         if (g2 <> 999) then
         writeln('~(^^/ ｵﾜｯﾀﾖ

   end;
```

{**画面表示手続き**}

```
procedure data_out;

   begin
      writeln('＊＊＊処理結果＊＊＊');
      write('[ ',filename,' ]の内容を、');
      if flag = 0 then
         write('学生証番号');
      if flag = 1 then
         write('数学');
      if flag = 2 then
         write('英語');
      write('順に');
      case flag2 of
         1 : write('挿入法で昇順');
         2 : write('挿入法で降順');
         3 : write('Quick Sort で昇順');
         4 : write('Quick Sort で降順')
      end;
      writeln('ソートしました。');
      writeln(order,' 回手順を踏みました。');
      writeln('ID:学生証番号  MT:数学  EN:英語');
      writeln('1..10_____11..20_____21..30_____31..40_____41..50____');
      writeln('ID     MT EN ID     MT EN ID     MT EN ID     MT EN ID     MT EN');
      for temp := 1 to 10 do
         begin
            flag := temp;
            write   (o_data[flag,0],' ',o_data[flag,1],' ',o_data[flag,2],' ');
            flag := temp+10;
            write   (o_data[flag,0],' ',o_data[flag,1],' ',o_data[flag,2],' ');
            flag := temp+20;
            write   (o_data[flag,0],' ',o_data[flag,1],' ',o_data[flag,2],' ');
            flag := temp+30;
            write   (o_data[flag,0],' ',o_data[flag,1],' ',o_data[flag,2],' ');
            flag := temp+40;
            writeln(o_data[flag,0],' ',o_data[flag,1],' ',o_data[flag,2],' ');
         end
   end;
```

{**メインルーチン**}

```
begin
```

{データファイルの名前を取得}

```
  writeln('＊＊＊ソートの実践・比較プログラム 手順表示機能付き＊＊＊');
  writeln('Created By eucalyptus. 1998');
  write   ('データファイル名を入力して下さい:');
  readln (filename);
```

{ファイル読み込み}

```
  file_load;
```

{処理の選択}

```
  writeln('ファイルの読み込み完了。処理を選択して下さい。');
  write('[0:学番でソート(笑) 1:数学でソート 2:英語でソート]:');
  readln(flag);
  write('[1:挿入法昇順 2:挿入法降順 3:quickSort 昇順 4:Quicksort 降順]:');
  readln(flag2);
  write('繰り返し回数を入力して下さい:');
  readln(lp);
```

{ソート}

```
  sort_data;
```

{処理結果出力}

```
  data_out

end.
```

§２．３：[STOP_W.BAT]

```
@echo off
del autolog.log
echo 開始
call stop_w2 1000000 0 1
call stop_w2 1000000 0 2
call stop_w2 1000000 0 3
call stop_w2 1000000 0 4
call stop_w2 1000000 1 1
call stop_w2 1000000 1 2
call stop_w2 1000000 1 3
call stop_w2 1000000 1 4
call stop_w2 1000000 2 1
call stop_w2 1000000 2 2
call stop_w2 1000000 2 3
call stop_w2 1000000 2 4
echo 完了
```

§２．４：[STOP_W2.BAT]

