Shellの分類法の一改良*

池野信一**　保坂　務**

1. はしがき

情報処理の分野における分類とは，順序集合に属する有限個の元を順序どおりに並べかえることであって，きわめて重要な地位を占めている．与えられたデータの間に定義される順序関係としては，たとえば単語の集合におけるアルファベットとか，電話料金計算における加入者番号などの例があげられる．電子計算機内で取り扱われる数値はすべて実数であるから，このような順序関係は実数の大小関係として取り扱われる．

計算機の主記憶装置内で行なわれる分類を特に区別して内部分類と呼ぶ．実際の分類操作は一般に主記憶装置だけでは容量が不足であるから，外部記憶装置を使用して，これらの間でデータの出し入れを行ないながら分類操作を行なうことになる．したがって，このような内外記憶装置の運用法ということもきわめて重要な問題となる．一方分類の方法という点から見れば，逐次的な外部記憶装置に使用できる分類の方法はごく限られている．それに比べて内部分類の場合には，主記憶装置がランダム・アクセスであり，プログラミングによって操作を行なわせるために種々な方法が可能となる[1]．現在までに得られている方法のうちでも，Shell[2]の考案した方法が非常にすぐれていて，特に項目数が少ない範囲で最も高速な方法とされている．そこで筆者等はこの方法をもとにして，並べかえにおける比較回数・転送回数に近似的な考察を加え，その結果改良の方法を得ることができたのでここにご報告する．

2. Shellの交換法

Shellの方法は，交換法と呼ばれる方法の一種の拡張である．まず交換法の説明を行なう．以下において，元はすべて互に異なる実数とし，並べかえは小さい順に行なうものとする．いま，n個の元からなる集合内のr個の元が，

$$a_1<a_2<\cdots\cdots<a_r,\ (r<n)$$

のように順序づけられているとしよう．集合の残りの部分から新しい元c_{r+1}を取り出してきて，まずa_rと比較を行なう．もし$c_{r+1}<a_r$ならば，さらにa_{r-1}と比較を行なう．このようにして比較をくり返してゆき，はじめて$a_j<c_{r+1}$となるa_jが見つけられるとc_{r+1}の入る位置が

$$a_1<a_2<\cdots\cdots a_j<c_{r+1}<a_{j+1}\cdots\cdots<a_r$$

のように定められることになる．この列にあらためて端から順に$a_1, a_2, \cdots\cdots, a_{r+1}$と名前をつけかえる．このような1個の元を列の中に入れる操作をくり返し用いることによってn個の元のすべてを順序づけることができる．すなわち，一番最初に取り出された元c_1をa_1とし，以下$r=1$から$r=n-1$について上に述べた手順をくり返せばよい．さて，このような並べかえの過程を考えてみると，明らかにこの方法は元c_{r+1}とあるaの元を比較して順序が逆ならばこの2元の位置を入れかえるという基本の操作のくり返しだけで行なうことができる．この操作を交換と呼び，したがって，この方法は交換法と呼ばれる．

交換法は元の数nが小さいときには十分有効であるが，nが大となると，きわめてその手数が増えてしまう．Shell[2]はこの交換法を拡張して次のような方法を考案した．すなわち，適当なパスp_iを定めてp_iだけ離れた元ごとに交換法によって順序づけを行ない，かつ

$$p_{i+1}=[p_i/2]^{***},\ p_0=n \qquad (1)$$

によってp_iを1に等しくなるまでしだいに減少させてゆく方法である．すなわち，まず

$$p_1=[n/2]$$

として1番目の元c_1からはじめて各々p_1だけ離れた元同志に着目して，これを順序どおりに並べかえる．この並べかえには上述した交換法が用いられる．次に2番目の元からはじめて，やはりp_1だけ離れた

---

* A Revised Shell's Sorting Procedure, by Nobuichi Ikeno and Tsutomu Hosaka (Electrical Communication Laboratory, Tokyo)

** 電気通信研究所

*** 整数部分

元同志について同様に交換法で並べかえを行なう．このようにしてすべて p_1 だけ離れた元同志が順序どおりに並べられると，次に $p_2=[p_1/2]$ として今度は p_2 について同様に p_2 だけ離れた元同志をすべて順序どおりに並べる．このようにして p_i を減らしてゆき最後に $p_i=1$ として並べかえを行なうとすべての元が順序づけられたことになる．

