STA N-CS-71-238

SU-SEL-71-051

A Lower Bound for Sorting Networks that Use the Divide-Sort-Merge Strategy

bY

David C. Van Voorhis

August 1971

Technical Report No. 17

This work was conducted while the author was a National Science Foundation graduate fellow and was partially supported by the Joint Services Electronics Program U.S. Army, U.S. Navy, and U.S. Air Force under contract N-00014-67-A-0112-0044 and by the National Science Foundation under grant GJ 1180.

DIGITAL SYSTEMS LABORATORY

STANFORD ELECTRONICS LABORATORIES

STANFORD UNIVERSITY . STANFORD, CALIFORNIA

STAN-CS-71-238 SEL-71-051

A LOWER BOUND FOR SORTING NETWORKS THAT USE THE DIVIDE-SORT-MERGE STRATEGY

by

David C. Van Voorhis

August 1971

Technical Report no. 17

DIGITAL SYSTEMS LABORATORY
Stanford Electronics Laboratories - Computer Science Department
Stanford University
Stanford, California

This work was conducted while the author was a National Science Foundation graduate fellow and was partially supported by the Joint Services Electronics Program U.S. Army, U.S. Navy, and U.S. Air Force under contract N-00014-67-A-0112-0044 and by the National Science Foundation under grant GJ 1180.

A LOWER BOUND FOR SORTING NETWORKS THAT USE THE DIVIDE-SORT-MERGE STRATEGY

by

David C. Van Voorhis

ABSTRACT

Let $M_g(g^{k+1})$ represent the minimum number of comparators required by a network that merges g sorted multisets containing g^k members each. In this paper we prove that $M_g(g^{k+1}) \geq g\, M_g(g^k) + g^{k-1} \Sigma_{\ell=2}^{g} \lfloor(\ell-1)g/\ell\rfloor$. From this relation we are able to show that an N-sorter network which *uses* the g-way **divide-**sort-merge strategy must contain at least order $N(\log_2 N)^2$ comparators.

A network with N inputs and N outputs is called an N-sorter network, or simply an N-sorter, if for any **multiset**[*] of inputs $I = \{i_1, i_2, \ldots, i_N\}$ it produces as output the **multiset** $0 = \{o_1, o_2,$ ⬮ ⌛⌛📫▮📖 where: 1) 0 is a permutation of I; and 2) $o_j \leq o_k$ if $j \leq k$. R. C. Bose and R. J. Nelson [2] have suggested constructing sorting networks using ranks of a basic comparator cell, which is essentially a **2-sorter.** For example, Fig. 1 depicts a b-sorter network that uses 5 comparators labeled A,B,C,D,E. (Note that comparators A-D move the smallest input to o_1 and the largest input to o_4, and then comparator E orders the remaining two inputs.)

From an engineering viewpoint it may be desirable to use as few comparators as possible when constructing an N-sorter, (An alternate design objective would be to minimize the delay required to sort N items.) Let S(N) represent the minimum number of comparators required by a network that sorts N inputs. R. W. Floyd and D. E. Knuth [3] have determined, S(N) for $N \leq 8$ by proving a lower bound for S(N) that is precisely equal to the number of comparators actually contained in the most economical N-sorter known. However, for $N > 8$, the value of $S(N)$ and even the asymptotic behavior of the function remain an open question. The strongest lower bound known for $S(N)$ increases as $N(\log_2 N)$, whereas the strongest upper bound known -- i.e. the number of comparators actually required by the most economical N-sorter yet constructed -- increases as $N(\log_2 N)^2$. (See D. Van Voorhis [4 , 5].)

[*] A **multiset** is like a set except that it may contain repetitions of elements. See D. E. Knuth [1] .

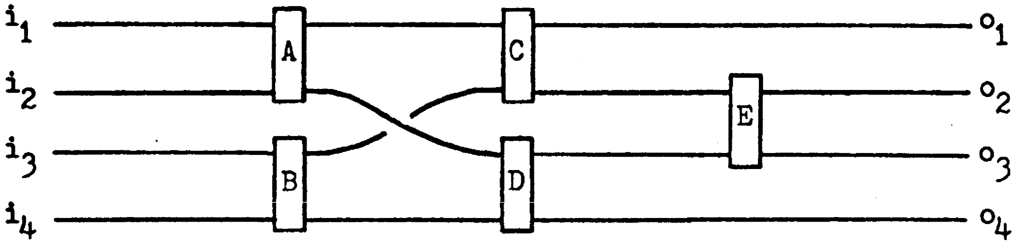

Fig. 1. **4-sorter** network.