```
@echo off
BU -e -nSEC CLOCKSECOND
BU -e -nMIN CLOCKMINUTE
BU -e -nSEC CALC %SEC% + ( %MIN% * 60 )
echo seiseki.dat> autolog.dat
echo %2>> autolog.dat
echo %3>> autolog.dat
echo %1>> autolog.dat
sort <autolog.dat >nul
BU -e -nSEC2 CLOCKSECOND
BU -e -nMIN CLOCKMINUTE
BU -e -nSEC CALC %SEC2% + ( %MIN% * 60 ) - SEC
echo 1_%2,2_%3,%SEC% >>autolog.log
```

§２．５：BATUTY 詳細

使用法:

BU [-w] [-e[STR]] [-n[NAME]] [-r] **コマンド** .... [**パラメータ**...]

各コマンド実行後 ERRORLEVEL **に** Ret **の数値を返します**.
なお ERRORLEVEL **には** 255 **までしか値が返りませんが** -e **では** 65536 **までの数を扱う事ができます**.

コマンドは他のコマンドと区別がつく任意の位置で省略可能です. **但しバッチファイル中等での使用時にはバージョンアップで似たようなコマンドが増える可能性があるのでなるべく省略しないでください**.

省略例)DSPERRORLEVEL → DSPERR

スイッチ:

スイッチの文字は英大, **小文字可**. **コマンドの前に** "-" **か** "/" **で指定してください**. **スイッチとスイッチの間はスペースを空けてください**.

* -e[STR]
コマンド実行後のリターン値(ERRORLEVEL)**を標準出力表示**

STR **を指定すれば数値の前に文字を表示します**.

例)BU -e **マシンコードは** MACHINE

* -n[NAME],-d[NAME] (NAME **のデフォルトは** ENVTMP)
コマンド実行後の標準出力を環境変数に代入

NAME **を指定すれば任意の環境変数に代入できます**. NAME **を指定せずに** -n **だけを指定時は** ENVTMP **という名前の環境変数に代入されます**.
キー入力のコマンドでは入力文字が代入されます(-e **時は** ret **を代入**)

例)BU -nCDIR CURDRVPATH (**カレントドライブ**:**パスを** CDIR **に代入**)
BU -e -nMCODE MACHINE (MACHINE **の戻り値を** MCODE **に代入**)

なお環境変数への代入は最大 512 **文字です**. -n **も**-d **も同じ意味です**.

* CLOCKYEAR (**年の下２桁**)
* CLOCKMONTH (**月**)
* CLOCKDAY (**日**)
* CLOCKWEEK (**曜日**,0=**日** 1=**月** .. 6=**土**)
* CLOCKHOUR (**時**)
* CLOCKMINUTE (**分**)
* CLOCKSECOND (**秒**)
日時を得る
Ret= **得た数値**

* CALC **計算式**
環境変数を数値変数とみなして計算をする(**数値は** 65535 **まで**)
変数には環境変数または数値を使用できます. **計算した答を** ERRORLEVEL **に返します**. ERRORLEVEL **としては** 254 **以内しか返せませんが**, **環境変数などへのセットは** 65535 **まで可能です**.
オーバーフロー等は考慮せずに 16BIT **で単純計算を行っています**. **例えば**(0-2)**を行うと** 65534 **が計算値になります**.
サポートする演算子は以下のものです.

		(**優先順位**)
変数	: **環境変数自体の値**	
()	: **括弧(先に演算)**	1
変数++	: **環境変数に** 1	

加える 2

変数-- ：環境変数を 1 減ずる 2
* ：乗算(掛け算) 3
/ ：剰算(割り算) 3
% ：剰余(割り算をした余り) 3
\+ ：加算(足し算) 4
\- ：減算(引き算) 4
& : AND 5
$: OR 5
^ : XOR 5

式は複数を混ぜ合わせて使用できますが，演算子は優先順位を持っており優先順位の高いもの(上記数値の小さいもの)ほど優先度が高く先に計算されます.

Ret= 0〜253 : 環境変数の値または計算値
254 : 環境変数の値または計算値が 254 以上
255 : 数値が大きすぎる/変数がない/変数が数値と違う

例)BU CALC ENVTMP--
BU -e -nENVANS CALC ENVRET/10 (ENVRET を 10 で割り ENVANS に入れる)
BU -e -nENVANS CALC 2*ENVBAR+20*(ENVFOO+2)

7JFC1121 佐藤圭一

§３：プログラム実行結果

§３．１：[SORT.PAS]（一例）

＊＊＊ソートの実践・比較プログラム＊＊＊
Created By eucalyptus. 1998
データファイル名を入力して下さい:seiseki.dat
ファイルの読み込み完了。処理を選択して下さい。
0:**学番でソート**(**笑**) 1:**数学でソート** 2:**英語でソート**]:1
1:**挿入法昇順** 2:**挿入法降順** 3:quickSort **昇順** 4:Quicksort **降順**]:3
繰り返し回数を入力して下さい:100
進行状況:_____ [**「~」一個**:5 **回 全部で** 100 **回**]
(-_-/~~~~~~~~~~~~~~~~~~~~(^^/ ｵﾜｯﾀﾖ
＊＊＊**処理結果**＊＊＊
seiseki.dat]**の内容を、数学順に** Quick Sort **で昇順ソートしました。**
D:**学生証番号** MT:**数学** EN:**英語**
..10_____11..20_____21..30_____31..40_____41..50____
D MT EN ID MT EN ID MT EN ID MT EN ID MT EN
046 12 39 1017 40 28 1035 60 68 1027 64 79 1019 76 67
031 16 49 1025 44 69 1008 60 80 1012 64 69 1005 76 58
026 16 39 1044 44 51 1004 60 57 1034 68 88 1016 76 81
037 20 51 1042 44 50 1029 64 50 1030 68 79 1050 80 92
006 24 53 1040 52 44 1020 64 61 1011 68 74 1045 80 67
003 32 59 1041 56 79 1048 64 74 1018 68 68 1039 80 76
