パソコンによるデータソート

——データ構造が処理時間に及ぼす影響——

久保忠延

1 緒言

最近のパーソナルコンピュータ（パソコン）の普及はめざましく，これまでコンピュータとは無縁と考えていた人々も身近にパソコンを置いて活用するようになりつつある。

パソコンはハード面では記憶容量の増大と処理速度の高速化が計られ，年々その性能が向上している。パソコンの性能に関しては，ハード面の改良とともに，ソフト面での改善も極めて重要である。問題処理のプログラムを作成する際，どのようなアルゴリズムを用いるかによって処理速度や計算精度が大きく変化することが多い。

大量のデータを取り扱い，それらを分類する必要性はいろいろなところで生ずるが，この分類（ソート）のアルゴリズムにはこれまで多くのものが考えられている。ソートの対象を数値に限定し，それを大きい順（又は小さい順）に並べるアルゴリズムについてもいろいろ提案され，それによる処理時間についても検討されている[1][2]。しかし，これらの場合の処理時間の測定にはソートの対象となる数値データとしてパソコンにより発生させた乱数そのもの（あるいはそれを単に整数化したもの）が用いられており，人為的に作成された構造をもつデータについてそのソートに要する時間とアルゴリズムの関係を詳しく調べたものは見当らない。

そこで本研究では，代表的なソートアルゴリズムについて，特徴的な4種の構造をもつ数値データを与え処理時間を調べ，データ構造が処理時間に及ぼす影響を明らかにした。

2 ソートプログラムとデータ構造

2.1 ソートプログラム

ソートのアルゴリズムとして知られているものは数多くあるが，主なものとしてはつぎのようなものをあげることができる。

(1) 逐次決定ソート（最小値選択法，択一決定ソート）
(2) シェルソート
(3) ヒープソート
(4) クイックソート
(5) 度数ソート
(6) バブルソート（隣接交換法）

本研究では(1)～(5)のアルゴリズムについて検討を行った。使用したプログラムのリストをリスト1からリスト5に示した。

```
1000 *MINIMUM
1010 FOR J=1 TO NN-1
1020 FOR K=J+1 TO NN
1030 IF T(J)<=T(K) THEN 1050
1040 SWAP T(J),T(K)
1050 NEXT K
1060 NEXT J
1070 RETURN
```

リスト1 逐次決定ソート

```
2000 *SHELL
2010 D=NN
2020 D=INT(D/2)
2030 IF D<1 THEN 2130
2040 DD=NN-D
2050 FOR K=1 TO DD
2060 J=K
2070 IF T(J)<=T(J+D) THEN 2110
2080 SWAP T(J),T(J+D)
2090 J=J-D
2100 IF J>=1 THEN 2070
2110 NEXT K
2120 GOTO 2020
2130 RETURN
```

リスト2　シェルソート

```
3000 *HEAP
3010 FOR I=2 TO NN
3020 J=I
3030 D=INT(J/2)
3040 IF T(D)>=T(J) THEN 3080
3050 SWAP T(D),T(J)
3060 J=D
3070 IF J>1 THEN 3030
3080 NEXT I
3090 I=NN
3100 FOR K=NN TO 1 STEP -1
3110 SWAP T(1),T(I)
3120 I=I-1
3130 J=1
3140 S=2
3150 IF S>I THEN 3230
3160 IF S=I THEN 3180
3170 IF T(S+1)>T(S) THEN S=S+1
3180 IF T(J)>=T(S) THEN 3230
3190 SWAP T(J),T(S)
3200 J=S
3210 S=J*2
3220 GOTO 3150
3230 NEXT K
3240 RETURN
```

