Z77-4469

TECHNICAL INFORMATION EXCHANGE

SORTING TECHNOLOGY IN PROGRAMMING

Mr. J. W. Toner
IBM Corporation
1701 North Street
Endicott, NewYork

IBM Corporation, Technical Publications Dept., 112 E. Post Road, White Plains, N.Y. 10601

IBM®

July 9, 1964

In many applications, sorting or the ordering of data records constitutes a large portion of the total usage of a data processing system. This investigation of sorting technology is aimed at realizing better approaches to the sorting of data.
An improved manner for carrying out the merging operation is achieved by conducting the Phase 2 portion of the program as an unbalanced merging operation. In this type of merging, the number of input tapes in phase 2 is greater than the number of output tapes. After each merge some of the tape assignments are changed in such a way that the unbalanced merge can continue. In order for unbalanced merging to operate properly, the distribution of sequences, as received in phase 2, must follow a definite and predetermined relationship. There are a number of different ways in which unbalanced merging can be carried out. Also there are several ways in which the sequences can be distributed prior to input to phase 2 to arrive at the correct sequence relationship.
This paper describes unbalanced merging, specifically the polyphase merge, and indicates a means for evaluating unbalanced merging performance. In addition, a number of sequence redistribution schemes are discussed and evaluated.

Z77-4469

SORTING TECHNOLOGY IN PROGRAMMING

by

J.W. Toner

INTRODUCTION

In many applications, sorting or the ordering of data records constitutes a large portion of the total usage of a data processing system. This investigation of sorting technology is aimed at realizing more optimum approaches to the sorting of data.

Sorting is generally accomplished by reading into the primary storage (core) of a computer as large a portion of the total data file as is possible. These groups are then sorted internally and written out on secondary storage (tapes or disks) in a series of sorted sequences. This internal sort is generally designated as Phase 1 of the total sort operation. These series of sequences are read into the machine, compared, and combined into longer sequences which are then written out again on tapes or disk(s). This operation of reading in, comparing, merging and writing out is repeated as many times as are needed to completely order the entire file and form a single sequence. This merging operation is generally designated as external merging in that the data (except for those records being compared) is external to core. The merging phase is also generally called Phase 2 of the total sorting operation.

Until a few years ago, the merging operation in Phase 2 was carried out in a "balanced mode". In balanced tape merging, the data is distributed out of Phase 1 onto $\frac{K+1}{2}$ tapes where $\frac{K+1}{2}$ is equal to half the total number of tapes available for merging. The data is passed back and forth in the merge in a balanced fashion, i.e. data is output on the same number of tapes as were used for input. One passage of the total file through the machine is designated as a pass. In the merging phase (Phase 2) there can be a series of passes, as many as needed to order the data into one sequence. In most applications, Phase 2 consumes the bulk (70-80%) of the total time required to perform the sort.

An improved manner of carrying out the merging operation is to conduct Phase 2 as an unbalanced merging operation. In unbalanced tape merging the number of input tapes for a merging operation in Phase 2 is greater than the number of output tapes. After each merge some of the tape assignments are changed in such a way that the unbalanced merge can continue. In order for unbalanced merging to operate properly, the distribution of sequences as received in Phase 2 must follow a definite, predetermined relationship. There are a number of different ways in which unbalanced merging can be carried out, and there are also a number of ways in which the sequences can be distributed before being input to Phase 2 to arrive at the correct sequence relationship.

Some of the unbalanced sorting techniques that can be employed in Phase 2 are:

A. Polyphase merging (1)
B. Oscillating merging (2)
C. Cascade merging (3)
D. Compromise merging (4)

A. POLYPHASE MERGING

Polyphase merging on a system with (K + 1) total system tapes causes (K) input tapes to be merged, i.e. a (K) - way merge onto a single tape. This (K) - way comparison, designated as a (K) - order merge continues throughout all of Phase 2 until the file is sorted.