002 32 29 1043 56 82 1014 64 67 1033 68 83 1022 80 81
001 32 60 1015 56 81 1007 64 81 1032 68 73 1010 84 88
038 32 42 1049 56 71 1023 64 86 1028 72 71 1021 84 54
047 40 32 1024 60 64 1013 64 53 1009 72 60 1036 88 79

§３．２：[SORT_0.PAS]（一例）

データファイル名を入力して下さい:seiseki.dat
ファイルの読み込み完了。処理を選択して下さい。
[0:**学番でソート**(**笑**) 1:**数学でソート** 2:**英語でソート**]:1
[1:**挿入法昇順** 2:**挿入法降順** 3:quickSort **昇順** 4:Quicksort **降順**]:3
繰り返し回数を入力して下さい:30
進行状況:_____ [**「~」一個**:1 **回 全部で** 30 **回**]
(-_-/~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~(^^/ ｵﾜｯﾀﾖ
＊＊＊**処理結果**＊＊＊
[seiseki.dat]**の内容を、数学順に** Quick Sort **で昇順ソートしました。**
62040 **回手順を踏みました。**
ID:**学生証番号** MT:**数学** EN:**英語**
1..10_____11..20_____21..30_____31..40_____41..50____
ID MT EN ID MT EN ID MT EN ID MT EN ID MT EN
1046 12 39 1017 40 28 1035 60 68 1027 64 79 1019 76 67
1031 16 49 1025 44 69 1008 60 80 1012 64 69 1005 76 58
1026 16 39 1044 44 51 1004 60 57 1034 68 88 1016 76 81
1037 20 51 1042 44 50 1029 64 50 1030 68 79 1050 80 92
1006 24 53 1040 52 44 1020 64 61 1011 68 74 1045 80 67
1003 32 59 1041 56 79 1048 64 74 1018 68 68 1039 80 76
1002 32 29 1043 56 82 1014 64 67 1033 68 83 1022 80 81
1001 32 60 1015 56 81 1007 64 81 1032 68 73 1010 84 88
1038 32 42 1049 56 71 1023 64 86 1028 72 71 1021 84 54
1047 40 32 1024 60 64 1013 64 53 1009 72 60 1036 88 79

§３．３[AUTOLOG.DAT](STOP_W.BAT により作られるCSV ファイル)

1_0,2_1,23
1_0,2_2,388
1_0,2_3,43
1_0,2_4,56
1_1,2_1,199
1_1,2_2,190
1_1,2_3,86
1_1,2_4,88
1_2,2_1,199
1_2,2_2,208
1_2,2_3,83
1_2,2_4,84

§４：処理結果

§４．1：学生証番号順・挿入法・昇順

[seiseki.dat]の内容を、学生証番号順に挿入法で昇順ソートしました。
588 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語
1..10_____11..20_____21..30_____31..40_____41..50____
ID MT EN ID MT EN ID MT EN ID MT EN ID MT EN
1001 32 60 1011 68 74 1021 84 54 1031 16 49 1041 56 79
1002 32 29 1012 64 69 1022 80 81 1032 68 73 1042 44 50
1003 32 59 1013 64 53 1023 64 86 1033 68 83 1043 56 82
1004 60 57 1014 64 67 1024 60 64 1034 68 88 1044 44 51
1005 76 58 1015 56 81 1025 44 69 1035 60 68 1045 80 67
1006 24 53 1016 76 81 1026 16 39 1036 88 79 1046 12 39
1007 64 81 1017 40 28 1027 64 79 1037 20 51 1047 40 32
1008 60 80 1018 68 68 1028 72 71 1038 32 42 1048 64 74
1009 72 60 1019 76 67 1029 64 50 1039 80 76 1049 56 71
1010 84 88 1020 64 61 1030 68 79 1040 52 44 1050 80 92

§４．2：学生証番号順・挿入法・降順

[seiseki.dat]の内容を、学生証番号順に挿入法で降順ソートしました。
6664 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語
1..10_____11..20_____21..30_____31..40_____41..50____
ID MT EN ID MT EN ID MT EN ID MT EN ID MT EN
1050 80 92 1040 52 44 1030 68 79 1020 64 61 1010 84 88
1049 56 71 1039 80 76 1029 64 50 1019 76 67 1009 72 60
1048 64 74 1038 32 42 1028 72 71 1018 68 68 1008 60 80
1047 40 32 1037 20 51 1027 64 79 1017 40 28 1007 64 81
1046 12 39 1036 88 79 1026 16 39 1016 76 81 1006 24 53
1045 80 67 1035 60 68 1025 44 69 1015 56 81 1005 76 58
1044 44 51 1034 68 88 1024 60 64 1014 64 67 1004 60 57
1043 56 82 1033 68 83 1023 64 86 1013 64 53 1003 32 59
1042 44 50 1032 68 73 1022 80 81 1012 64 69 1002 32 29
1041 56 79 1031 16 49 1021 84 54 1011 68 74 1001 32 60

§４．3：学生証番号順・QuickSort・昇順

[seiseki.dat]の内容を、学生証番号順に Quick Sort で昇順ソートしました。
1160 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語