リスト3　ヒープソート

```
4000 *QUICK
4010 K=0:L=1:R=NN
4020 IF L>R THEN 4210
4030 I=L
4040 J=R+1
4050 Q=T(L)
4060 J=J-1
4070 IF I>=J THEN 4150
4080 IF Q<=T(J) THEN 4060
4090 T(I)=T(J)
4100 I=I+1
4110 IF I>=J THEN 4150
4120 IF T(I)<=Q THEN 4100
4130 T(J)=T(I)
4140 GOTO 4060
4150 T(J)=Q
4160 K=K+1
4170 LS(K)=J+1
4180 RS(K)=R
4190 R=J-1
4200 GOTO 4020
4210 IF K=0 THEN 4260
4220 L=LS(K)
4230 R=RS(K)
4240 K=K-1
4250 GOTO 4020
4260 RETURN
```

リスト4　クイックソート

```
5000 *DOSU
5010 FOR I=1 TO NN
5020 K=T(I):Z(K)=Z(K)+1
5030 NEXT I
5040 P=1
5050 FOR J=0 TO MM
5060 IF Z(J)=0 THEN 5080
5070 JJ=Z(J):Z(J)=P:P=P+JJ
5080 NEXT J
5090 FOR I=1 TO NN
5100 W=Z(T(I))
5110 IF S(W)=0 THEN 5120 ELSE 5130
5120 S(W)=I:U(W)=W+1:GOTO 5140
5130 S(U(W))=I:U(W)=U(W)+1
5140 NEXT I
5150 RETURN
```

リスト5　度数ソート

2.2　データ構造

データ構造としてつぎの4種類のものを与えて処理を行わせた。

(1)　乱数データ

(2)　正規乱数データ

(3)　順データ

(4)　逆順データ

乱数データ(1)は，パソコンにより発生させた乱数を整数化（0～100の値に）したものである。正規乱数(2)は Box-Mueller 法[3][4] により発生させた乱数を整数化したものであり，本研究の場合は平均値50，標準偏差15としたときの数値をデータとして与えた。ただし，まれに負のデータが出現するため，この場合はその値を0で置き換えてデータとした。順データ(3)は，すでに並べ換えを要しないように分類されたデータ配列をなすもの