B. OSCILLATING MERGING

Oscillating merging on a system with (K + 1) total system tapes causes (K - 1) input tapes to be merged, i.e. a (K - 1) - order merge onto a single tape. This (K - 1)-way merge continues throughout all of Phase 2 until the file is sorted. The name "oscillating" derives from the fact that the execution of Phase 1 and Phase 2 is interleaved so that the sort "oscillates" between performing Phase 1 and Phase 2. In all other techniques, both balanced and unbalanced, Phase 2 is not started until all the data has been internally sorted and Phase 1 is therefore completed, releasing the input tape for use in Phase 2.

C. CASCADE MERGING

Cascade merging in a system with (K + 1) total system tapes causes (K) input tapes to be merged; then (K - 1), (K - 2) — down to a 2-way merge. This variation in the order of the merge exists within each pass of the data. Each pass starts with (K) - way comparison until one of the input tapes becomes empty; then (K - 1) comparison until another input tape becomes empty; then (K - 2), etc.

D. COMPROMISE MERGING

Compromise merging on a system with (K + 1) total system tapes causes (K), (K - 1) — input tapes to be merged as in Cascade Merging. In Compromise Merging, the order of merge does not vary from (K) down to 2, but can end at some intermediate value (K - Z) where Z is an integer and may depend on the total number of tapes on the system; the total number of sequences, or some other sort parameter. (K - Z) must be a positive integer.

The initial effort in the investigation of sorting technology has been directed towards the evaluation of the polyphase merge as applied to tape operation. Polyphase merging appears to be advantageous compared to oscillating merging on systems equipped with a small number of tapes (3-6 tape units). In this evaluation, the definition of a polyphase merge will be that unbalanced merge technique in which the number of sequences on the merging tapes is a linear combination of the Kth generalized Fibonacci numbers (5) (where K + 1 = the number of tapes available). A further assumption will be that the sequences as they are generated in Phase 1 are all of equal length (i.e., each sequence contains an equal number of records).

In this discussion only the relative merits of the merging phase of the total sort program will be considered. For unbalanced sorting procedures, (A,C,D) there is a redistribution pass (or a "lower order of merge" pass for procedure B) to handle the situation where the number of sequences is not exactly equal to an unbalanced merge table entry combination or where (procedure B) the number of sequences is not an exact power of the order of merge.

II UNBALANCED MERGE EVALUATION PROCEDURE

One common factor that can be used to compare various approaches to merging is the factor "r" which is the total number of times the records are handled (R'), divided by the total number of records being sorted (R), i.e. an average record handling figure. For oscillating merging this can be expressed as:

$$(K - 1)^r \geq \frac{R}{G}$$

where

$$r = \frac{R'}{R}$$

R' = Total number of times records are handled (written).

R = Total number of records to be sorted.

G = Number of records sorted internally in Phase I.

For a polyphase merge, "r" can be expressed as:

$$r = \frac{G}{R} \sum_i f_i \; g_i$$

where:

f_i = Length in number of sequences merged in each individual polyphase merge operation.

g_i = Number of records operated upon in an individual polyphase merge operation.

If the sequence distribution at the start of the polyphase merge is given by

Tape Number	1	2	3	K	K + 1
Original Distribution	C_1	C_2	C_3 --	C_K	0

where $C_1 \geq C_2 \geq C_3 \geq \ldots C_K$ and the C terms are linear combinations of the Kth generalized Fibonacci numbers, then the f_i sequence can be expressed as:

First Term = C_K

Second Term = $C_{(K-1)} - C_K$

Third Term = $C_{(K-2)} - C_{(K-1)}$

These are the first K terms in the sequence

(K + 1) Term = $C_K - \frac{(C_{(K-1)} - C_K) - (C_{(K-2)} - C_{(K-1)}) - \ldots}{\text{Number of terms subtracted} = K - 1}$

(K + 2) Term = $(C_{(K-1)} - C_K) - \frac{(C_{(K-2)} - C_{(K-1)}) - (C_{(K-3)} - C_{(K-2)}) - \ldots}{\text{Number of terms subtracted} = K - 1}$

etc.

