

Previous work, which analyzed certain merge-sorting methods with the aid of difference equations, is extended to include a wider range of methods. Matrices are introduced to represent the set or sets of difference equations associated with a merge-sort. Two or more matrices are required to define a Class II method, whereas a Class I method can be defined with one matrix. The merge-sorts of most interest fall into a special subclass called Class Ia.

It is shown that an asymptotic solution to the set of difference equations for a Class Ia merge-sort is readily obtainable. Carter's analysis of cascade and polyphase merge-sorts is generalized and extended to include, among other things, the compromise merges. Various properties of the Class Ia merge-sorts, including relative performance measures and explicit merge patterns, are shown to be obtainable by matrix multiplication. Although the analysis emphasizes Class Ia merges, suggestions are given for applying the matrix technique to other merge-sorts of Classes I and II.

Merge-sort analysis by matrix techniques

by C. E. Radke

Simply stated, the problem of sorting is that of arranging a large number of data records into a prescribed order. Sorting usually consists of two phases, *internal presort* to form sequences (strings) and *sequence merge*. This paper is concerned with an analysis of the second phase, sequence merge. However, the analysis does not treat the internal handling of records, but rather the dynamics of their external ordering. The procedures discussed relate to the ordering of records on tapes and to the ordering of the tapes by which records are read into the computer, and after merging, the ordering of records and tapes on which the resulting merged sequences are written. Considering tape drives as external memory devices, tapes can then be regarded as memory locations of sets (possibly empty) of ordered sequences of records.

In general, balanced merges using tape memory begin with a set of N tapes partitioned into two groups, K and $N - K$ (where K and $N - K$ are not necessarily equal). The balanced merge procedure is as follows. A sequence of records from each of K tapes is read, and the set of sequences is merged, i.e., properly ordered. The resultant ordered sequence is written on one of the initially empty $N - K$ tapes and other sequences on succeeding tapes. The procedure of reading sequences from the K tapes and writing merged sequences on successive $N - K$ tapes continues until all K tapes are empty. The merge switches back and forth between the two fixed, disjointed sets of tapes until all records reside on one tape in a single sequence.

In 1962, W. C. Carter¹ analyzed several unbalanced merge-sort methods using difference equations. Unbalanced merging permutes the choice of the set of initially nonempty tapes from which sequences of records are to be read. The choice of the output tape is also permuted among the set of tapes. Carter's analysis includes the unbalanced cascade and polyphase merge-sorts.

The present analysis classifies and gives generalized treatment not only to the unbalanced merge-sorts discussed by Carter, but also the compromise merge-sorts described by Knuth.² The analysis is generalized by the use of difference equations. Among the several classes of merge-sorts, Class Ia is explicitly defined and includes certain of the cascade, compromise, and polyphase merge-sorts. Class Ia and the larger defined Class I can both be described by a single set of difference equations. Description by a single set of difference equations is interpreted as implying that a merge go to completion by the use of a single permutation procedure. By this property, the merge is said to go to a *proper completion*.

On the other hand, Class II merge-sorts are defined as those described by more than one set of difference equations. In this case, more than one permutation procedure is required to enable the merge to go to completion; the merge is then said to go to an *improper completion*.

Matrices are used to represent the set(s) of difference equations associated with a particular merge-sort. It is shown that properties of various merge-sorts, especially those of Class Ia, can be obtained by matrix manipulation. These properties include relative performance measures and explicit patterns of ascending and descending sequences; particular patterns are required from the presort phase for read-backward tape operation. Tape read-backward operation is assumed throughout the present analysis.

Asymptotic solutions to difference equations can provide an approximation to the number of records merged into sequences on each tape during each pass. Carter determined asymptotic solutions to the difference equations representing the cascade merge for a limited number of examples. The results of the present analysis show that one can obtain asymptotic solutions to all sets of difference equations that describe either the cascade or polyphase merge-sorts of Class Ia.

Classes of merge-sorts

Class Ia merge-sorts with $K + 1$ system tapes are described as procedures in which one cycle of the merge consists of (K) -way, $(K - 1)$ -way, $(K - 2)$ -way, $\dots$, $(K - i)$ -way consecutive merges. The i may be chosen to be 0, 1, 2, $\dots$, $K - 2$, or $K - 1$, and the tape with the lowest number of sequences is emptied first, the second lowest next, etc., down to the $(i + 1)$ st lowest such that a sequence of permutations of the K tapes is repeated over and over. In such a description of Class Ia merge-sorts, a given distribution of sequences on K tapes is assumed.

class Ia

Using the difference equation approach of Carter, we can define the set of difference equations that describes the merge by the matrix equation

$$\mathbf{m}_i A = \mathbf{m}_{i+1} \quad (1)$$

A is a $(0, 1)$ -matrix, $\mathbf{m}_i$ is a row vector whose elements indicate the number of sequences on the K tapes at the end of the $(r - j)$ th cycle, $j = 0, 1, 2, \dots, r$. In the above, r is the total number of cycles required for completion of the merge. Then, defining $E\mathbf{m}_i = \mathbf{m}_{i+1}$, where E is the difference operator, as is customary, we have

$$\mathbf{m}_i A = E\mathbf{m}_i$$

The elements in the vector $\mathbf{m}_i$ for Class Ia merges are assumed to be in a definite order

$$m_{1i} \geq m_{2i} \geq \dots \geq m_{ii} \geq \dots \geq m_{Ki} \quad j = 0, 1, 2, \dots, r$$

However, a definite order will not be assumed for the other classes of merge-sorts. For $j = 0$, the vector at the end of the r th cycle is

$$\mathbf{m}_0 = (m_{10}, m_{20}, \dots, m_{K0}) = (1, 0, \dots, 0)$$

The elements of the matrix A for Class Ia are explicitly defined as follows for any given i , where $i = 0, 1, 2, \dots, K - 1$:

$$a_{st} = \begin{cases} 1 & t = 1, 2, \dots, K - s + 1 \\ 0 & \text{otherwise} \end{cases} \quad s = 1, 2, \dots, i + 1$$

and

$$a_{st} = \begin{cases} 1 & t = s - (i + 1) \\ 0 & \text{otherwise} \end{cases} \quad s = i + 2, \dots, K$$

Observe that we do not assign a fixed number to any tape. Instead, we determine which set of tapes is to be merged from only the Class Ia description and the ordering of the elements of the vector $\mathbf{m}_i$ for each cycle of the merge. We define a *step* as each $(K - i)$ -way merge. (A step is sometimes referred to as a *level*.) Each *cycle* then is made up of $(i + 1)$ steps, where again $i = 0, 1, 2, \dots$, or $K - 1$.