で，ソート法の種類に応じて小さい順または大きい順に数値が並んだものである。逆順データ(4)は，(3)の順データのデータ構造の数値の並び方を逆にしたものである。

3　結果と考察

前項2.2で述べた(1)～(4)の構造をもつデータを

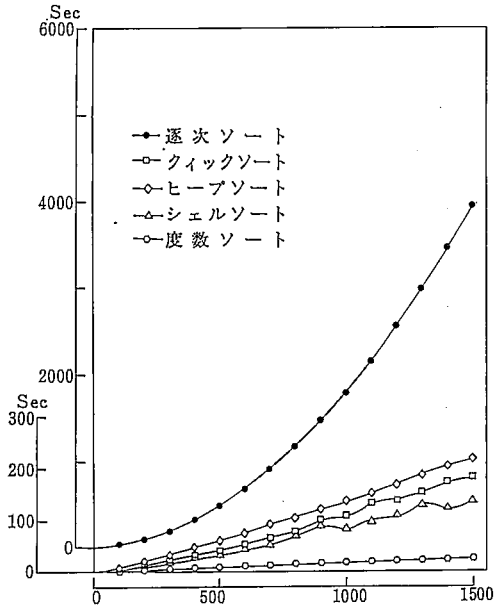

図1　乱数データの処理時間

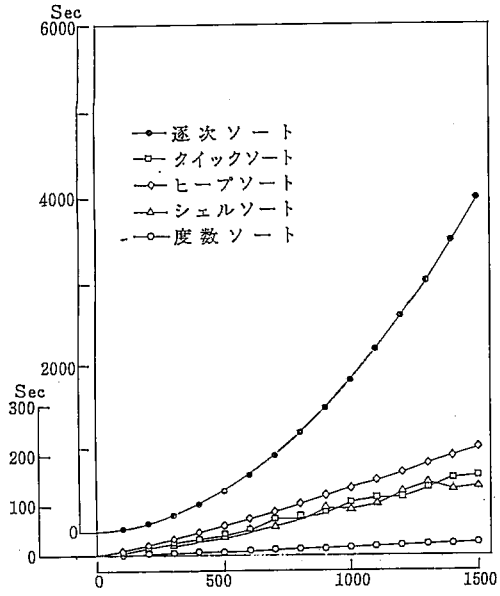

図2　正規乱数データの処理時間

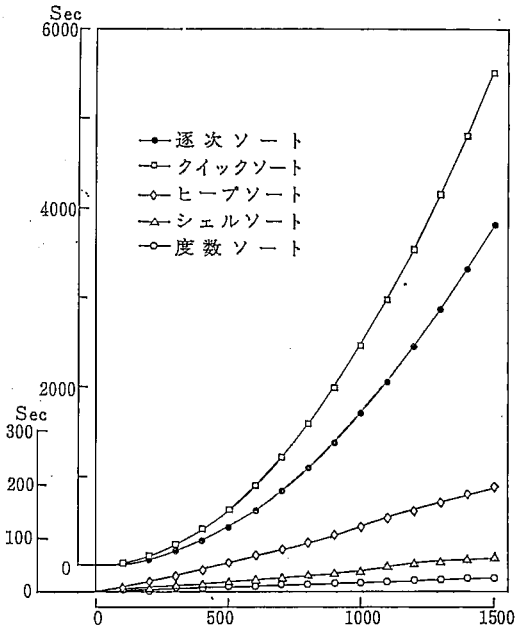

図3　順データの処理時間

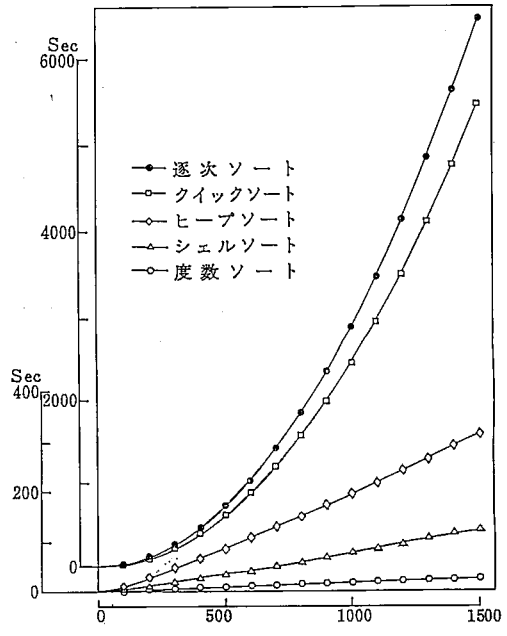

図4　逆順データの処理時間

各ソート法により並べ換え処理を行わせたときに要した時間をグラフにして比較したものを図１～図４に示した。

縦軸に処理時間を秒単位で示し，横軸にデータ個数をとって示してある。ソート法により処理時間が著しく相異するため縦軸は尺度の異る２種の目盛を使っている（後出の図８についても同様）。尚，使用したパーソナルコンピュータはPC—9801である。

図１は乱数データを，図２は正規乱数データを与えた場合の処理時間であるが，両者とも類似の傾向を示している。すなわち，ソートアルゴリズムの最も単純な逐次決定ソート法（以下逐次ソートと略す）では，データ数の増加とともに急激に処理時間が増加し，２次曲線的なカーブを描いている。これに比べ他の４つのソート法ではデータ数の増加に対して処理時間はほぼ直線的に増加している。処理時間は度数ソート，シェルソート，クイックソート，ヒープソート，逐次ソートの順に短い。

図３，図４に順データおよび逆順データを与えた場合の結果を示した。図１，図２に比べ，ソートに要する時間の多少の順位が入れ代っており，順データの場合は度数ソート，シェルソート，ヒープソート，逐次ソート，クイックソートの順に，また逆順データの場合は度数ソート，シェルソート，ヒープソート，クイックソート，逐次ソートの順に処理時間が短い。