Any term (K + n) can be obtained by taking the Kth previous term in the sequence and repetitively subtracting the (K-1) following term from it.

The g_i sequence can be expressed for K = 3 by dropping the first K terms of the following sequence

$g_1\ g_2\ g_3\ g_4\ g_5$

1, 1, 1, 3, 5, 9, 17, 31, ---

First K terms — Start of sequence

Generally the sequence starts following K ones. Each succeeding term is the sum of the previous (K) terms.

When two successive terms of the f_i sequence are unity, the sequence terminates.

Example

$K + 1 = 4$

Express $R = 31\ G$

i.e., R is an exact multiple of a polyphase merge table entry combination.

Tape No.	1	2	3	4	
	13	11	7	0	← 13, 11 and 7 are Number of Sequences
	6	4	0	7	
	2	0	4	3	
	0	2	2	1	
	1	1	1	0	
	0	0	0	1	

$C_1 = 13$

$C_2 = 11$

$C_3 = 7$

Assume each of these sequences of equal length = G

f_i sequence

$C_K = 7 = C_3$	Series	C_3	$= 7$	$= f_1$
$C_{K-1} = 11 = C_2$		$C_2 - C_3$	$= 4$	$= f_2$
$C_{K-2} = 13 = C_1$		$C_1 - C_2$	$= 2$	$= f_3$

$(K + 1)$ Term $= C_3 - (C_2 - C_3) - (C_1 - C_2) \quad 1 = f_4$

$(K + 2)$ Term $= (C_2 - C_3) - (C_1 - C_2) - (K+1)$ Term $= 1 = f_5$

g_i sequence

$g_1\ g_2\ g_3\ g_4\ g_5$

1, 1, 1, 3, 5, 9, 17, 31, ----

K = 3 ones — Start of sequence

$$r = \frac{G}{R} \quad f_i\ g_i$$

$$= \frac{G}{R} \quad (f_1 g_1 + f_2 g_2 + f_3 g_3 + f_4 g_4 + f_5 g_5)$$

$$r = \frac{G}{R} \quad 7 \times 3 + 4 \times 5 + 2 \times 9 + 1 \times 17 + 1 \times 31)$$

$$R = (C_1 + C_2 + C_3)\ G = 31\ G$$

$$r = \frac{(21 + 20 + 18 + 17 + 31)}{31} = \frac{107}{31} = 3.45$$

The record handling figure "r" can be expressed more rigorously. The K-generalized Fibonacci numbers $f_{j,K}$ are defined in (9) as

$$f_{j,K} = 0 \qquad 0 \leq j \leq K - 2$$

$$f_{K-1,K} = 1$$

$$f_{j,K} = \sum_{n=1}^{K} f_{n-n,K} \qquad j \geq K$$

In (5), a generalized polyphase merge of sequences using K + 1 tapes is defined in terms of linear combinations of the K-generalized Fibonacci numbers where for $j \geq (K + 1 - n)$

$$C_{n,j,K} = \sum_{a=j-(K-n+1)}^{a=j-1} f_{a,K} \; (n = 1, 2 - - - -, K)$$

where $C_{n,j,K}$ is the number of sequences on tape n at the original distribution at index level j for K tapes.

For $j < (K + 1 - n)$

$$C_{n,j,K} = 0 \qquad n \neq 1$$

and
$$C_{1,j,K} = \begin{cases} 0 & j < K - 1 \\ 1 & j = K - 1 \end{cases}$$

The set of the Kth generalized numbers varies on index j. For any fixed value of j there is generated a value for $C_{n,j,K}$, the proper number of sequences on tape n for the polyphase merge.