We define the elements of the row vector $\mathbf{d}_i = (d_{1i}, d_{2i}, \dots, d_{Ki})$, where $j = 0, 1, 2, 3, \dots, r$, to be the lengths of the sequences at the end of the j th cycle. Observe that d_{k0} , where $k = 1, 2, \dots, K$, is the length of the sequences at the output of the presort. As for $\mathbf{m}_i$, we assume an order for the elements of $\mathbf{d}_i$ for Class Ia merges,

$$d_{1i} \geq d_{2i} \geq \dots \geq d_{Ki}$$

Again a definite order cannot be assumed for the general merge in Classes I and II. Also, we assume³ that for $j = 0$

$$d_{10} = d_{20} = \dots = d_{K0}$$

As for m_j , we can define the set of difference equations that describe the merge by the matrix equation

$$d_j B = E d_j \quad (2)$$

where B is a $(0, 1)$ -matrix. Equations 1 and 2 may not hold for all values of j ; that is, different matrices may be required to satisfy these equations for the various values. We define *Class I* as that class of merge-sorts that can be described by $(0, 1)$ -matrices A and B such that Equation 1 holds for all $j \geq 0$ for a matrix A , and Equation 2 holds for all $j \geq 0$ for a matrix B .

Class II, in turn, is that class of merge-sorts which can be described by a finite set of $(0, 1)$ -matrices $A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m$, where $n > 1$ and $m \geq 1$.

The A matrix is such that

$$\begin{array}{ll} m_{j_1-1} A_1 = m_{j_1} & j_1 \in J_1 \\ m_{j_2-1} A_2 = m_{j_2} & j_2 \in J_2 \\ \vdots & \vdots \\ m_{j_n-1} A_n = m_{j_n} & j_n \in J_n \end{array} \quad (3)$$

where $|J_1| + |J_2| + \dots + |J_n| = r$ (r is the total number of cycles required, and $|J_i|$ indicates cardinality of the set J_i) and $A_i \neq A_j$ for $i \neq j$.

The B matrix is such that

$$\begin{array}{ll} d_{i_1-1} B_1 = d_{i_1} & i_1 \in I_1 \\ d_{i_2-1} B_2 = d_{i_2} & i_2 \in I_2 \\ \vdots & \vdots \\ d_{i_m-1} B_m = d_{i_m} & i_m \in I_m \end{array} \quad (4)$$

where $|I_1| + |I_2| + \dots + |I_m| = r$ (r is the total number of cycles required, and $|I_i|$ indicates the cardinality of the set I_i) and $B_i \neq B_j$ for $i \neq j$.

We say that merges in Class I go to proper completion (i.e., $n = 1$) and merges in Class II go to improper completion (i.e., $n > 1$). As the definition implies, Class I merges continue using the same merge pattern until completion, i.e., until a single tape has a single sequence and all other tapes are void. In Class II, the merge pattern varies among n possible merge patterns until completion. Class Ia is a subclass of Class I.

Class Ia merges

As an example of a cascade merge⁴ ($i = K - 1$), we have

$$A_{cas} = [a_{st}]$$

where $a_{st} = 1$ if $t \leq K - s + 1$ and $a_{st} = 0$ if $t > K - s + 1$ for $s, t = 1, 2, \dots, K$. Normally, the set of difference equations would be described in the form

$$m_{k,i+1} = \sum_{t=1}^{K-k+1} m_{ti} \quad k = 1, 2, \dots, K \quad j = 1, 2, \dots, r$$

class I

class II

properties

Consider a cascade merge. For the d_{ki} , we have the same set of difference equations, namely,

$$d_{k,i+1} = \sum_{i=1}^{K-k+1} d_{ti} \quad k = 1, 2, \dots, K \quad j = 1, 2, \dots, r$$

Thus, all merge patterns are the same for a cascade merge, that is, $n = 1, m = 1$. In general, we can define the vectors $\mathbf{m}_i$ and $\mathbf{d}_i$ to be

$$\begin{aligned} \mathbf{d}_i &= E^i \mathbf{d}_0 & \mathbf{m}_i &= E^i \mathbf{m}_0 \\ \mathbf{d}_0 A_{cas}^i &= E^i \mathbf{d}_0 & \mathbf{m}_0 A_{cas}^i &= E^i \mathbf{m}_0 \\ \mathbf{d}_i &= \mathbf{d}_0 A_{cas}^i & \mathbf{m}_i &= \mathbf{m}_0 A_{cas}^i \end{aligned}$$

When $A = B$ as for the cascade merge, we say that $\mathbf{d}_i$ and $\mathbf{m}_i$ are *self-duals*.

Define I_{ss} as an *identity matrix* of order s , and 0_{st} as a zero rectangular matrix of dimension s by t . Then we can define a matrix for a polyphase merge ($i = 0$) as follows:

$$A_{pol} = \begin{bmatrix} 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 \\ & & & & & & & 0 \\ & & I_{K-1, K-1} & & & & & 0 \\ & & & & & & & \vdots \\ & & & & & & & \vdots \\ & & & & & & & 0 \end{bmatrix}$$

Similarly, we have for a compromise merge ($0 \leq i \leq K - 1$) the following matrix:

$$\begin{aligned} & (K-i) \quad (K-i+1) \\ & \quad \downarrow \downarrow \\ A_{com_i} &= \begin{bmatrix} 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 & 1 & 1 \\ 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 & 0 & \cdot & \cdot & \cdot & \cdot & 0 & 0 & 0 \\ I_{K-(i+1), K-(i+1)} & & & & & & & 0_{K-(i+1), i+1} & & & & & & & \end{bmatrix} \end{aligned}$$

Clearly, $A_{com_0} = A_{pol}$ and $A_{com_{K-1}} = A_{cas}$.

From the corresponding sets of difference equations, it can be shown that

$$B_{com_i} = A_{com_i}^T \quad 0 \leq i \leq K - 1$$

Since $A_{com_{K-1}}$ is symmetric, we have the self-duality property as already illustrated for A_{cas} . However, for $0 \leq i < K - 1$, the resultant matrix is not symmetric; in this latter case we say that $\mathbf{d}_i$ and $\mathbf{m}_i$ are *duals*. Further, we have that all merge patterns are the same for $0 \leq i \leq K - 1$; namely, only one matrix is required to describe the merge process (i.e., $n = 1, m = 1$).

The defining difference equations for $0 \leq i \leq K - 1$ are

$$|A_{\text{com}_i} - EI| d_{k_i} = 0$$

$$|A_{\text{com}_i} - EI| m_{k_i} = 0 \quad k = 1, 2, \dots, K \quad j \geq 0$$

Much information can be obtained from the powers of A_{com_i} and B_{com_j} . For example, if one desires to know the sequence distribution required on the K tapes so that proper completion will be obtained in r cycles, one is required only to find $A_{\text{com}_i}^r$. The first row of $A_{\text{com}_i}^r$ gives the required original distribution ($\mathbf{m}_r$) for a merge of r cycles; the $(i + 1)$ st row gives the distribution ($\mathbf{m}_{r-1}$) at the end of the first cycle; the $2(i + 1)$ st row gives the distribution ($\mathbf{m}_{r-2}$) at the end of the second cycle, etc., until the K rows are exceeded.