Shell の方法はきわめてすぐれた方法であるが，元の数 n が2の累乗であるときには（1）式よりわかるように，$2p_{i+1}=p_i$ となってきわめて手数が増えてしまうのである[1]．このような矛盾を避けるためには，まず p_i を奇数に選ぶことが考えられる．さらに Frank および Lazarus[3] は p_i として

$$\begin{cases} p_{i+1}=2[p_i/4]+1, & p_i\leqq 15 \quad (2\cdot a) \\ p_{i+1}=2[p_i/8]+1, & p_i>15 \quad (2\cdot b) \end{cases}$$

のようにとるのが良いとした．実際 p_i をこのようにとれば2の累乗の場合の問題は完全に避けられ，その上に平均して比較の手数は Shell の方法よりもかなり少なくなる．この方法* の特色は，第1に大部分について（2・b）からわかるように，$p_{i+1}\simeq p_i/4$ としたことである．その他に（2）式を一見して理解できることは，この式はきわめて巧妙に作られていて，2進計算機においてはシフト演算を用いて簡単に計算できることであろう．しかし10進計算機が使用される場合には2の累乗による乗除算が特に有利とはならないから，さらにもっと適当な係数があればこれを採用した方がよい．

3. 交換の回数

2元の位置を入れかえる操作が交換である．近似的にではあるが Shell の方法における交換の回数について考察しよう．

まず交換法の交換の回数は次のようにして求められる．いま n 個の元からなる数列が与えられたとき，$i<j$ なるすべての場合について i 番目の元 c_i と j 番目の元 c_j を取り出し，その大小関係を表わす量 r_{ij} を $r_{ij}=0(c_i<c_j)$，$r_{ij}=1(c_i>c_j)$ と定める．そのとき

$$E=\sum_{i<j} r_{ij} \qquad (3)$$

を作れば，これは与えられた数列の中にある逆順関係の総数となる．いま交換法を考えるとき，常に隣接した2個の元のみが比較・交換の操作に関与する．したがって1回の交換によって逆順関係は1個だけ減少するが，他に新しく逆順を生じたり，減少させたりすることがない．また並べかえの完了とは明らかに逆順の数がゼロとなることであるから，並べかえの完了までに必要な交換回数は（3）式で与えられることになる．（3）式の r_{ij} を $c_i>c_j$ となる確率で置きかえれば，交換回数の期待値が得られる．

次に準備として以下の計算を行なっておく．

2組の数列 $\{a_i\}$，$\{b_j\}$ が各々大きさの順に並べられていて，

$$a_1<a_2<\cdots\cdots<a_i<\cdots\cdots a_r$$
$$b_1<b_2<\cdots\cdots<b_j<\cdots\cdots b_s \qquad (4)$$

であるとする．両方の数列から任意の元 a_i，b_j を取り出して比較を行ない $a_i<b_j$ となる確率を $P(i,j)$ とかく．

$$a_{r-t}<b_j<a_{r-t+1} \qquad (0\leqq t\leqq r-i)$$

となる場合の数は

$$\binom{s-j+t}{t}\cdot\binom{j-1+r-t}{r-t}$$

であるから

$$P(i,j)=\frac{1}{\binom{r+s}{s}}\sum_{t=0}^{r-i}\binom{s-j+t}{t}\cdot\binom{j-1+r-t}{j-1} \qquad (5)$$

となる．ここで対称性から，$r=s$，$i=j$ ならば当然（5）式は 1/2 である．次に $r=s$ とおき，2列の相互的な逆順確率の総和として次の

$$E_m=\sum_{i<j}\{1-P(i,j)\} \qquad (6)$$

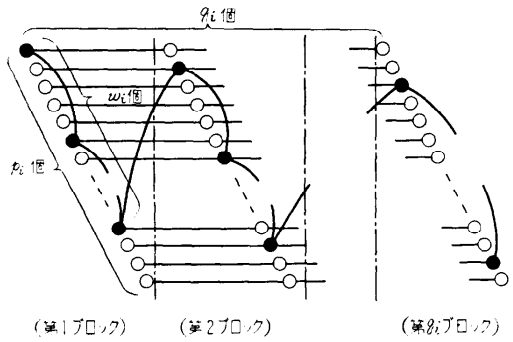
を作れば，計算の過程は省略するが

$$E_m\simeq\frac{s}{2}\cdot\frac{s(s-1)}{(s+2)(s+1)}\simeq\frac{s}{2} \qquad (7)$$