The total record handling (R') is the sum of the records handled at each merge of the polyphase merge and can be expressed as

$$R' = C_{1,j-1,K}\left(G \sum_{n=1}^{n=K} C_{n,K,K}\right) + C_{1,j-2,K}\left(G \sum_{n=1}^{n=K} C_{n,K+1,K}\right)$$

$$+ C_{1,j-3,K}\left(G \sum_{n=1}^{n=K} C_{n,K+2,K}\right) + - - - + C_{1,j-a,K}\left(G \sum_{n=1}^{n=K} C_{n,K+a-1,K}\right)$$

$$R = G \sum_{n=1}^{n=K} C_{n,j,K}$$

$$r = \frac{R'}{R} = \frac{\sum_{a=1}^{a=j-K+1} C_{1,j-a,K} \sum_{n=1}^{n=K} C_{n,K+a-1,K}}{\sum_{n=1}^{n=K} C_{n,j,K}}$$

The record handling figure "r" can be expressed in terms of linear combinations of the K-generalized Fibonacci numbers by substituting the appropriate Fibonacci summations in the expression for "r" just given:

$$r = \frac{\sum_{a=1}^{a=j-K+1}\left(\sum_{\gamma=j-a-K}^{\gamma=j-a-1} f_{\gamma,K}\right)\sum_{n=1}^{n=K}\left(\sum_{\gamma=a-2+n}^{\gamma=K+a-2} f_{\gamma,K}\right)}{\sum_{n=1}^{n=K}\sum_{\gamma=j-(K-n+1)}^{\gamma=j-1} f_{\gamma,K}}$$

Since by definition

$$\sum_{\gamma=j-a-K}^{\gamma=j-a-1} f_{\gamma,K} = f_{j-a,K}$$

Then r can also be expressed as

$$r = \frac{\sum_{a=1}^{a=j-K+1} f_{j-a,K} \sum_{n=1}^{n=K}\left(\sum_{\gamma=a-2+n}^{\gamma=K+a-2} f_{\gamma,K}\right)}{\sum_{n=1}^{n=K}\left(\sum_{\gamma=j-(K-n+1)}^{\gamma=j-1} f_{\gamma,K}\right)}$$

Figure 1 shows a plot of the average number of record handlings as a function of file size. File size is given in terms of C x G records where C is an integer.

From this analysis, it appears that polyphase sorting has some advantages when the number of tapes is small. If a three tape system, using unbalanced merging, were compared with a 4 tape system using balanced techniques, the average record handling figure for the 3 tape system would be approximately 10% greater than for the 4 tape system. Using this as a comparison criterion, the three tape unbalanced system would run approximately 10% more slowly than the 4 tape balanced system. If one compares 3 tape unbalanced with 4 tape unbalanced, one finds that the 3 tape would operate somewhere in the range of 50% of the speed of the 4 tape. This estimate is only approximate since the tape block sizes are not the same; 3 tape unbalanced will accommodate a larger block size than 4 tape unbalanced.

Further investigation of unbalanced merging procedures is called for to indicate other possible areas of improvement and optimization.

III SEQUENCE REDISTRIBUTION SCHEMES

In unbalanced polyphase merging, some redistribution of the sequences as they are output from Phase 1 is needed to arrive at an exact polyphase table sequence distribution level. This redistribution can be accomplished in a variety of ways. Approaches that will be evaluated are the redistribution schemes described in "An Elementary Polyphase Merge Algorithm" (6), "A Generalized Polyphase Merge Algorithm" (5) and "A Dispersion Pass Algorithm for the Polyphase Merge" (8). This comparison will be made for a three tape system (where K + 1 = the number of tapes available = 3). A further assumption will be that the sequences as they are generated in Phase 1 are all equal length (i.e., each sequence contains an equal number of records).