図５～図８に各ソート法の各々に対して構造の異ったデータを与えて処理したときの時間をグラフにしたものを示した。

図５に示した逐次ソート法の場合は，いずれのデータ構造のデータに対してもデータ数の増加に伴って同様の処理時間の増加傾向を示している。逆順データの場合最も長い処理時間を要し，順データの場合最も処理時間が短い。乱数データ，正規乱数データの処理時間の間にはほとんど差異がなく，これらは順データの処理時間に近い。

図６にシェルソート法による処理時間を示したが，この場合には順データ，逆順データ，正規乱数データ，乱数データの順に処理時間が増加している。シェルソート法の場合，グラフ上のプロッ

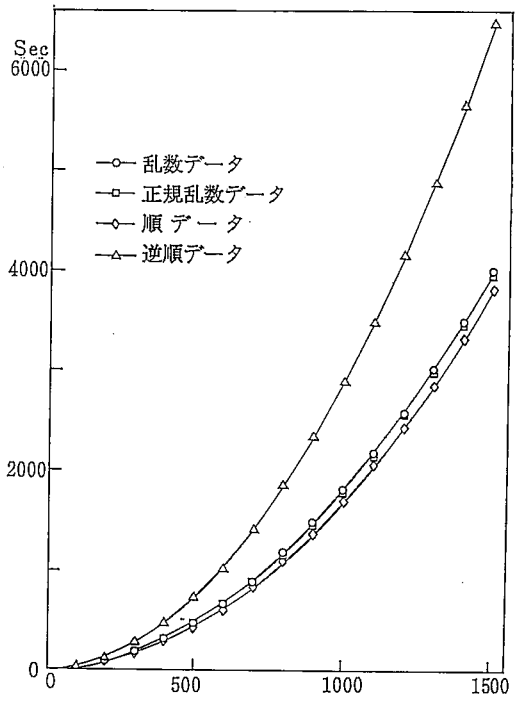

図５　逐次ソート法による処理時間

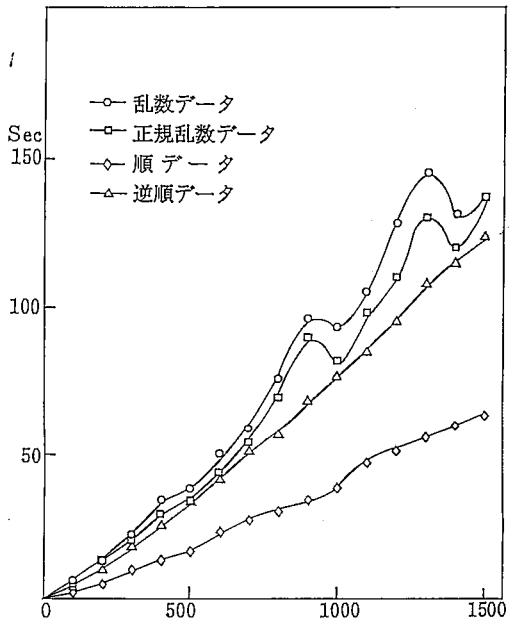

図６　シェルソート法による処理時間

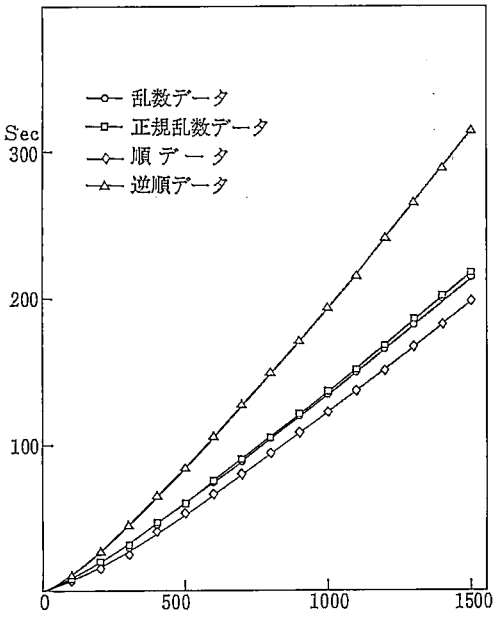

図7　ヒープソート法による処理時間

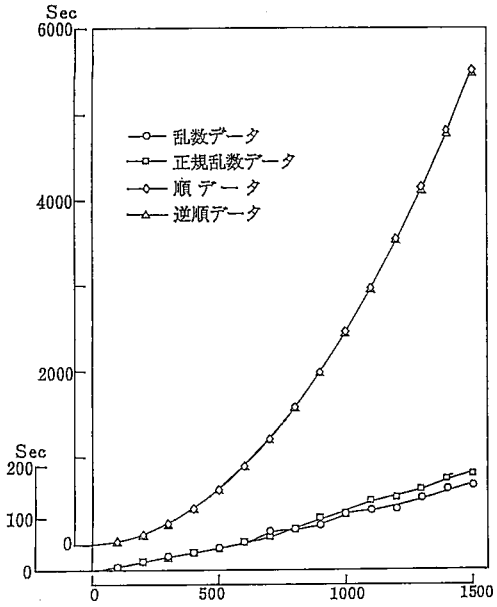

図8　クイックソート法による処理時間

ト点がばらついており，データ数が増加しているのに逆に処理時間が減少した箇所が見られる。これはシェルソートのアルゴリズムがデータをある間隔ごとに区切って比較を行うため，その区切りの状態にデータ個数が影響を与えるためである。

ヒープソート法による処理時間を図7に示した。ヒープソート法では，逆順データの処理に最も多くの時間を要し，順データの処理が最も短時間で済んでいる。乱数データと正規乱数データの間にも，差異は小さいが，はっきりした時間差の傾向が見られ，正規乱数データの方が処理時間が多くかかっている。