となる．ただし $P(i,j)\simeq P(1,j-i+1)$ とした．これらの関係を用いて交換の回数が求められる．**第1図**のように q_i 個の元からなる p_i 組の順序づけられた元の列を考える．第 k 番目の元の集合を第 k ブロックと呼ぶ．これは Shell の方法でパス p_i を用いて並べかえを終った状態を表わすものとする．次に $p_{i+1}\simeq p_i/w_i$ で並べかえを行なうときの交換回数を求めてみる．この場合簡単のため各ブロックから，ちょうど w_i 個ずつの元がとり出されると仮定しよう．また，そのとき p_{i-1} 以前のパスの影響を無視する．そのとき交換回数 N_s は逆順確率の総和を求めればよいことから，次のように計算される．

（1） 同一ブロック内では二元の逆順確率は 1/2 で

* 一般に改訂 Shell の方法と呼ばれるが，本稿では Frank & Lazarus の方法と呼ぶ．



第1図　Shell の方法

q_i 個の順序づけられた元からなる列の，p_i 個の組において各ブロックから，ちょうど w_i 個ずつの元がとり出されるという理想的な場合を考えて，Shell の方法を近似する．

あるとみなしてその総和は，

$$N_s^{(a)}=\frac{1}{2}\binom{w_i}{2}\cdot q_i\cdot\frac{p_i}{w_i}=\frac{1}{4}n(w_i-1) \qquad (8)$$

（2） 他ブロック間では，場合を二つに分けて

（a） p_i と p_{i+1} の最小公倍数を l とするとき，$l\geqq n$ ならば第1図で抜き出された黒丸の元は再び p_i に関する同一の連糸にもどらない．これを交叉がないと呼ぶ．すなわち，二元の逆順確率がゼロとなる場合は一つも含まれない．このときある二つの異ったブロックからとられる二元の関係は合計 w_i^2 個あり，一方すべての関係を一通りずつとって加えたものが（6）式の E_m に相当するから，

$$N_s^{(b)}=w_i^2E_m(q_i)\cdot\frac{p_i}{w_i}=w_{ii}pE_m(q_i) \qquad (9)$$

がえられる．

（b） 交叉が最も多くなるのは $p_i=w_i$ の場合で，このときは逆順確率がゼロとなる場合を除くと

$$N_s^{(c)}=(w_i-1)\cdot p_iE_m(q_i) \qquad (10)$$

がえられる．

したがって p_{i+1} で並べかえを行なう場合の $N_s^{(i)}$ は，並べかえの前半および後半について各々次式で近似されることになる．

$$\begin{cases}N_{s(1)}^{(i)}=N_s^{(a)}+N_s^{(b)}\\ N_{s(2)}^{(i)}=N_s^{(a)}+N_s^{(c)}\end{cases} \qquad (11)$$

この式を求めるために，いくつもの近似を重ねたので，これをまとめると，

（1） E_m が計算できないために近似を行なったこと．

（2） 第1図のように理想的な場合のみを考えたこと．

（3） p_{i-1} 以前のパスの影響を無視したこと．

（4） 交叉の有無に関する仮定．

の4点となる．実験値とどの程度一致するかを**第1表**に示した．

第1表　交換の回数（$n=995$ の例）

p_i	249	63	15	7	3	1
実験値	761	1,576	2,510	1,760	971	822
$N_{s(1)}^{(i)}$	746	1,521	2,436	1,284	1,464	
$N_{s(2)}^{(i)}$				818	982	1,466