A. ELEMENTARY POLYPHASE ALGORITHM

The redistribution scheme described in (6) was developed for a four tape system and has been modified and applied to a three tape system for this comparison. This analysis has already been described (7) and only the results will be included for this comparison. In this operation, the sequences from Phase 1 are output on 2 tapes alternately (the same as for balanced 2 way sort operation). The redistribution phase redistributes these sequences to a correct Polyphase table value. The following terms are used in this evaluation:

$a + K'$	The number of sequences of sorted records on tape 1 prior to redistribution.
a	The number of sequences of sorted records on tape 2 prior to redistribution.
K'	The number, either zero or one, of sequences by which the number of sequences on tape 1 differs from the number of sequences on tape 2 prior to redistribution.
S	The total number of sequences to be merged prior to redistribution.
N	The total number of sequences to be merged after redistribution.
a1, b1	The number of sequences on tapes 1 and 2 respectively after redistribution.
p1	The number of sequences passed from tape 3 to tape 1.
p2	The number of sequences passed from tape 3 to tape 2.
X'	The number of sequences merged from tapes 1 and 2 onto tape 3.

The redistribution is accomplished by:

1. A 2 way merge of X' sequences from tapes 1 and 2 to 3.
2. A pass of p1 sequences from tape 3 to tape 1.
3. A pass of p2 sequences from tape 3 to tape 2.

<u>Redistribution Phase</u>

	Tape 1	Tape 2	Tape 3
S	$a + K'$	a	0
	$a + K' - X'$	$a - X'$	X'
	$a + K' - X' + p1$	$a - X'$	$X' - p1$
	$a + K' - X' + p1$	$a - X' + p2$	$X' - p1 - p2$
N	$= a_1$	$= b_1$	$= 0$

In this redistribution the following equations apply:

$$S = 2a + K'$$
$$N = a_1 + b_1$$
$$a_1 = a + K' - X' + p1$$
$$b_1 = a - X' + p2$$
$$X' - p1 - p2 = 0$$

From these above equations, the following equations may be derived:

$$X' = S - N$$
$$p1 = a_1 + X' - (a + K')$$
$$p2 = b_1 + X' - a$$

A further restriction that must be applied is that

$S \leq 2N$ (This determines the maximum value of N for any value of S).

From this analysis one can determine the amount of record handling that is required in the redistribution phase. If the sequences from Phase 1 are of length G, then R' (Number of record handlings) is given by

$$R' (7) = 4 (S - N) G$$

R' (7) is the number of record handlings for algorithm (Reference 7).

The resulting record handlings as a function of file size are shown in the accompanying graph (Figure 2). At values of S one higher than 2 x N the value of S - N is a minimum which accounts for the steep slopes in the plots.

Example:

N = 13	S	N	S - N	4 (S - N) G = R' (7)
	24	13	11	44G
	25	13	12	48G
S = 2 N	26	13	13	52G
	27	21	6	24G
	28	21	7	28G

As S goes from 26 to 27, R' changes from 52G to 24G.

B. GENERALIZED POLYPHASE ALGORITHM

For the Generalized Polyphase Algorithm, the redistribution will begin with either an excess of sequences (X) on tape 1 or an excess on tape 2.

Case I

Tape No.	1	2	3
	$C_1 + X$	C_2	0
	C_1	$C_2 - X$	X
	C_1	0	C_2

Case II

Tape No.	1	2	3
	$C_1 + C_2$	$C_2 + X$	0
	$C_1 - X$	0	$C_2 + X$
	C_2	0	C_1

For Case I

$$R' = (C_2 + X) \; G$$

For Case II

$$R' = (C_1 + C_2 + X) \; G$$

When X = 0

$$R' = 0 \text{ for Case I}$$

$$R' = 2 C_2 G \text{ for Case II}$$

When X = 0, the full algorithm need not be accomplished; hence, R' is defined as shown.

C. DISPERSION PASS ALGORITHM

For the "dispersion pass algorithm" described in (8), the sequences are built up in such a manner that the largest number of sequences at any polyphase table entry value (C_1) remains constant as the sequences on the remaining tape(s) C_2 , C_3 , etc., are increased alternately to attain the next table values. Using the same notation as was used in (5), we can conclude for a 3 tape case:

C_1 at level i of polyphase table is to remain constant until level i + 1 is reached.