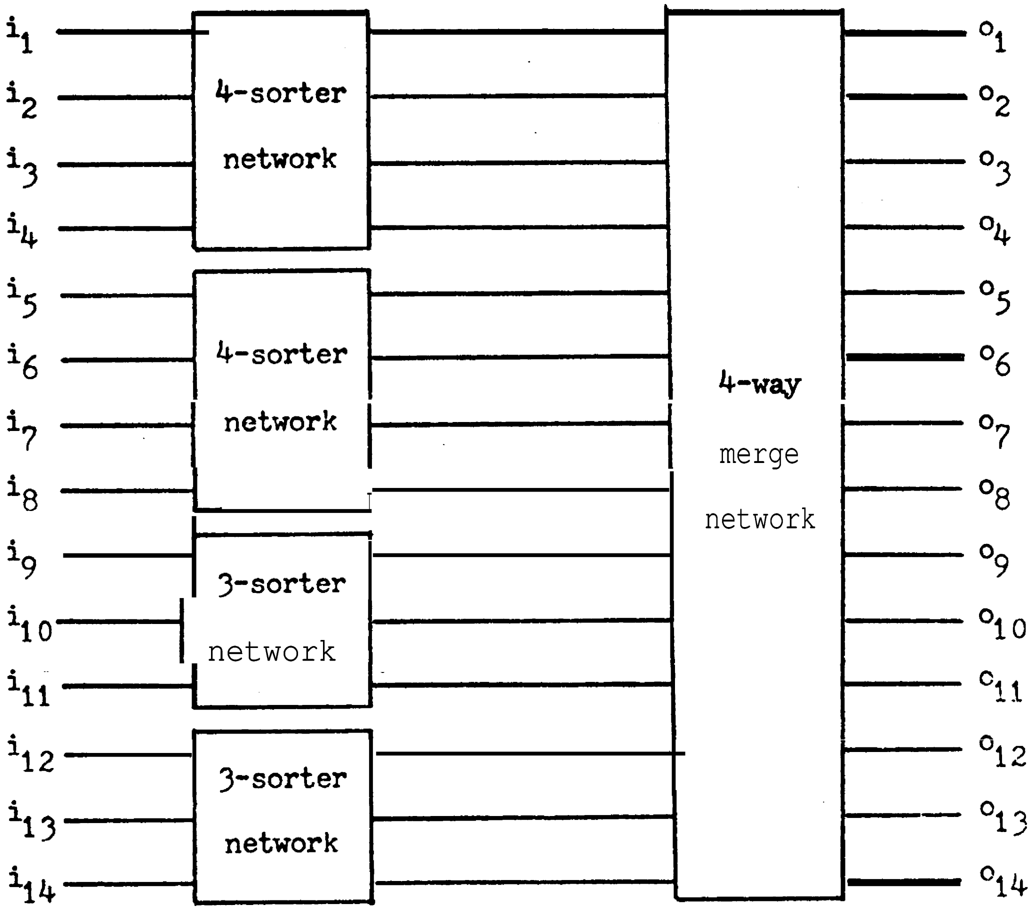

Fig. 2. **14-sorter** that uses the **4-way** divide-sort-merge strategy.

For N > 34 the most economical N-sorter networks yet constructed use the g-way divide-sort-merge strategy. That is, they consist of:

i) g sorting networks of size $N_1, N_2, \ldots, N_g$ where $N_i = \lfloor (N+g-i)/g \rfloor$, that also use the g-way divide-sort-merge strategy; followed by

ii) a network that combines the outputs of the N_1-, N_2-, . . ., N_g-sorter networks into a single sorted sequence. This network is called a g-way merge network.

'The g-way divide-sort-merge strategy is illustrated in Fig. 2 for the case $N = 14$, $g = 4$. In this paper we show that an N-sorter network which uses the g-way strategy, $g \geq 2$, must contain at least order $N(\log_2 N)^2$ comparators.

Let $S_g(N)$ represent the minimum number of comparators required by an N-sorter network that uses the g-way strategy. Then $S_g(N)$ satisfies the recurrence relation

$$S_g(N) \quad , = \sum_{1 \leq i \leq g} S_g(N_i) + M_g(N), \tag{1}$$

where $N_i = \lfloor (N+g-i)/g \rfloor$ and $M_g(N)$ is the minimum number of comparators required by a network that merges g sorted multisets of size $N_1, N_2, \ldots, N_g$. In order to determine the asymptotic growth of $S_g(N)$ we may restrict out attention to the values $N = g^k$. From (1) we obtain

$$S_g(g^{k+1}) = g\ S_g(g^k) + M_g(g^{k+1}). \tag{2}$$

Theorem 1 below provides a lower bound for $M_g(g^k)$, which in turn allows us to bound $S_g(g^k)$. It is convenient to use one lemma.