4. 比較の回数

比較の回数 N_c について考察するが，本章の議論もまた近似的なものである．

新しい元 c_{r+1} を取り出してきて，すでにでき上った列の末端の元 a_r と行なう比較を1次比較と呼ぶ．1次比較回数は同時に最低必要な比較回数 N_{c0} に等しい．1次比較の結果，交換が生ずれば，一般に再び比較が必要となる．このようにして必要ならば2次，3次，……と比較がくり返される．したがって，さらに N_s 回だけ比較が必要となる．ここで列の頂点で交換が生じた場合は比較は不要となるが，Shell の方法では新しい元 c_{r+1} はあまり遠くまで移動しないから，これを無視することができる．したがってパス p_{i+1} について必要な比較回数は

$$N_c^{(i)}\simeq N_{c0}^{(i)}+N_s^{(i)} \qquad (12)$$

となる．いま係数 $w_i=w$（一定）とおくと

$$p_i\simeq n\cdot w^{-i}(i=1,2,\cdots\cdots,v,\ v=[\log_w n]) \qquad (13)$$

$$N_c\simeq\sum_{i=0}^{v-1}(N_{c0}^{(i)}+N_s^{(i)}) \qquad (14)$$

となる．w を増加させるとき N_s は単調増大するが，v は単調に減少して適当な w に対して N_c は最小値をとる．これを求めてみよう．計算の便宜上二つの場合に分けて考える．

（1） まず $p_i>\sqrt{n}$ である範囲については，p_i と p_{i+1} の交叉がないと仮定できる．さらに E_m についても場合を二つに分けて，まず $E_m\simeq 0$ とおくと，

$$N_c\simeq\sum_{i=0}^{v-1}\left\{\frac{1}{4}n(w-1)+n\left(1-\frac{1}{w^{(i+1)}}\right)\right\}$$

$$\simeq\frac{1}{4}n(w+3)\cdot v-\frac{n-1}{w-1} \qquad (15)$$

となるが，ここで

$$\frac{dN_c}{dw}=0 \qquad (16)$$

とおけば，$v\simeq\log_w n$ として

$$\ln w-\frac{1}{w}(w+3)+\frac{1}{\ln n}\cdot\frac{4(\ln w)^2}{(w-1)^2}=0 \qquad (17)$$

がえられる．$n\to\infty$ とすれば，この解は

$$w_\alpha=4.95 \qquad (18)$$

となる．すなわち以上の仮定のもとでは w の最適値は 4.95 となる．

次に q_i が大きなときは（7）式によって $E_m\simeq q_i/2$ とみなせるから，上と全く同様な計算を行なうと同様に n を大として，

$$w_\beta=3.6 \qquad (19)$$

という解がえられる．したがって $p_i>\sqrt{n}$ では w の最適値は 4.95 ないし 3.6 の間にあることになる．

（2） 次に $p_i\leqq\sqrt{n}$ では並べかえの後半となり第1図の q_i が長くなるから $E_m\simeq q_i/2$ とみなされて，最適値は w_β となる．また，交叉がきわめて多くなった極限として $N_{s(2)}{}^{(i)}$ を用いた場合には，やはり（15）（16）（17）とほぼ同様な計算を行なうと，同じく n を大として，

$$w_\gamma=3.0 \qquad (20)$$

という解がえられる．したがって $p_i\leqq\sqrt{n}$ では w の最適値は 3.6 ないし 3.0 の間にあることになる．

前述したように Shell の方法は常に $w=2$ ととるものであるが，それよりも Frank & Lazarus のように $w=4$ とする方が一層最適値に近いことになる．参考のために（15）式およびそれと同様な w_β, w_γ の式から N_c の概算を行なうと**第2表**のようになる．これから Frank & Lazarus の方法はきわめて良い近似であるということができる．

第2表 比較回数の比

		Shell	Frank & Lazarus	最適値
		$w=2$	$w=4$	$w=w_\alpha, w_\beta, w_\gamma$
前半	w_α の式	146%	102%	100%
後半	w_β 〃	121	101	100
	w_γ 〃	112	104	100

並べかえの前半では w_α ないし w_β の式，後半では w_β ないし w_γ の式がなり立つものとして，二つの方法について比較回数を表わしたもの．

5. 改良の方法

これまでの議論を押し進めて，さらに最適に近い方法を作ってみよう．係数 w_i の最適値は交叉の有無によって影響される．p_i と p_{i+1} の避け難い交叉が生ずる範囲は，これらが互に素にとられているものとすれば $p_i\cdot p_{i+1}<n$ となる．すなわち $p_i\simeq\sqrt{n}$ に対して w_i の分岐点がある．