If we desire to know the distribution of the lengths of the sequences at the end of the r th cycle, and assume $d_{10} = d_{20} = \dots = d_{K0} = 1$, we need only to sum the columns in $(A_{\text{com}_i}^T)^r$. In general, we have

$$\mathbf{d}_0(A_{\text{com}_i}^T)^r = \mathbf{d}_r$$

For $r = 0$, of course, $\mathbf{d}_0 = \mathbf{d}_0$.

Let us now determine how many records are handled per cycle for an arbitrary cycle of Class Ia merge-sorts. Assume that we start out with the distribution as described by the first row in $A_{\text{com}_i}^r$. The number of sequences per tape at the end of the first cycle is given by the distribution described in the first row of $(A_{\text{com}_i})^{r-1}$. The lengths of the sequences at the end of the first cycle are described by the elements in $d_0(A_{\text{com}_i})$. Therefore, the total number of records handled in the first cycle becomes

$$S_1 = \sum_{k=1}^{i+1} d_{k1} m_{k,r-1}$$

The total number of records on the K tapes would simply be

$$S_r = d_{1r} m_{10}$$

since $m_{k0} = 0$ for $k > 1$.

In general, the total records handled for the j th cycle is

$$S_j = \sum_{k=1}^{i+1} d_{ki} m_{k,r-j} \quad j = 0, 1, \dots, r$$

If $i = K - 1$, then

$$S_j = (\mathbf{m}_{r-j})(\mathbf{d}_j)^T \quad j = 0, 1, 2, \dots, r$$

and upon substituting for $\mathbf{d}_j$ and $\mathbf{m}_{r-j}$ we obtain

$$S_j = \mathbf{m}_0(A_{\text{com}_i})^r \mathbf{d}_0^T$$

The total number of records is constant for all j and equals the total number of records available, i.e., each cycle handles all of the records. We call the handling of all records a *pass*. For some of the Class Ia merges, a pass is equal to a cycle. Such

merges are ones where $i = K - 1$ or $i = K - 2$, e.g., all cascade merges are of this kind.

The last cycle as exhibited in the Class Ia examples usually is considered as a single step. However, i additional steps may be added to be mathematically consistent with the stated $i + 1$ steps per cycle.

A measure of efficiency of a merge-sort is p , the number of passes required. In the cascade merge, a cycle and pass are equivalent, hence $p = r$. In general, a pass is not equal to a cycle. In this case, we define the number of required passes to be

$$p = \frac{\sum_{i=1}^r S_i}{S_r}$$

To compare the merges, we can compare the plots of the passes required versus the number of records of one type of merge with similar plots of another type. Carter¹ gives several of these plots.

examples *Example 1.* A cascade merge ($i = K - 1$) with $K = 4$ tapes and $r = 5$ cycles is described by a matrix

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

the powers of A and the vectors $\mathbf{d}_i$ and $\mathbf{m}_i$ are

$$A^2 = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad A^3 = \begin{bmatrix} 10 & 9 & 7 & 4 \\ 9 & 8 & 6 & 3 \\ 7 & 6 & 4 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix} \quad A^4 = \begin{bmatrix} 30 & 26 & 19 & 10 \\ 26 & 23 & 17 & 9 \\ 19 & 17 & 13 & 7 \\ 10 & 9 & 7 & 4 \end{bmatrix}$$

$$A^5 = \begin{bmatrix} 85 & 75 & 56 & 30 \\ 75 & 66 & 49 & 26 \\ 56 & 49 & 36 & 19 \\ 30 & 26 & 19 & 10 \end{bmatrix}$$

$$\begin{array}{ll} \mathbf{m}_0 = (1, 0, 0, 0) & \mathbf{d}_0 = (1, 1, 1, 1) \\ \mathbf{m}_1 = (1, 1, 1, 1) & \mathbf{d}_1 = (4, 3, 2, 1) \\ \mathbf{m}_2 = (4, 3, 2, 1) & \mathbf{d}_2 = (10, 9, 7, 4) \\ \mathbf{m}_3 = (10, 9, 7, 4) & \mathbf{d}_3 = (30, 26, 19, 10) \\ \mathbf{m}_4 = (30, 26, 19, 10) & \mathbf{d}_4 = (85, 75, 56, 30) \\ \mathbf{m}_5 = (85, 75, 56, 30) & \mathbf{d}_5 = (246, 216, 160, 85) \end{array}$$

Table 1 Class Ia cascade merge for $i = 3$ and $K = 4$

Step	Number of sequences					Cycle or Pass
	Tape 1	Tape 2	Tape 3	Tape 4	Tape 5	
—	85	75	56	30	0	0
1	55	45	26	0	30	1
2	29	19	0	26	30	
3	10	0	19	26	30	
4	0	10	19	26	30	
1	10	0	9	16	20	2
2	10	9	0	7	11	
3	10	9	7	0	4	
4	10	9	7	4	0	
1	6	5	3	0	4	3
2	3	2	0	3	4	
3	1	0	2	3	4	
4	0	1	2	3	4	
1	1	0	1	2	3	4
2	1	1	0	1	2	
3	1	1	1	0	1	
4	1	1	1	1	0	
1	0	0	0	0	1	5
2	Only one step required					
3						
4						

The total records handled are

$$s_1 = 30 \cdot 4 + 26 \cdot 3 + 19 \cdot 2 + 10 \cdot 1 = 246$$

$$s_2 = 10 \cdot 10 + 9 \cdot 9 + 7 \cdot 7 + 4 \cdot 4 = 246$$

$$s_3 = 4 \cdot 30 + 3 \cdot 26 + 2 \cdot 19 + 1 \cdot 10 = 246$$

$$s_4 = 1 \cdot 85 + 1 \cdot 75 + 1 \cdot 56 + 1 \cdot 30 = 246$$

$$s_5 = 1 \cdot 246 = 246$$

$$p = 5 \quad p/s_5 = 0.0203$$

Table 1 outlines, for a Class Ia cascade merge, the procedure by cycle and step of reading sequences from four (K) tapes and writing merged sequences on a fifth tape until the four tapes are empty and all records reside on a fifth tape in a single sequence.