C_2 at level i is to be incremented by m (Reference 8: Case 3) until level i + 1 is reached.

where

m is the number of sequences that C_2 has been incremented towards the (i+1) table level.

Algorithm

1. Copy $(C_1 - m)$ sequences from "C_1" tape onto empty tape No. 3.
2. 2 way merge m sequences from "C_1" and "C_2" tapes onto No. 3.

For this algorithm:

$$R' = (C_1 + m) \ G$$

In (5), starting from any table value, C_1 is always incremented first until the next level $(C_1 + C_2)$ is attained.

For Case I

$$R' = (C_2 + X) \ G$$

Where X is number of sequences C_1 has been incremented towards the (i+1) table level.

For this range

$$\therefore R'(5) < R'(8)$$

R' (5) is the number of record handlings for algorithm (Reference 5).

R' (8) is the number of record handlings for algorithm (Reference 8).

For the range (Case II in (5)).

C_1 in (5) has been incremented to the next level $(C_1 + C_2)$ and X is now the number of sequences C_2 has been incremented towards the (i+1) table level.

In (8) this intermediate level (Case II) is not reached as m increases uniformly from 0 to $(C_1 - 1)$.

For Case II range in (5) then

$$X + C_2 = m$$

Hence

$$R'(8) = (C_1 + m) \ G = (C_1 + C_2 + X) \ G$$

which is the same as in (5).

$$\therefore R'(5) = R'(8)$$

Since

$$R'(8) = (C_1 + m) \ G$$

and

$$C_1 + C_2 + m = S$$

$$\therefore R'(8) = (S - C_2) \ G$$

<u>Example:</u>

<u>Algorithm (5)</u>

Number of Sequences	C_1	C_2	Type Case	x	$R' = (C_2 + x)$ Case I; $X = 0$, $R' = 0$; $R' = (C_1 + C_2 + x)$ Case II; $X = 0$ $R' = 2C_2$
<u>13</u>	8	5	-	0	0
14	9	5	I	1	6
15	10	5	I	2	7
16	11	5	I	3	8
17	12	5	I	4	9
18	<u>13</u>	5	II	0	10
19	13	6	II	1	14
20	13	7	II	2	15
<u>21</u>	<u>13</u>	<u>8</u>	-	0	0

Algorithm (8)

Number of Sequences	C_1	C_2	m	R' = (C_1 + m); m = 0, R' = 0
13	8	5	0	0
14	8	6	1	9
15	8	7	2	10
16	8	8	3	11
17	8	9	4	12
18	8	10	5	13
19	8	11	6	14
20	8	12	7	15
21	8	13	0	0

Algorithm (5) x	Algorithm (5) R'		Algorithm (8) m	Algorithm (8) R'
0	0		0	0
1	6		1	9
2	7	Case I	2	10
3	8		3	11
4	9		4	12
0	10		5	13
1	14	Case II	6	14
2	15	C_2 = 5	7	15
0	0	x + C_2 = m	0	0

A plot of R' as a function of file size for these redistribution schemes is shown in Figure 2.

The redistribution scheme proposed in (5) requires fewer record handlings than the scheme proposed in (6) modified for 3 tapes. Furthermore, as the number of sequences increases, (6) requires proportionally more record handling than does (5). Redistribution scheme (5) requires less than or an equal number of record handlings than does (8).

There are a number of different redistribution schemes that could be investigated and evaluated as to their relative efficiency. In addition, the mode of outputting from Phase 1 affects the type of redistribution phase needed and must also be considered in evaluating the total sequence distribution and redistribution.