1..10_____11..20_____21..30_____31..40_____41..50____
ID MT EN ID MT EN ID MT EN ID MT EN ID MT EN
1001 32 60 1011 68 74 1021 84 54 1031 16 49 1041 56 79
1002 32 29 1012 64 69 1022 80 81 1032 68 73 1042 44 50
1003 32 59 1013 64 53 1023 64 86 1033 68 83 1043 56 82
1004 60 57 1014 64 67 1024 60 64 1034 68 88 1044 44 51
1005 76 58 1015 56 81 1025 44 69 1035 60 68 1045 80 67
1006 24 53 1016 76 81 1026 16 39 1036 88 79 1046 12 39
1007 64 81 1017 40 28 1027 64 79 1037 20 51 1047 40 32
1008 60 80 1018 68 68 1028 72 71 1038 32 42 1048 64 74
1009 72 60 1019 76 67 1029 64 50 1039 80 76 1049 56 71
1010 84 88 1020 64 61 1030 68 79 1040 52 44 1050 80 92

7JFC1121 佐藤圭一

§４．４：学生証番号順・QuickSort・降順

[seiseki.dat]の内容を、学生証番号順に Quick Sort で降順ソートしました。
1454 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語
1..10_____11..20_____21..30_____31..40_____41..50____
ID MT EN ID MT EN ID MT EN ID MT EN ID MT EN
1050 80 92 1040 52 44 1030 68 79 1020 64 61 1010 84 88
1049 56 71 1039 80 76 1029 64 50 1019 76 67 1009 72 60
1048 64 74 1038 32 42 1028 72 71 1018 68 68 1008 60 80
1047 40 32 1037 20 51 1027 64 79 1017 40 28 1007 64 81
1046 12 39 1036 88 79 1026 16 39 1016 76 81 1006 24 53
1045 80 67 1035 60 68 1025 44 69 1015 56 81 1005 76 58
1044 44 51 1034 68 88 1024 60 64 1014 64 67 1004 60 57
1043 56 82 1033 68 83 1023 64 86 1013 64 53 1003 32 59
1042 44 50 1032 68 73 1022 80 81 1012 64 69 1002 32 29
1041 56 79 1031 16 49 1021 84 54 1011 68 74 1001 32 60

§４．５：数学順・挿入法・昇順

[seiseki.dat]の内容を、数学順に挿入法で昇順ソートしました。
3518 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語
1..10_____11..20_____21..30_____31..40_____41..50____
ID MT EN ID MT EN ID MT EN ID MT EN ID MT EN
1046 12 39 1047 40 32 1008 60 80 1029 64 50 1005 76 58
1026 16 39 1025 44 69 1024 60 64 1048 64 74 1016 76 81
1031 16 49 1042 44 50 1035 60 68 1011 68 74 1019 76 67
1037 20 51 1044 44 51 1007 64 81 1018 68 68 1022 80 81
1006 24 53 1040 52 44 1012 64 69 1030 68 79 1039 80 76
1001 32 60 1015 56 81 1013 64 53 1032 68 73 1045 80 67
1002 32 29 1041 56 79 1014 64 67 1033 68 83 1050 80 92
1003 32 59 1043 56 82 1020 64 61 1034 68 88 1010 84 88
1038 32 42 1049 56 71 1023 64 86 1009 72 60 1021 84 54
1017 40 28 1004 60 57 1027 64 79 1028 72 71 1036 88 79

§４．６：数学順・挿入法・降順

[seiseki.dat]の内容を、数学順に挿入法で降順ソートしました。
3309 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語
1..10_____11..20_____21..30_____31..40_____41..50____
ID MT EN ID MT EN ID MT EN ID MT EN ID MT EN
1036 88 79 1009 72 60 1013 64 53 1035 60 68 1047 40 32
1010 84 88 1028 72 71 1014 64 67 1015 56 81 1001 32 60
1021 84 54 1011 68 74 1020 64 61 1041 56 79 1002 32 29
1022 80 81 1018 68 68 1023 64 86 1043 56 82 1003 32 59
1039 80 76 1030 68 79 1027 64 79 1049 56 71 1038 32 42
1045 80 67 1032 68 73 1029 64 50 1040 52 44 1006 24 53
1050 80 92 1033 68 83 1048 64 74 1025 44 69 1037 20 51
1005 76 58 1034 68 88 1004 60 57 1042 44 50 1026 16 39
1016 76 81 1007 64 81 1008 60 80 1044 44 51 1031 16 49
1019 76 67 1012 64 69 1024 60 64 1017 40 28 1046 12 39

7JFC1121 佐藤圭一

§４．７：数学順・QuickSort・昇順

[seiseki.dat]の内容を、数学順に Quick Sort で昇順ソートしました。
2068 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語