Table 2 Class Ia compromise merge for $i = 1$ and $K = 4$

Step	Number of sequences					Cycles
	Tape 1	Tape 2	Tape 3	Tape 4	Tape 5	
—	54	47	39	21	0	0
1	33	26	18	0	21	1
2	15	8	0	18	21	
1	7	0	8	10	13	2
2	0	7	8	3	6	
1	3	4	5	0	3	3
2	3	1	2	3	0	
1	2	0	1	2	1	4
2	1	1	0	1	1	
1	0	0	1	0	0	5
2	Only one step required					

Example 2. A compromise merge ($i = 1$) with $K = 4$ and $r = 5$ cycles is described by a matrix A in which the powers of A and vectors $\mathbf{d}_i$ and $\mathbf{m}_i$ are as follows:

$$A^1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad A^2 = \begin{bmatrix} 3 & 3 & 2 & 1 \\ 3 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad A^3 = \begin{bmatrix} 8 & 7 & 6 & 3 \\ 7 & 6 & 5 & 3 \\ 3 & 3 & 2 & 1 \\ 3 & 2 & 2 & 1 \end{bmatrix}$$

$$A^4 = \begin{bmatrix} 21 & 18 & 15 & 8 \\ 18 & 16 & 13 & 7 \\ 8 & 7 & 6 & 3 \\ 7 & 6 & 5 & 3 \end{bmatrix} \quad A^5 = \begin{bmatrix} 54 & 47 & 39 & 21 \\ 44 & 41 & 34 & 18 \\ 21 & 18 & 15 & 8 \\ 18 & 16 & 13 & 7 \end{bmatrix}$$

$$\begin{aligned} \mathbf{m}_0 &= (1, 0, 0, 0) & \mathbf{d}_0 &= (1, 1, 1, 1) \\ \mathbf{m}_1 &= (1, 1, 1, 1) & \mathbf{d}_1 &= (4, 3, 1, 1) \\ \mathbf{m}_2 &= (3, 3, 2, 1) & \mathbf{d}_2 &= (9, 8, 4, 3) \\ \mathbf{m}_3 &= (8, 7, 6, 3) & \mathbf{d}_3 &= (24, 21, 9, 8) \\ \mathbf{m}_4 &= (21, 18, 15, 8) & \mathbf{d}_4 &= (62, 54, 24, 21) \\ \mathbf{m}_5 &= (54, 47, 39, 21) & \mathbf{d}_5 &= (161, 140, 62, 54) \end{aligned}$$

Table 3 Class Ia polyphase merge for $i = 0$ and $K = 4$

Step	Number of sequences					Cycles
	Tape 1	Tape 2	Tape 3	Tape 4	Tape 5	
—	56	52	44	29	0	0
1	27	23	15	0	29	1
2	12	8	0	15	14	2
3	4	0	8	7	6	3
4	0	4	4	3	2	4
5	2	2	2	1	0	5
6	1	1	1	0	1	6
7	0	0	0	1	0	7

The total records handled are

$$s_1 = 4 \cdot 29 = 116$$

$$s_2 = 7 \cdot 15 = 105$$

$$s_3 = 13 \cdot 8 = 104$$

$$s_4 = 25 \cdot 4 = 100$$

$$s_5 = 49 \cdot 2 = 98$$

$$s_6 = 94 \cdot 1 = 94$$

$$s_7 = 181 \cdot 1 = 181$$

$$p = 798/181 = 4.41 \quad p/s_7 = 0.0244$$

Table 3 shows, for a polyphase merge, the procedure for reading sequences and writing merged sequences until all records are on a single tape in a single sequence.

Asymptotic solutions to difference equations

The solution to the set of difference equations for a cascade merge ($i = K - 1$) for d_{ki} and m_{ki} may be written as

$$d_{ki} = c_{k1}\lambda_1^i + c_{k2}\lambda_2^i + \cdots + c_{kK}\lambda_K^i$$

where the $\lambda_1, \lambda_2, \dots, \lambda_K$ are the distinct real roots of $|A_{\text{cas}} - \lambda I| = g_K(\lambda) = 0$. The author⁵ has demonstrated that the characteristic values for a (0, 1)-matrix having the form of the matrix used

for the cascade merge in Class Ia are real and distinct and have the property for K odd,

$$\lambda_1 > \lambda_3 > \cdots > \lambda_K > 0 > \lambda_{K-1} > \cdots > \lambda_2$$

and for K even,

$$\lambda_1 > \lambda_3 > \cdots > \lambda_{K-1} > 0 > \lambda_K > \cdots > \lambda_2$$

where

$$\lambda_1 > |\lambda_2| \geq |\lambda_3| \geq |\lambda_4| \cdots \geq |\lambda_K| \geq 0$$

Asymptotic solutions to the set of difference equations can provide an approximation to the number of records m_{kj} stored on a particular tape k during each merge cycle j . Similarly, the length of each sequence d_{kj} can be approximated for each k and j . Carter¹ determined that the solution to the set of difference equations representing the cascade merge had distinct roots for only a limited number of examples. Results presented in this paper extend Carter's work for all cascade merges of Class Ia. Hence, according to Carter, the asymptotic approximation is

$$d_{kj} = c_{k1} \lambda_1^j$$

Similarly, for m_{kj} we have

$$m_{kj} = c_{k1} \lambda_1^j$$

Carter shows that the coefficients in these expressions can be obtained from K initial conditions and the relationship

$$c_{k1} = \left(\sum_{i=0}^{K-1} d_{ki} \delta_i \right) / g'_K(\lambda_1)$$

where δ_i are the coefficients of

$$g_K(\lambda) / (\lambda - \lambda_1) = \sum_{i=1}^K \delta_i \lambda^{i-1}$$

and $g'_K(\lambda_1)$ is the derivative of $g_K(\lambda)$ evaluated at $\lambda = \lambda_1$.

For polyphase merge sorts, one can show that

$$|A_p - \lambda I| = g_K(\lambda) = \lambda^K - \lambda^{K-1} - \lambda^{K-2} - \cdots - \lambda - 1 = 0$$

The characteristic equation for a polyphase merge of K tapes onto one tape is lemma 1

$$(-1)^K (\lambda^K - \lambda^{K-1} - \lambda^{K-2} - \cdots - \lambda - 1) = 0.$$

Proof: It previously has been shown in this paper that the matrix defining the polyphase merge of Class Ia is

$$A_p = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\ & & & & & 0 \\ & & & & & 0 \\ & & I_{K-1, K-1} & & & \vdots \\ & & & & & \vdots \\ & & & & & 0 \end{bmatrix}$$

For $K = 1$, the characteristic equation is $1 - \lambda = 0$. If we assume that the hypothesis is true for $K = k$, then we can show that it is true for $K = k + 1$. Thus,

$$\begin{aligned}
 |A_p - \lambda I| &= \begin{vmatrix} 1 - \lambda & 1 & 1 & \cdots & 1 & 1 & 1 \\ 1 & -\lambda & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -\lambda & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -\lambda & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -\lambda \end{vmatrix} = g_{k+1}(\lambda) \\
 &= (-\lambda)g_k(\lambda) - 1(-1)^{k-1} \\
 &= (-1)^{k+1}(\lambda^{k+1} - \lambda^k - \lambda^{k-1} - \cdots - \lambda) + (-1)^k(-1)^2 \\
 &= (-1)^{k+1}(\lambda^{k+1} - \lambda^k - \lambda^{k-1} - \cdots - \lambda - 1)
 \end{aligned}$$

lemma 2 The roots of the characteristic equation for a polyphase merge in Class Ia are simple. Also, the equation has one root λ_1 , lying between 1 and 2 on the real axis and approaching 2 as K increases. All remaining roots lie within the unit circle.