いま，あらためて次のような方法を考えてみる．係数 w_i は毎回変化してゆき，最初は w_α であるが最後に w_γ となる．しかも分岐点である $p_i=\sqrt{n}$ に対して n によらない値をとるようなものである．たとえば次のようにおくと，ほぼこの条件が満される．

$$w_i=w_\gamma+(w_\alpha-w_\gamma)\cdot x,\ x=\frac{\ln p_i}{\ln n} \qquad (21)$$

$p_i=\sqrt{n}$ に対して $w_i=\dfrac{w_\alpha+w_\gamma}{2}$ となっている．

これから

$$\frac{d\ln p_i}{di}=\Delta\ln p_i=-\ln w_i$$

を用いて

$$\frac{dw_i}{\ln w_i}=-\frac{(w_\alpha-w_\gamma)}{\ln n}di \qquad (22)$$

の関係がえられる．積分を実行して左辺の級数のうち最大項 $\ln w_i$ をとって

$$w_i=\exp\left\{-\frac{(w_\alpha-w_\gamma)}{\ln n}\cdot i+c\right\} \qquad (23)$$

を得る．すなわち

$$w_i=w_\alpha/u^i,\ \ln u=\frac{c_0}{\ln n} \qquad (24)$$

の形の計算式が得られる．定数 c_0 を v 回目に $w_i=w_\gamma$, $p_v=1$ となるように定めて，

$$\left.\begin{aligned} &v\simeq\ln n/\ln\bar{w},\ \bar{w}=\sqrt{w_\alpha w_\gamma}\\ &\ln u\simeq\frac{\ln^2 w_\alpha-\ln^2 w_\gamma}{2\ln n}=\frac{c_1}{\ln n}\end{aligned}\right\} \qquad (25)$$

を得る．すなわちこの方法は毎回先に使用した w_{i-1} を定数 u で割って w_i を作り，この w_i で p_i を割って p_{i+1} を作るのである．w_i は等比的に変化するのでこれを係数変化法と呼ぶことにする．

係数変化法を用いてパスを計算する関数形を与えるには交叉の問題を考えねばならない．いま1に等しくない2種のパス p_i, p_j の最小公倍数を $l(p_i, p_j)$ とかくとき

$$l(p_i, p_j)<n$$

ならば p_i と p_j は交叉しているという．このとき l だけ離れた元同志は，二重に並べかえをうける恐れがある．特に p_{i+1} が p_i を割り切るとき，すなわち

$$p_i\equiv 0\ (\mathrm{mod}\ p_{i+1}),\ p_{i+1}\neq 1$$

を直接交叉と呼ぶ．直接交叉は Shell の方法で n が2の累乗のときに現れる．Frank & Lazarus の方法では p_i を奇数にとって直接交叉を避けているが（2・b）のようにとるとこれだけでは（15→3），（9→3）

なる2種の組み合わせが残る．そのために $p_i \leqq 15$ の場合を（2・a）のように分離する必要が生じたのである．いま w_i が変化してゆく場合を考えて，Frank & Lazarusにならって直接交叉を避けるという条件で計算式を作ってみる．ここで $w_i<5$ であるから直接交叉を除く条件は

$$p_i \not\equiv 0 \pmod{4, 3, 2}$$

となる．これからただちに

$$p_i \equiv 1 \text{ or } 5 \pmod 6$$

が得られる．

以上の考察をもとにして次のような計算式が作られるが，この場合にも $p_i \leqq 7$ を分離せざるを得ない．

（係数変化法）

$$\left.\begin{array}{l} p_i>7 \\ \quad \dfrac{p_i}{6\,w_i}=\alpha_i p_i=[\alpha_i p_i]+r_i,\ p_0=n \\ \quad r_i \geqq 1/2,\ p_{i+1}=6\,[\alpha_i p_i]+5 \\ \quad r_i<1/2,\ p_{i+1}=6\,[\alpha_i p_i]+1 \\ p_i \leqq 7 \\ \quad p_{i+1}=2\,[p_i/4]+1 \end{array}\right\} \qquad (26)$$