```
1..10_____11..20_____21..30_____31..40_____41..50____
ID   MT EN ID   MT EN ID   MT EN ID   MT EN ID   MT EN
1046 12 39 1017 40 28 1035 60 68 1027 64 79 1019 76 67
1031 16 49 1025 44 69 1008 60 80 1012 64 69 1005 76 58
1026 16 39 1044 44 51 1004 60 57 1034 68 88 1016 76 81
1037 20 51 1042 44 50 1029 64 50 1030 68 79 1050 80 92
1006 24 53 1040 52 44 1020 64 61 1011 68 74 1045 80 67
1003 32 59 1041 56 79 1048 64 74 1018 68 68 1039 80 76
1002 32 29 1043 56 82 1014 64 67 1033 68 83 1022 80 81
1001 32 60 1015 56 81 1007 64 81 1032 68 73 1010 84 88
1038 32 42 1049 56 71 1023 64 86 1028 72 71 1021 84 54
1047 40 32 1024 60 64 1013 64 53 1009 72 60 1036 88 79
```

§４．８：数学順・QuickSort・降順

[seiseki.dat]の内容を、数学順に Quick Sort で降順ソートしました。
2090 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語

```
1..10_____11..20_____21..30_____31..40_____41..50____
ID   MT EN ID   MT EN ID   MT EN ID   MT EN ID   MT EN
1036 88 79 1028 72 71 1023 64 86 1008 60 80 1047 40 32
1021 84 54 1009 72 60 1014 64 67 1049 56 71 1038 32 42
1010 84 88 1034 68 88 1013 64 53 1015 56 81 1001 32 60
1045 80 67 1018 68 68 1012 64 69 1041 56 79 1002 32 29
1050 80 92 1033 68 83 1048 64 74 1043 56 82 1003 32 59
1022 80 81 1032 68 73 1020 64 61 1040 52 44 1006 24 53
1039 80 76 1011 68 74 1007 64 81 1044 44 51 1037 20 51
1016 76 81 1030 68 79 1035 60 68 1042 44 50 1026 16 39
1019 76 67 1027 64 79 1024 60 64 1025 44 69 1031 16 49
1005 76 58 1029 64 50 1004 60 57 1017 40 28 1046 12 39
```

§４．９：英語順・挿入法・昇順

[seiseki.dat]の内容を、英語順に挿入法で昇順ソートしました。
3513 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語