Proof: Assume that the equation of interest is

$$g_k(\lambda) = \lambda^k - \lambda^{k-1} - \cdots - \lambda - 1 = 0$$

First, multiply by $(\lambda - 1)$ to obtain

$$f_{k+1}(\lambda) = \lambda^{k+1} - 2\lambda^k + 1 = 0$$

Since the column sums are 1 and 2, we know from the original matrix that there exists a positive real root λ_1 such that $1 \leq \lambda_1 \leq 2$ and λ_1 approaches 2 as K increases. Further, we know that $\lambda_1 \neq 1$ by the substitution of $\lambda_1 = 1$ into the first equation. A theorem attributed to Pellet⁶ states that if, for $f(\lambda) = a_0 + a_1\lambda + \cdots + a_p\lambda^p + \cdots + a_n\lambda^n$ where $a_p \neq 0$, then the equation

$$\begin{aligned}
 F_p(\lambda) &\equiv |a_0| + \cdots + |a_{p-1}| \lambda^{p-1} - |a_p| \lambda^p \\
 &\quad + |a_{p+1}| \lambda^{p+1} + \cdots + |a_n| \lambda^n = 0
 \end{aligned}$$

has two positive zeros r_1 and r_2 , where $r_1 < r_2$. Also, $f(\lambda)$ has exactly p zeros in or on the circle $|\lambda| \leq r_1$, and no zeros can be in the annular ring $r_1 < |\lambda| < r_2$. This theorem directly applies since there is a root $\lambda_2 = 1$ and $1 < \lambda_1 \leq 2$. Hence, all λ_i , $i \neq 1, 2$, lie within the unit circle.

Next, take the derivative of $f_{k+1}(\lambda)$ to obtain

$$f'_{k+1}(\lambda) = (k+1)\lambda^k - 2k\lambda^{k-1} = 0$$

We know that if $f_{k+1}(\lambda)$ has multiple roots, then these roots are also roots of $f'_{k+1}(\lambda)$. The roots of $f'_{k+1}(\lambda)$ consist of $k - 1$ zero roots and one root of value $2k/k + 1$, but $2k \geq k + 1$ for $k \geq 1$. We have said that neither $\lambda_2 = 1$ nor $1 < \lambda_1 \leq 2$ is a multiple root. Thus, there are no multiple roots.

As for the cascade merge, we can obtain asymptotic approximations for the polyphase merge, i.e.,

$$d_{k_i} = c_{k_i} \lambda_1^i \quad \text{and} \quad m_{k_i} = c'_{k_i} \lambda_1^i$$

where again the c_{k_i} and c'_{k_i} can be determined from the initial conditions.

Sequence patterns in strings

Another problem readily handled by the manipulation of matrices is that of determining the arrangement of the distributed sequences into the required descending and ascending pattern to permit the merge to go to completion. In the A_{com_i} matrices, the presence of an $I_{K-(i+1), K-(i+1)}$ block identity matrix was indicated. Let this block of 0's and 1's remain unchanged. All other 1's in A are replaced by α . Call the resultant matrix A_α . Let α designate an operator, which upon operating on a sequence $a_1 a_2 \cdots a_k \cdots a_{n-1} a_n$, results in a sequence $a_n^{-1} a_{n-1}^{-1} \cdots a_k^{-1} \cdots a_2^{-1} a_1^{-1}$.

The sequences being merged are actually strings of ascending (a) and descending (d) sequences. Thus, $a^{-1} = d$ and $d^{-1} = a$. Assume that we wish to end with an ascending sequence. To determine the desired pattern at the beginning of the merge with r cycles, we use the following expressions:

$$(\cdots (\underbrace{(a \ 0 \ 0 \ \cdots \ 0)}_K A_\alpha) \underbrace{A_\alpha}_{r} \cdots A_\alpha) \quad (5a)$$

or

$$(A_\alpha^T \cdots (A_\alpha^T (\underbrace{A_\alpha^T (a \ 0 \ 0 \ \cdots \ 0)^T}_K) \cdots)^T) \quad (5b)$$

Observe that associativity is not present when raising A_α to a power, i.e.,

$$(A_\alpha \cdot A_\alpha) A_\alpha \neq A_\alpha (A_\alpha \cdot A_\alpha)$$

Expression 5a shows right multiplication by A_α , and 5b shows left multiplication by A_α . If we let R and L designate right and left multiplication, we have

$$\left[(A_\alpha^T)_L^r \begin{bmatrix} a \\ 0 \\ \vdots \\ 0 \end{bmatrix} \right]^T = (a \ 0 \ \cdots \ 0) (A_\alpha)_R^r$$

Therefore, either method may be used to obtain the correct results.

Instead of the usual addition operation in vector-matrix multiplication we use juxtaposition. Observe that α operating on α is 1 and α operating on 1 is α . Thus,

$$\alpha(1 \ 1 \ \alpha \ 1 \ \alpha) = (1 \ \alpha \ 1 \ 1 \ \alpha \ \alpha)$$

Looking at the pattern made by the sequences for a cascade merge, we have as the elements of $(A_{\alpha\alpha\alpha})_K^r$ sequences of all 1's or all α 's depending upon whether r is even or odd. Therefore, if all sequences from the presort are distributed on the K tapes in an ascending fashion, then the merge will go to completion in a correct manner, and the resultant merged sequence will be descending or ascending depending upon whether the number of cycles is odd or even.

For the polyphase case in Class Ia, the patterns made by the string of sequences in $(A_{\alpha\alpha})_K^r$ alternate between the 1's and the α 's.

lemma 3

For a polyphase merge of K tapes, the first row in $(A_p)^r$ consists of all odd integers when $r = 1 + j(K + 1)$ for any $j \geq 0$, and consists of $K - 1$ even integers and one odd integer otherwise. Furthermore, when $r \neq 1 + j(K + 1)$, the odd integer occurs for each consecutive r in a different column.

Proof: For any K for $j = 0$ there is a row of all 1's (all odd integers); for $r = 2$ there are even integers in the first $K - 1$ columns and an odd integer in the last. That is, for $r = 1$, there is the following matrix:

$$\begin{bmatrix} 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & 1 \\ & & & & & & & 0 \\ & & & & & & & \vdots \\ & & I_{K-1, K-1} & & & & & \vdots \\ & & & & & & & 0 \end{bmatrix}$$

and for $r = 2$, the following matrix:

$$\begin{bmatrix} 2 & 2 & 2 & \cdot & \cdot & \cdot & 2 & 1 \\ 1 & 1 & 1 & \cdot & \cdot & \cdot & 1 & 1 \\ & & & & & & & 0 & 0 \\ & & & & & & & \vdots & \vdots \\ & & I_{K-2, K-2} & & & & & \vdots & \vdots \\ & & & & & & & 0 & 0 \end{bmatrix}$$

Since, for the polyphase merge, each i th row is the first row for $[A_p]^{r-i+1}$, and the first row in $[A_p]^r$ is obtained by adding up all integers in each column in $[A_p]^{r-1}$, we have the result for $j = 0$ and $r = 1$ to $r = K + 1$. For $r = K + 1$, the first row has an odd integer in the first column, and the last row has an odd integer in the last column; i.e., all odd elements are on the principal diagonal. Therefore, since each column has exactly one odd element, the first row in A^r , where $r = K + 1 + 1$, has all odd elements. By assuming the hypothesis for any j we can show, in a like manner, that it is true for $j + 1$. Hence, the lemma is proved for j by induction.

lemma 4

If $r \neq 1 + j(K + 1)$ for any $j \geq 0$, then the set of sequences generated by $(A_{\alpha})_K^r$ is identical to one set of those that can be generated by starting with a set of K sequences $B, B, \dots, B, B^{-1}$ (left juxtapositioning the inverse to each element of the set).