図8にクイックソート法による処理時間を示したが，乱数データ，正規乱数データの処理に比べ順データと逆順データの処理時間が極端に長くなっている点が特徴的である。順データ，逆順データの処理にはほとんど同じ時間を要している。乱数データと正規乱数データの処理では，データ数の多いところで差異が見られる。

度数ソートについては，そのアルゴリズムから当然のことながら，データ構造の違いはその処理時間にほとんど影響を及ぼさなかったので図は省略した。他の4つのアルゴリズムに比べ度数ソートはいずれのデータ構造に対しても最も処理時間が短く優れているように見えるが，ソート対象の数値データが正の整数に限定されるのが欠点である。

4　結　言

本研究では，代表的なソートアルゴリズムを使って数値データを並べ換える処理を行うのに要する時間を実験的に調べた。処理対象のデータとしては，従来一般的に用いられている乱数データの他に特徴的な構造をもつデータ（正規乱数データ，順データ，逆順データ）を用いて実験を行った。その結果，乱数データを処理させただけでは見出せなかったソートアルゴリズムのいくつかの興味深い特性を知ることができた。実験に使用した5つのアルゴリズムのうち，度数ソート以外のもの（逐次ソート，シェルソート，ヒープソート，クイックソート）ではデータ構造が処理時間に大きな影響を及ぼし，その影響の及ぼし方はそれぞれのアルゴリズムに対して異ることがわかっ

た。

本研究により得られた結果は，ソート対象となるデータの性質がある程度予測できる場合，いずれのソートアルゴリズムを利用すれば良いのかの判断基準を与え得るものと考えられる。

文　献

(1) 大河内・古賀・銀島：基礎電子計算機，実教出版 124（1982）
(2) 涌井良幸：高速ソートプログラミング，誠文堂新光社 10（1984）
(3) 磯田・大野：数値計算ハンドブック，オーム社 671（1980）
(4) 石原辰雄：BASICによる統計，共立出版 79（1984）