ただし $w_i=w_a/u^{(i+1)}$，$\ln u=\dfrac{c_1}{\ln n}$

定数 w_a，c_1 は実験値を考慮して $w_a=5$，$c_1=1.01$ と定められた．(25) で $w_r=2.13$ としたことに相当する．u の計算式は

$$u=1+\cfrac{c_2}{d_{10}(n)-1+\dfrac{n}{n_0}}$$

$$d_{10}(n)=1+[\log_{10} n]=\log_{10} n_0$$

$$0<n/n_0<1,\ c_2=0.458 \qquad (27)$$

で近似してよい．分類を行なうにあたって定数 u は一度だけ計算すればよく，(27) からわかるように n の10進表示桁数 $d_{10}(n)$ と n の小数表示 n/n_0 とを用いて計算することができる．この方法を用いた実験例を**第3表**に示す．同一の乱数列に対して2方法を用い比較回数と転送回数を数えたものである．この表の範囲の n に対しては数%の差が認められるが，n がさらに小となるとあまり大きな差は現われなくなる．

第3表　係数変化法の実験例

元の数	Frank & Lazarus の方法		係数変化法			
n	比較回数 N_C	転送回数 N_T	N_C	%	N_T	%
982	13,510	30,787	13,044	96.7	29,796	96.9
1,353	21,081	47,406	19,734	93.5	44,777	94.5
1,765	28,728	64,540	26,918	93.8	61,283	95.1
2,000	33,125	74,290	31,028	93.7	70,491	94.8

6. 転送の方法

比較する操作とは切り離して，数を転送する操作だけについて考察を行なう．順序づけられていない n 個の元 $c_k(k=1, 2, \cdots\cdots, n)$ を順序通りに並べかえることは置換

$$\begin{pmatrix} c_1, c_2, \cdots\cdots, c_n \\ a_1, a_2, \cdots\cdots, a_n \end{pmatrix} \qquad (28)$$

を行なうことである．任意の置換は巡回置換の積で表わされ，さらに互換の積として表わされる．

実際，この置換を互換（2章で交換と呼んだもの）のくり返えしによって行なうものが交換法である．1個の交換を正直に実行すれば，元を較送する手数は4回必要である．そこで2章で述べたように，a_1，a_2，$\cdots\cdots$，a_r という順序づけられた列に c_{r+1} を入れる場合を考えると，c_{r+1} の入る場所は $(r+1)$ 個あって，これはすべて等確率でおこる．したがって，その手数は互換の数を k として

$$\sum_{k=0}^{r} \frac{1}{r+1} \cdot 4\,k = 2r \qquad (29)$$

となり，n 個のものをすべて並べるためには

$$N_T^{(1)}=\sum_{r=1}^{n-1}(2r)=n(n-1) \qquad (30)$$

だけ転送する必要があることになる．

以上の計算では互換の度に4回の転送が必要であるとしたが，一般にはある1次比較後の一連の置換は巡回置換とすることができる．すなわち移動する元

$$(a_{j+1}, a_{j+2}, \cdots\cdots, a_r, c_{r+1}) \qquad (31)$$

のすべての元を各々2ステップずつで移動するのである．この場合の較送の手数は，やはり $(r+1)$ 個の c_{r+1} が入る場所を考え，(30) 式を求めたと同様の計算を行なえば，

$$N_T^{(2)}=\frac{n}{2}(n+3)-2\sum_{i=1}^{n}\frac{1}{i} \qquad (32)$$

となる．これは巡回置換法とでも呼ぶべきもので，1次比較で交換が生じた場合に小さい方の元をレジスタに保持して進む方法である．明らかにこの方法を用いた方が (30) の方法に比べて手数が少なく，一般に広く用いられている．