```
1..10_____11..20_____21..30_____31..40_____41..50____
ID   MT EN ID   MT EN ID   MT EN ID   MT EN ID   MT EN
1017 40 28 1037 20 51 1020 64 61 1049 56 71 1007 64 81
1002 32 29 1044 44 51 1024 60 64 1032 68 73 1015 56 81
1047 40 32 1006 24 53 1014 64 67 1011 68 74 1016 76 81
1026 16 39 1013 64 53 1019 76 67 1048 64 74 1022 80 81
1046 12 39 1021 84 54 1045 80 67 1039 80 76 1043 56 82
1038 32 42 1004 60 57 1018 68 68 1027 64 79 1033 68 83
1040 52 44 1005 76 58 1035 60 68 1030 68 79 1023 64 86
1031 16 49 1003 32 59 1012 64 69 1036 88 79 1010 84 88
1029 64 50 1001 32 60 1025 44 69 1041 56 79 1034 68 88
1042 44 50 1009 72 60 1028 72 71 1008 60 80 1050 80 92
```


7JFC1121 佐藤圭一

§４．10：英語順・挿入法・降順

[seiseki.dat]の内容を、英語順に挿入法で降順ソートしました。
3614 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語
1..10_____11..20_____21..30_____31..40_____41..50____
ID MT EN ID MT EN ID MT EN ID MT EN ID MT EN
1050 80 92 1008 60 80 1049 56 71 1001 32 60 1029 64 50
1010 84 88 1027 64 79 1012 64 69 1009 72 60 1042 44 50
1034 68 88 1030 68 79 1025 44 69 1003 32 59 1031 16 49
1023 64 86 1036 88 79 1018 68 68 1005 76 58 1040 52 44
1033 68 83 1041 56 79 1035 60 68 1004 60 57 1038 32 42
1043 56 82 1039 80 76 1014 64 67 1021 84 54 1026 16 39
1007 64 81 1011 68 74 1019 76 67 1006 24 53 1046 12 39
1015 56 81 1048 64 74 1045 80 67 1013 64 53 1047 40 32
1016 76 81 1032 68 73 1024 60 64 1037 20 51 1002 32 29
1022 80 81 1028 72 71 1020 64 61 1044 44 51 1017 40 28

§４．１１：英語順・QuickSort・昇順

[seiseki.dat]の内容を、英語順に Quick Sort で昇順ソートしました。