V REFERENCES

1. Gilstad, R.L., Polyphase Merge Sorting — An Advanced Technique, Proc. EJCC, December, 1960.

2. Sobel, S., Oscillating Sort — A New Merge Sorting Technique, JACM, July, 1962.

3. Betz, B.K. and W.C. Carter, New Merge Sorting Techniques, Proc. EJCC, September, 1959.

4. Knuth, D., Letter to Editor , Comm. ACM, October, 1963.

5. Reynolds, S.W., A Generalized Polyphase Merge Algorithm, Com. ACM, August, 1961.

   Addendum to A Generalized Polyphase Merge Algorithm, Comm. ACM, Nov., 1961.

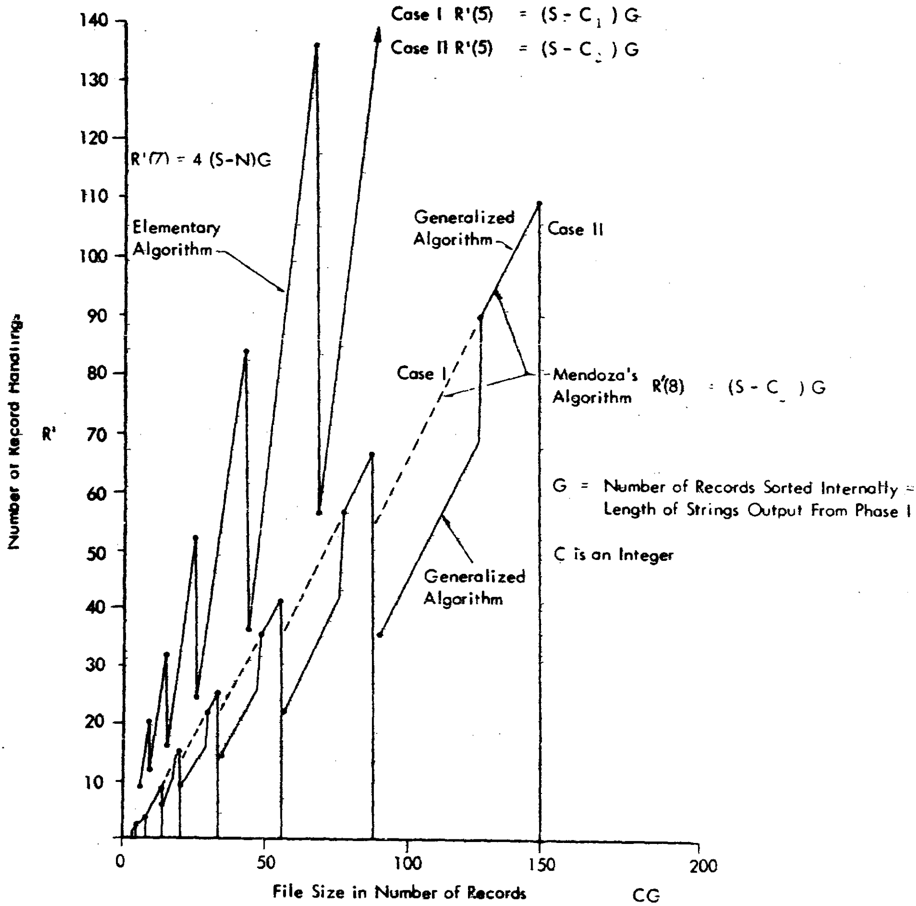
6. Reynolds, S.W., An Elementary Polyphase Merge Algorithm, IBM Unpublished Paper.