しかし (28) に見られるように n 個のもののすべての大小関係を知ってから転送を行なえば各元はすべて2ステップずつで移動できる．このとき位置の移動しない元が，n 個中平均して1個あることが示されるから，転送の回数は

$$N_T^{(3)}=2(n-1) \qquad (33)$$

でよいことになる．これは転送回数の下限を与える．しかしながら，このような方法は実現困難であるから，その中間的存在として次のような方法を考える．でき上った数列 $a_1, a_2, \cdots\cdots, a_r$ に対して新しく2個の元 c_{r+1}, c_{r+2} を取って来てこれを保持しておく．この2個の元と数列 $a_1, a_2, \cdots\cdots, a_r$ について完全な置換を行なう．すなわち，最初に a_r, c_{r+1}, c_{r+2} の間のすべての大小関係を比較によって定める．これは最高3回の比較を行なえばよい．このようにして3個の元の順序関係が定められたならば，移動すべき元を最小の手数で移動する．位置の未定な2個の元または1個の元は，レジスタに保持したまま列の若番の方へ移動して，次に a_{r-1} と比較を行なう．このようにして c_{r+1}, c_{r+2} のうち，小なる方が列中に入れられたときに，この一連の置換が終了する．この方法による転送回数を求めると計算過程は省略するが，

（1） c_{r+2} が移動するときは，c_{r+1}, c_{r+2} のうち小さな方が入る位置までの元をすべて2ステップずつで移動する．

（2） c_{r+2} が移動しないときは c_{r+1} に関する巡回置換を行なう．

（3） 両者とも移動しないときは手数をゼロとする．

として次式がえられる．

$$N_T^{(4)}=\frac{1}{3}(n^2+4n-5)-\sum_{i=1}^{(n-1)/2}\frac{1}{i} \quad (n\geqq 3;\ \text{奇数})$$
$$=\frac{1}{3}(n^2+5n-5)-\sum_{i=1}^{n/2}\frac{1}{i} \quad (n\geqq 4;\ \text{偶数}) \qquad (34)$$

これは二重巡回置換法とでも呼ぶべきものである．新しく取って来る元の数 δ を増すほど手数が減り，$r=1$ に対して $\delta=n-1$ とすれば(33)に一致することになる．しかしながら，比較の結果をプログラム上で分岐して区別してゆくとすれば，δ を大にするとプログラムがきわめて長大なものになってしまう．その上この方法では δ 個の元を保持できるレジスタが必要となる．

以上は交換法について考察したが Shell の方法では多少異なったものとなるであろうし，実際の分類ではこの他に比較のために元をとり出す操作を合せて考えなければならない．

この方法を応用できる例としては，まず第一にドラム計算機の場合に磁気コアのようなバッファメモリを用いる場合ならば，ドラムからの転送回数を減らす方法としてそのまま使用できる．$\delta=2$ の場合の例で，NEAC 2203 を用いて行なった実験例を**第4表**に示す．プログラムの長さは約2倍に増加したが，分類時間は約 74% に減少した．プログラム部分は磁気コアに，データは内部ドラムに入れられてある．また他の例としては計算機の機能上，比較・転送に使用できるレジスタの数に余裕がある場合にも近似的に使用可能となり，転送回数を相当に減らすことができる．

第4表 2重巡回置換法の実験例

n	巡回置換法	2重巡回置換法	
181	25秒	18秒	72%
395	1分06	46	70
779	2. 16	1分40	74
982	3. 05	2. 24	78
		（平均	74%）

NEAC 2203 を使用し，プログラム部分を磁気コアにデータを内部ドラムに入れた場合の分類時間．

7. むすび

以上すぐれた内部分類の手法である Shell の分類法について近似的な考察を行ない，Frank & Lazarus の与えた改良は最適値に対してかなり良い近似であることを示した．しかし元の数が大になるとさらに良い方法があることも示された．この方法は実時間においても高速となることは十分期待される．

また数を転送する方法については多重巡回置換を行なうことが有効となる場合があって，実験的にも十分高速であることがわかった．

しかしながら，本稿に述べた解析はきわめて近似的な計算であるから正確な解が得られたとはいい難い．さらに今後 Shell の方法以上のすぐれた方法が現れるかどうかに関しては何ともいえない．

終りに，ご指導いただいた当所福井電子回路，岸上情報処理両研究室長に感謝の意を表する．

参考文献

1) 淵 一博：分類（1），情報処理，1961年3月
2) D.L. Shell: A High Speed Sorting Procedure, Comm. of ACM, July 1959
3) R.M. Frank and R.B. Lazarus: A High Speed Sorting Procedure, Comm. of ACM, Jan. 1960
4) 喜安善市・池野信一：分類の情報理論的考察，電気通信学会インホメーション理論研究専門委員会資料，1960年1月19日

（昭和38年6月28日受付）