1974 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語
1..10_____11..20_____21..30_____31..40_____41..50____
ID MT EN ID MT EN ID MT EN ID MT EN ID MT EN
1017 40 28 1037 20 51 1020 64 61 1028 72 71 1007 64 81
1002 32 29 1044 44 51 1024 60 64 1032 68 73 1022 80 81
1047 40 32 1006 24 53 1014 64 67 1048 64 74 1016 76 81
1026 16 39 1013 64 53 1019 76 67 1011 68 74 1015 56 81
1046 12 39 1021 84 54 1045 80 67 1039 80 76 1043 56 82
1038 32 42 1004 60 57 1035 60 68 1027 64 79 1033 68 83
1040 52 44 1005 76 58 1018 68 68 1030 68 79 1023 64 86
1031 16 49 1003 32 59 1025 44 69 1041 56 79 1034 68 88
1029 64 50 1009 72 60 1012 64 69 1036 88 79 1010 84 88
1042 44 50 1001 32 60 1049 56 71 1008 60 80 1050 80 92

§４．１２：英語順・QuickSort・降順

[seiseki.dat]の内容を、英語順に Quick Sort で降順ソートしました。
2002 回手順を踏みました。
ID:学生証番号 MT:数学 EN:英語
1..10_____11..20_____21..30_____31..40_____41..50____
ID MT EN ID MT EN ID MT EN ID MT EN ID MT EN
1050 80 92 1008 60 80 1028 72 71 1001 32 60 1042 44 50
1034 68 88 1030 68 79 1025 44 69 1009 72 60 1029 64 50
1010 84 88 1041 56 79 1012 64 69 1003 32 59 1031 16 49
1023 64 86 1036 88 79 1018 68 68 1005 76 58 1040 52 44
1033 68 83 1027 64 79 1035 60 68 1004 60 57 1038 32 42
1043 56 82 1039 80 76 1014 64 67 1021 84 54 1046 12 39
1007 64 81 1011 68 74 1019 76 67 1006 24 53 1026 16 39
1022 80 81 1048 64 74 1045 80 67 1013 64 53 1047 40 32
1016 76 81 1032 68 73 1024 60 64 1037 20 51 1002 32 29
1015 56 81 1049 56 71 1020 64 61 1044 44 51 1017 40 28

7JFC1121 佐藤圭一

§５：各手順の比較