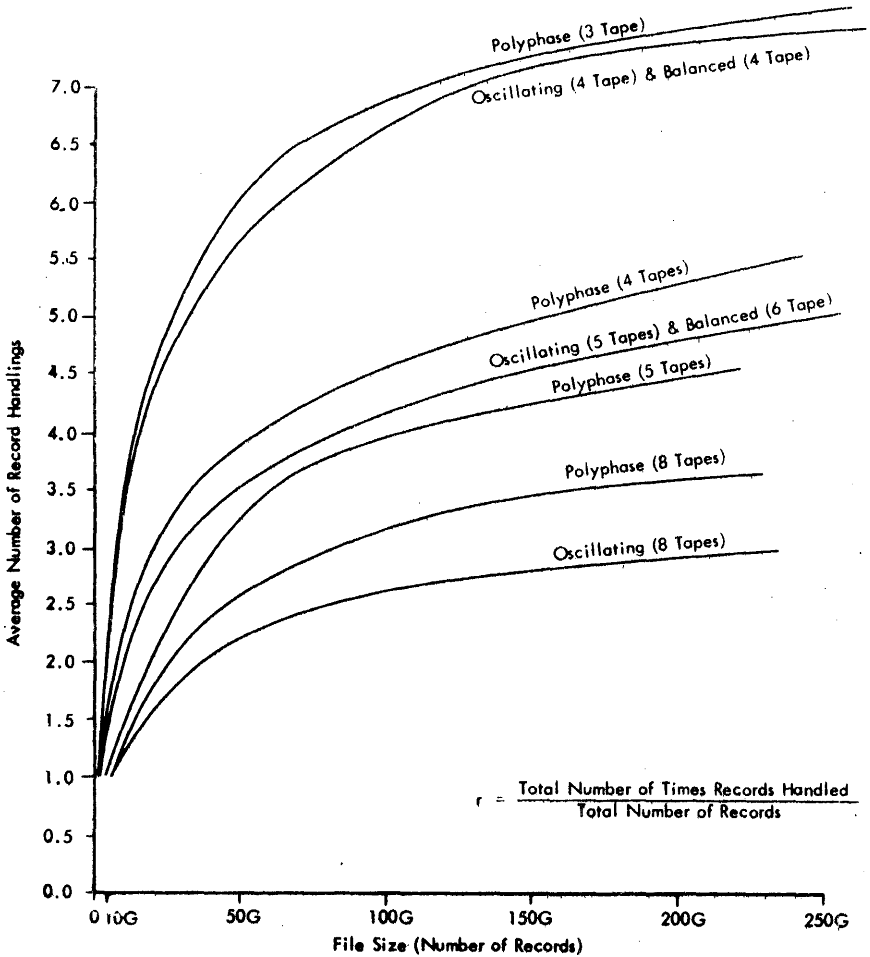
7. Blankenship, J.H., Polyphase Merge Algorithm for Three Tape System, IBM Unpublished Paper.

8. Mendoza, A.G., A Dispersion Pass Algorithm for the Polyphase Merge, Comm. ACM, October, 1962.

9. Miles, E.P., Jr., Generalized Fibonacci Numbers and Associated Matrices, Amer. Math Monthly, 67, October, 1960.

LIST OF FIGURE CAPTIONS


Case I R'(5) = (S - C1) G
Case II R'(5) = (S - C2) G
R'(7) = 4 (S-N)G
Elementary Algorithm
Generalized Algorithm
Case II
Case I
Mendoza's Algorithm
R'(8) = (S - C) G
G = Number of Records Sorted Internally = Length of Strings Output From Phase I
C is an Integer
Generalized Algorithm
Number of Record Handlings
R'
File Size in Number of Records
CG
0
50
100
150
200
10
20
30
40
50
60
70
80
90
100
110
120
130
140



G Number of Records Sorted Internally - Ph I

C is an integer

Note: — Oscill ting Sort (2) requires added input tape drive and read backwards feature.

The balanced an oscillating sort record handling figure i actually a discontinuous function of (R/G

$$\frac{K+1}{2} \geq \frac{R}{G} \quad \text{for Balanced Sort}$$

$$(K-1) \geq \frac{R}{G}, \quad \text{for Oscillating Sort}$$

For comparison purposes, they are shown as continuous functions through the points of discontinuity.

$$r = \frac{\text{Total Number of Times Records Handled}}{\text{Total Number of Records}}$$ Avg. Number of Record Handling