Repeat this step until a string of alternating ordered sequences of length agreeing with that given by the first row in A^r is built up,⁷ where B is either a or d . This step is repeated until the desired r is reached. For example, start with $a, a, \dots, a, d$. Juxtapose the inverse for $r = 2$ to obtain $da, da, \dots, da, d$ (for the last element, a is not juxtaposed to d because the length is restricted to length one). Juxtapose the inverse for $r = 3$ to obtain $dada, dada, \dots, dada, ada, ad$.

Proof: If $r = 2$, then the first row of $(A_\alpha)_R^2$ is $1\alpha, 1\alpha, \dots, 1\alpha, 1$. If $r = 3$, then the first row of $(A_\alpha)_R^3$ is $1\alpha 1\alpha, 1\alpha 1\alpha, \dots, 1\alpha 1\alpha, 1\alpha 1, 1\alpha$. In general, the strings with an odd number of sequences must start with a 1, whereas all others begin (on the right) with α . When $r = 1 + k + 1$, then the first row of $(A_\alpha)_R^r$ is made up of strings all with an odd number of sequences that begin with a 1. However, this condition violates the method of generating the strings. In general, each element of the first row of $(A_\alpha)_R^r \equiv [d_{st}]$ is obtained by taking the corresponding column in $(A_\alpha)_R^{r-1} \equiv [c_{st}]$ and forming the following right juxtaposition

$$d_{1t} = \alpha(c_{1t}) \cdot \alpha(c_{2t}) \cdot \dots \cdot \alpha(c_{kt})$$

Since at all times the leftmost elements in all strings are all 1's, if $K - 1$ strings are made up of an even number of sequences and alternate in their order, these $K - 1$ strings must begin with α . Thus, if the string has an odd number of sequences and ends with a 1, then it must begin with a 1.

Therefore, from Lemma 3, if r is prevented from being equal to $[1 + j(K + 1)]$, the original pattern of the sequences distributed on the K tapes can be generated from the presort by starting with B , alternating between B^{-1} and B , and ending with a B^{-1} if the number of sequences on the tape is to be even. Start with a B^{-1} , alternate between B and B^{-1} , and end with a B^{-1} if the number is odd. In this manner, proper completion is guaranteed.

We now observe the patterns for a compromise merge. Assume that $i = 1$, then

$$A_\alpha = \begin{bmatrix} \alpha & \alpha & \cdot & \cdot & \cdot & \cdot & \alpha & \alpha \\ \alpha & \alpha & \cdot & \cdot & \cdot & \cdot & \alpha & 0 \\ & & & & & & 0 & 0 \\ & & & & & & \vdots & \vdots \\ & I_{K-2, K-2} & & & & & \vdots & \vdots \\ & & & & & & 0 & 0 \end{bmatrix} \quad A_\alpha^2 = \begin{bmatrix} 11\alpha & 11\alpha & \cdot & \cdot & \cdot & \cdot & 11\alpha & 11\alpha & 11 & 1 \\ 11\alpha & 11\alpha & \cdot & \cdot & \cdot & \cdot & 11\alpha & 11 & 11 & 1 \\ \alpha & \alpha & \cdot & \cdot & \cdot & \cdot & \alpha & \alpha & \alpha & \alpha \\ \alpha & \alpha & \cdot & \cdot & \cdot & \cdot & \alpha & \alpha & \alpha & \alpha \\ & & & & & & 0 & 0 & 0 & 0 \\ & & & & & & \vdots & \vdots & \vdots & \vdots \\ & I_{K-4, K-4} & & & & & 0 & 0 & 0 & 0 \end{bmatrix}$$

The first row of A_α^3 would be

$$1\alpha\alpha 1\alpha\alpha 11\alpha, 1\alpha\alpha 1\alpha\alpha 11\alpha, \dots, 1\alpha\alpha 1\alpha\alpha 11\alpha, 1\alpha\alpha 1\alpha\alpha 11, 1\alpha\alpha\alpha\alpha 11, \alpha\alpha\alpha\alpha 11, \alpha\alpha 1.$$

Thus, we see that there are no simple means of generating a string of sequences such that the resultant pattern is valid for proper completion for most r 's.

However, for any Class Ia merge-sort, if we add $K - i + 1$ 1-way merge or copy steps¹¹ to each cycle of $i + 1$ steps, then (as for the cascade merge) the pattern consists of all (a)'s or all (d)'s, and proper completion is guaranteed. In such a case, the A_α matrix is obtained from the A matrix by substituting α 's for all 1's.

Extension of the analysis

The method that we have used for analyzing Class Ia merge-sorts can be extended to other classes of merge-sorts that go to either proper or improper completion. For example, a balanced merge in which K tapes are merged onto K output tapes can be described by the set of difference equations

$$\begin{aligned} \mathbf{d}_0 A^i &= \mathbf{d}_i & j &= 0, 1, 2, \dots, r \\ \mathbf{m}_0 (A^T)^i &= \mathbf{m}_i & j &= 0, 1, 2, \dots, r \end{aligned}$$

where all elements of A are 1's. Here again, the first row in A^k gives the distribution on the K tapes at the end of $r - k$ cycles. Each cycle can be interpreted as having K steps, and for the cascade merge, no attention need be paid to the patterns of the strings of sequences if all are put into the same ordering.

In the balanced method, if initially we have K_1 input tapes and K_2 output tapes and then K_2 input tapes and K_1 output tapes (with this scheme repeated in a like manner), then we can represent the balanced merge-sort by a square matrix A of dimension K_1 and all its elements by K_2 . For such a merge, there are $k_1 \times k_2$ steps. For each element K_2 in A , if we substitute a sequence of $K_2 \alpha$'s, then we obtain the desired A_α .

Let us consider merges belonging in either Class I or Class II and not restricted to Class Ia. A *general polyphase merge* is described as a merge such that on each of K input tapes at any level $r - j$ there are distributed m_{ij} ($i = 1, 2, \dots, K$) sequences, and the next level down, i.e., the $(r - j + 1)$ st level, has the following number of sequences distributed on K tapes: $m_{1j} - m_{1j}$, $m_{2j} - m_{1j}$, $\dots$, $m_{i-1,j} - m_{1j}$, m_{ij} , $0, \dots, 0$. Here, m_{ij} is the smallest positive number of sequences on the K tapes at the $r - j$ th level. General compromise and cascade merges may be similarly defined.