§５．１：測定に使用したコンピューターのスペック

Factor:IBM PC/AT Compatible		OS:MS-Windows98	
Compiler:Borland Turbo PASCAL 5.5			
Either CMOS-RTC or 14.318MHz clock seems strange.			
factor : :1.4271			
L1 cache size 16KB	L2 cache size 64KB	L3 cache size 96KB	L4 cache size 128KB
MMX:　CELERON-A [GenuineIntel Fam6 Mdl6 Stp 0]			

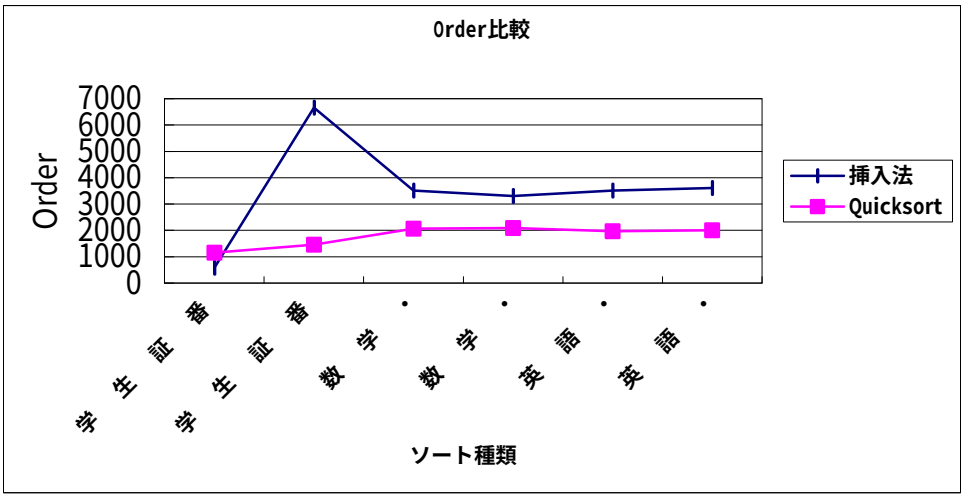
375.87[MHz]	[ns/dword]	[CPUclocks]
E-cache read	12.349	4.642
E-cache write	19.118	7.186
Main read	12.340	4.638
Main write	26.443	9.939

From pfm686

§５．２：Order での比較

SORT_O.PAS の実行結果より

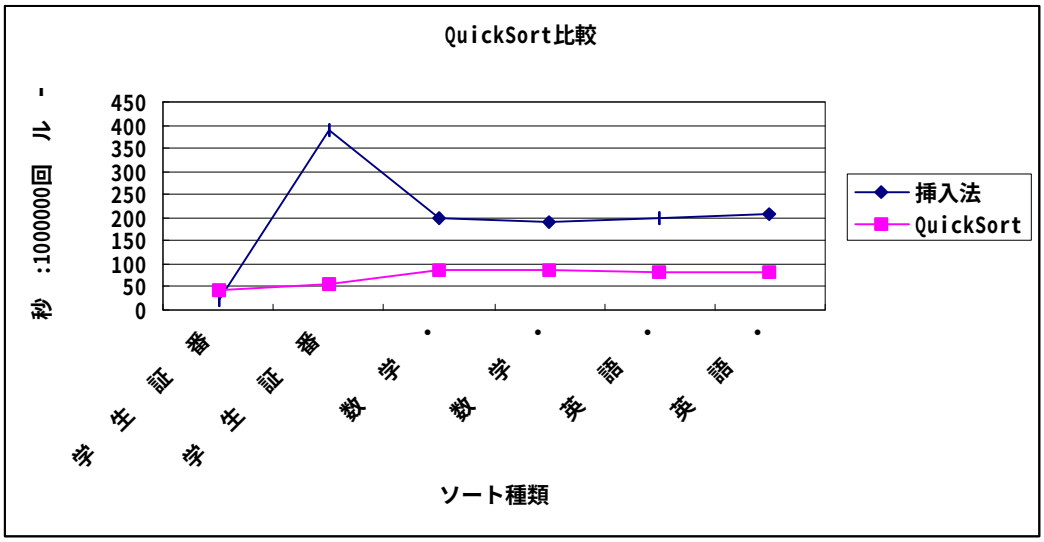
	挿入法・昇順	挿入法・降順	QuickSort・昇順	QuickSort・降順
学生証番号	588	6664	1160	1454
数学	3518	3309	2068	2090
英語	3513	3614	1974	2002



§５．３：演算時間での比較

STOP_W.BAT の実行結果より

	挿入法・昇順	挿入法・降順	QuickSort・昇順	QuickSort・降順
学生証番号	23	388	43	56
数学	199	190	86	88
英語	199	208	83	84



§６：結果分析

- QuickSort の方が、挿入法よりも手順・実行時間共に少なくソート出来る事が分かった。
- が、極度に整列されたデータでは、逆転現象が起こることが確認された。

7JFC1121 佐藤圭一

§７：考察

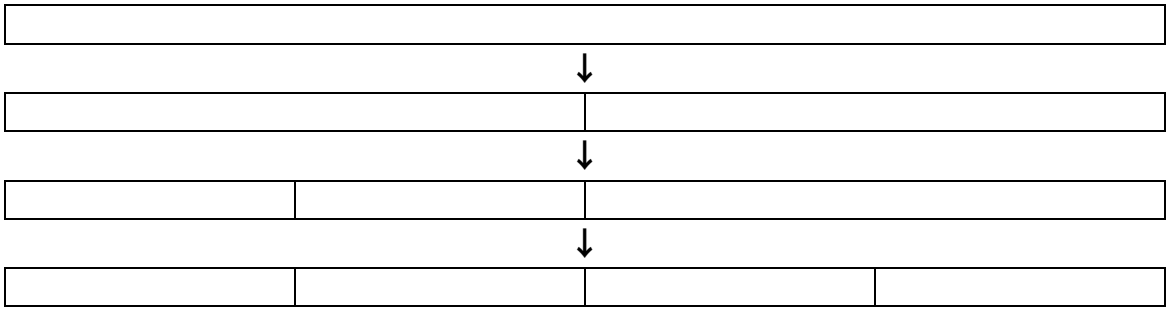
§7．1：全体的な考察

結果的には、QuickSort の方がより優れたソート法であることが確認されたが、それは演算回数面のみの比較でしかなく、メモリ領域等、リソースをどれだけ消費しているかを考慮にいれた場合、「どちらが優れている」とは一概には言えない。

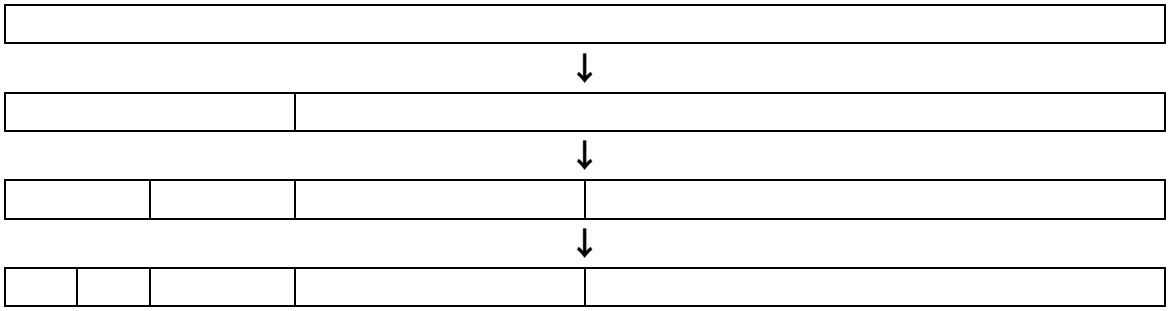
§7.2：QuickSort 基準値の位置と計算量

QuickSort では、基準値の位置を中央に取るのが基本となっているが、何故であろうか？

基準値を中央に取った場合、下図の様に演算されていく。



もし、基準値が左右どちらかにズれていたら…



この様に、計算量が増えてしまう。

何故なら、左右均等にみていく事により、「左」「右」の要素が均等に振り分けられ、左右がバランス良く演算処理されていくからである。

よって、QuickSort では、基準値を中央に取るのであろう。

§８：感想

order の解析方法を、力任せにやってしまった。本来なら実行形式ファイルを逆アセンブルすべきだと思う。

2 つのプログラムで、共通の procedure は、uses 命令にてまとめた方がきれいにまとまったと思う。