For the entire merge of Class II, sets of difference equations as illustrated in Equations 3 and 4 can be established. If for some j and for some i , $\mathbf{m}_{i-1} A_i = \mathbf{m}_i$, then $\mathbf{d}_{r-1} A_i^T = \mathbf{d}_{r-i+1}$. That is, for any one merge cycle the set of difference equations for the $\mathbf{m}_i$ is the dual of the set for the $\mathbf{d}_{r-i+1}$. Therefore, in general, Equations 3 and 4 become

$$\begin{aligned} \mathbf{m}_{i_1-1} A_1 &= \mathbf{m}_{j_1} & j_1 &\in J_1 \\ \mathbf{m}_{i_2-1} A_2 &= \mathbf{m}_{j_2} & j_2 &\in J_2 \\ &\vdots & &\vdots \\ \mathbf{m}_{i_n-1} A_n &= \mathbf{m}_{j_n} & j_n &\in J_n \end{aligned} \quad (6)$$

$$\begin{array}{ll}
d_{r-i_1+1} = d_{r-i_1} A_1^T & j_1 \in J_1 \\
d_{r-i_2+1} = d_{r-i_2} A_2^T & j_2 \in J_2 \\
\vdots & \vdots \\
d_{r-i_n+1} = d_{r-i_n} A_n^T & j_n \in J_n
\end{array} \quad (7)$$

For the merges in Class Ia, we have no problem in determining the patterns of descending and ascending strings of sequences because the d_{ij} are in a definite order, i.e., $d_{1i} \geq d_{2i} \geq \dots \geq d_{Ki}$, and the row sums of the associated matrix produce a K -length vector $\mathbf{S} = (K, K-1, \dots, K-i, 1, 1, \dots, 1)$. This condition also agrees with the descriptions of the general compromise, polyphase, and cascade merges. In general, one cannot expect the ordering of the elements of the vectors $\mathbf{m}_i$, $\mathbf{d}_i$, and $\mathbf{S}$ to be as desired. Therefore, for the general merge in Class I and Class II, the ordering of the elements of $\mathbf{m}_i$ and $\mathbf{d}_i$ depends upon the ordering of the elements of $\mathbf{S}$, as determined from the description of the merge.

Two examples are now given for an improper polyphase merge of Class II.

examples

$$A_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \begin{cases} \mathbf{S} = (3, 1, 1) \\ m_{1i} \geq m_{2i} \geq m_{3i} \geq 0 \\ d_{1i} \geq d_{2i} \geq 0 \end{cases}$$

Example 1

Example 2

$$A_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{S} = (3, 1, 1) \quad A_2 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{S} = (2, 1, 1)$$

	Tape no.					Tape no.			
	1	2	3	4		1	2	3	4
Number of sequences	19	16	10	0	Number of sequences	20	17	11	0
on each	9	6	0	10	on each	9	6	0	11
tape at the	3	0	6	4	tape at the	3	0	6	5
end of	0	3	3	1	end of	0	3	3	2
each cycle	1	2	2	0	each cycle	2	1	1	0
	0	1	1	1		1	0	0	1
	1	0	0	0		0	0	1	0

$\mathbf{m}_0 A_1 = \mathbf{m}_1$	$\mathbf{d}_0 A_1^T = \mathbf{d}_1$	$\mathbf{m}_0 A_2 = \mathbf{m}_1$	$\mathbf{d}_0 A_1 = \mathbf{d}_1$
$\mathbf{m}_1 A_2 = \mathbf{m}_2$	$\mathbf{d}_1 A_1^T = \mathbf{d}_2$	$\mathbf{m}_1 A_1 = \mathbf{m}_2$	$\mathbf{d}_1 A_1 = \mathbf{d}_2$
$\mathbf{m}_2 A_2 = \mathbf{m}_3$	$\mathbf{d}_2 A_1^T = \mathbf{d}_3$	$\mathbf{m}_2 A_1 = \mathbf{m}_3$	$\mathbf{d}_2 A_1 = \mathbf{d}_3$
$\mathbf{m}_3 A_1 = \mathbf{m}_4$	$\mathbf{d}_3 A_2^T = \mathbf{d}_4$	$\mathbf{m}_3 A_1 = \mathbf{m}_3$	$\mathbf{d}_3 A_1 = \mathbf{d}_4$
$\mathbf{m}_4 A_1 = \mathbf{m}_5$	$\mathbf{d}_4 A_2^T = \mathbf{d}_5$	$\mathbf{m}_4 A_1 = \mathbf{m}_5$	$\mathbf{d}_4 A_1 = \mathbf{d}_5$
$\mathbf{m}_5 A_1 = \mathbf{m}_6$	$\mathbf{d}_5 A_1^T = \mathbf{d}_6$	$\mathbf{m}_5 A_1 = \mathbf{m}_6$	$\mathbf{d}_5 A_2 = \mathbf{d}_6$

	Tape no.					Tape no.			
	1	2	3	4		1	2	3	4
Length of sequences	1	1	1	0	Length of sequences	1	1	1	0
on each tape at the end of each cycle	1	1	0	3	on each tape at the end of each cycle	1	1	0	3
	1	0	5	3		1	0	5	3
	0	9	5	3		0	9	5	3
	17	9	5	0		17	9	5	0
	0	9	5	31		17	⑨	0	31
	45	⑨	⑤	0		48	③①	①⑦	0

In these examples we see that

$$A_{1\alpha} = \begin{bmatrix} \alpha & \alpha & \alpha \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (A_{2\alpha})_1 = \begin{bmatrix} \alpha & \alpha & \alpha \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (A_{2\alpha})_2 = \begin{bmatrix} \alpha & \alpha & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

As another example of an "unorthodox" merge-sort, select a matrix A whose row sums agree with the general compromise description for $i = 1$. (The m_{ij} in this illustration are not necessarily ordered as in Class Ia, but the d_{ij} are ordered.)

Table 4 Class I compromise merge

Step	Number of sequences					Cycles
	Tape 1	Tape 2	Tape 3	Tape 4	Tape 5	
—	86	37	53	76	0	0
1	49	0	16	39	37	1
2	33	16	0	23	37	
1	17	0	16	7	21	2
2	10	7	16	0	14	
1	3	0	9	7	7	3
2	0	3	6	7	4	
1	3	0	3	4	1	4
2	3	1	2	3	0	
1	2	0	1	2	1	5
2	1	1	0	1	1	
1	0	0	1	0	0	6
2	Only one step required					

Let A be

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

then we have the following information concerning this merge:

$$A^2 = \begin{bmatrix} 3 & 1 & 2 & 3 \\ 2 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad A^3 = \begin{bmatrix} 7 & 3 & 4 & 6 \\ 5 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \\ 3 & 1 & 2 & 3 \end{bmatrix} \quad A^4 = \begin{bmatrix} 16 & 7 & 10 & 14 \\ 11 & 5 & 7 & 10 \\ 3 & 1 & 2 & 3 \\ 7 & 3 & 4 & 6 \end{bmatrix}$$

$$A^5 = \begin{bmatrix} 37 & 16 & 23 & 33 \\ 26 & 11 & 16 & 23 \\ 7 & 3 & 4 & 6 \\ 16 & 7 & 10 & 14 \end{bmatrix} \quad A^6 = \begin{bmatrix} 86 & 37 & 53 & 76 \\ 60 & 26 & 37 & 53 \\ 16 & 7 & 10 & 14 \\ 37 & 16 & 23 & 33 \end{bmatrix}$$

$d_0 = (1, 1, 1, 1)$	$m_0 = (1, 0, 0, 0)$	$s_1 = 4 \cdot 37 + 3 \cdot 16 = 196$
$d_1 = (4, 3, 1, 1)$	$m_1 = (1, 1, 1, 1)$	$s_2 = 9 \cdot 16 + 6 \cdot 7 = 186$
$d_2 = (9, 6, 1, 4)$	$m_2 = (3, 1, 2, 3)$	$s_3 = 20 \cdot 7 + 14 \cdot 3 = 182$
$d_3 = (20, 14, 4, 9)$	$m_3 = (7, 3, 4, 6)$	$s_4 = 47 \cdot 3 + 33 \cdot 1 = 174$
$d_4 = (47, 33, 9, 20)$	$m_4 = (16, 7, 10, 14)$	$s_5 = 109 \cdot 1 + 76 \cdot 1 = 185$
$d_5 = (109, 76, 20, 47)$	$m_5 = (37, 16, 23, 33)$	$s_6 = 252 \cdot 1 = 252$
$d_6 = (252, 176, 47, 109)$	$m_6 = (86, 37, 53, 76)$	$p = 4.66 \quad p/s_5 = 0.0185$

In this example, the A_α matrix operator used to determine the pattern of ascending and descending strings is

$$A_\alpha = \begin{bmatrix} \alpha & \alpha & \alpha & \alpha \\ \alpha & 0 & \alpha & \alpha \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Table 4 shows the pattern of reading sequences and writing merged sequences for an example of a Class I compromise merge sort. Observe the similarity between the last example and that of the Class Ia compromise merge (Table 2) for $i = 1$ and $K = 4$.

It is evident that the Class Ia compromise merge is the more efficient. In terms of (0, 1)-matrix notation, the matrices

$$A_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

belong to the same class,¹² namely, $\mathfrak{u}(\mathbf{S}, \mathbf{T})$, as determined by equivalent ordered sets of *row sums* $\mathbf{S}$ and unordered sets of *column sums* $\mathbf{T}$. The ordering of the row sums is, of course, determined by the description of the merge. In the above example, since the merge is a general compromise merge with $i = 1$, $\mathbf{S} = (4, 3, 1, 1)$ and $\mathbf{T} = (3, 3, 2, 1)$, the size of the set $\mathfrak{u}(\mathbf{S}, \mathbf{T})$ is less than or equal to 4 factorial.

Each member of Class Ia establishes a unique set $\mathfrak{u}(\mathbf{S}, \mathbf{T})$. The preceding analysis supports the proposition that members of Class Ia are more efficient merge-sorts than the other members of the sets $\mathfrak{u}(\mathbf{S}, \mathbf{T})$. It is further proposed that if we let $A_1 \in$ Class Ia and $A_i \in \mathfrak{u}(\mathbf{S}, \mathbf{T})$, then

$$|A_1 - \lambda I| = 0$$

has a greater maximum positive root than

$$|A_i - \lambda I| = 0 \qquad i \neq 1$$

However, each member of $\mathfrak{u}(\mathbf{S}, \mathbf{T})$ can be described by the same set of merge procedures (e.g., the general statement described above for the polyphase merge), and all members of this class go to proper completion.

Summary

A classification of merge-sorts has been introduced in which members of Class Ia for $i = 0$ and $i = k - 1$ agree with Carter's definition of polyphase and cascade merge-sorts. For $0 < i < k - 1$, the defined merge-sorts are classified in the general category of compromise merge-sorts. Class Ia and the defined larger Class I merge-sorts are shown to have interesting properties, the most important being that members of these classes can be characterized by a single matrix. Interpreted, this property indicates that the particular merge can go to completion using the same procedure throughout the merge. This property defines a merge going to proper completion. The introduction of matrix descriptions of Classes I and Ia merge-sorts is useful since methods of matrix manipulation are well known. By defining new operators for the matrices, it is shown how patterns of ascending and descending sequences required as a result of the presort phase can be determined.

Class II merge-sorts require more than one matrix to describe and become more difficult to analyze as the number of describing matrices becomes larger. For the Class II merge-sorts, the merge

does not go to completion using the same merge procedure. This property defines a merge going to improper completion.

For those classes analyzed in this paper, it is asserted that the merge-sorts of most interest are in the specially defined Class Ia.

ACKNOWLEDGMENT

The author wishes to thank S. W. Reynolds and J. W. Toner for constructive discussions relating to the general merging problem.

CITED REFERENCES AND FOOTNOTES

1. W. C. Carter, "Mathematical analysis of merge-sorting techniques," *Information Processing 1962, Proceedings of IFIP 1962*, North-Holland Publishing Co., Amsterdam, 62-66 (1963).
2. D. E. Knuth, Letters to the Editor, *Communications of the Association for Computing Machinery* **6**, 585-587 (October 1963).
3. In the literature, it is usually assumed that the length of the sequences is a constant at the output of the presort, although in reality, it may not be true. The argument used is that, by assuming the sequence length to be a constant (the shortest possible length), one is assuming the worst case. As a consequence of the equal-length assumption, we set $d_{10} = d_{20} = \dots = d_{K0}$.
4. Cascade, polyphase, or compromise merge-sorts are members of Class Ia unless otherwise explicitly stated. In their broader context, these merge-sorts are also members of Classes I and II.
5. C. E. Radke, "Classes of matrices with distinct, real characteristic values," (submitted to the Society of Industrial and Applied Mathematics for publication).
6. M. Marden, *The Geometry of the Zeros of a Polynomial in a Complex Variable*, American Mathematical Society, Providence, R. I. (1949).
7. There are several different ways of constructing sequences to the desired level when the read-backward capability is not available (see S. W. Reynolds⁸ and A. G. Mendoza⁹). R. L. Gilstad¹⁰ discusses the polyphase merge with read-backward capability.
8. S. W. Reynolds, "A generalized polyphase merge algorithm," *Communications of the Association for Computing Machinery* **4**, 347-349 (August 1961).
9. A. G. Mendoza, "A dispersion pass algorithm for the polyphase merge," *Communications of the Association for Computing Machinery* **5**, 502-504 (October 1962).
10. R. L. Gilstad, "Read-backward polyphase sorting," *Communications of the Association for Computing Machinery* **6**, 220-223 (May 1963).
11. A rewind would function as well; we would then merge in the same manner as when using the read-forward capability only.
12. Note that this set is not identical to that described by Ryser;¹³ rather it is a larger set.
13. H. J. Ryser, "Matrices of zeros and ones," *Bulletin of the American Mathematical Society* **66**, 442-464 (November 1960).