Lemma 1: $$M_g(rg) \geq r\ M_g(g) + \sum_{2 \leq \ell \leq r} \lfloor (\ell-1)g/\ell \rfloor. \tag{3}$$

Proof:

Consider the network T that contains $M_g(rg)$ comparators and that will merge g sorted multisets containing r members each. Let the inputs to T, namely $X = \{x_1, x_2, \ldots, x_{rg}\}$, be numbered so that the g sorted multisets of inputs are

$$C_j = \bigcup_{1 \leq i \leq r} \{x_{(i-1)g+j}\}, \quad 1 \leq j \leq g. \tag{4}$$

Note that if we consider X to be an $r \times g$ array, with $X_{(i,j)} = x_{(i-1)g+j}$, then the g columns of X are ordered. Fig. 3 illustrates X for the case $r = 3, g = 5$.

The comparators in T may be divided into two distinct classes as follows. A comparator is said to be in class A if it compares two elements in the same row of X and in class B if it compares elements in different rows. We shall prove that the two terms in the right-hand-side of (3) are lower bounds, respectively, for the number of class A and class B comparators in T.

Since T is a g-way merge network, it must complete the ordering of any $r \times g$ array X that has sorted columns. In particular, it must order X when

$$X_{(i,j)} = \begin{cases} 0, & i < \ell; \\ 1,2,\ldots, \text{ or } g, & i = \ell; \\ g+1, & i > \ell, \end{cases} \tag{5}$$

x_1	x_2	x_3	x_4	x_5
x_6	x_7	x_8	x_9	x_{10}
x_{11}	x_{12}	x_{13}	x_{14}	x_{15}

Fig. 3. Inputs to T,

0	0	0	0	0
2	5	1	3	4
6	6	6	6	6

(a)

0	0	1	1	1
0	0	1	1	1
1	1	1	1	1

(b)

Fig. 4. Possible values of inputs to T.

where $\ell \in [1,r]$. That is, it must complete the ordering of X when the first $\ell-1$ rows of X each contain r 0's, the last $r-\ell$ rows each contain r $(g+1)$'s, and the ℓ^{th} row contains values in $[1,g]$. (This situation is illustrated in Fig. 4(a) for the case $r=3, g=5$, $k=2$.) Since (5) may be satisfied when the ℓ^{th} row of X contains any permutation of the numbers $1,2,\ldots,g$, T must contain at least $M_g(g)$ comparators that sort the ℓ^{th} row. And since no class B comparator that compares an element in the ℓ^{th} row to an element in another row will cause an interchange, these $M_g(g)$ comparators must all be class A. Letting ℓ vary from 1 to r we verify that T must contain at least $r\,M_g(g)$ class A comparators, $M_g(g)$ for each row.

Now suppose that the inputs to T are given by

$$X_{(i,j)} = \begin{cases} 0, & i \le \ell,\ j \le \lfloor(\ell-1)g/\ell\rfloor; \\ 1, & \text{otherwise}, \end{cases} \qquad (6)$$

where $\ell \in [2,r]$. That is, suppose that the first ℓ rows of X each contain $\lfloor(\ell-1)g/\ell\rfloor$ 0's and that the remaining elements of X are 1. Since X contains only $\ell\lfloor(\ell-1)g/\ell\rfloor \le (\ell-1)g$ 0's, all of the 0's in X belong in the first $\ell-1$ rows. And since no comparator will move a 0 from the ℓ^{th} row to a higher indexed row, T must contain at least $\lfloor(\ell-1)g/\ell\rfloor$ class B comparators that connect an element in the ℓ^{th} row to an element in a lower indexed row. Letting ℓ very from 2 to r we conclude that the second term in the right-hand-side of (3) provides a lower bound for the number of class B comparators in T.

Q.E.D.

The second term in theright-hand-side of (3) is a function of the two variables r and g, namely

$$\sigma(r,g) = \sum_{2\le\ell\le r} \lfloor(\ell-1)g/\ell\rfloor. \tag{7}$$

With this definition we are now ready to prove Theorem 1.

Theorem 1:* $$M_g(rg^2) \ge g\ M_g(rg) + r\ \sigma(g,g). \tag{8}$$

Proof:

Consider the merge network $\hat{T}$ that contains $M_g(rg^2)$ comparators and that will merge g sorted multisets containing rg members each. Let the rg^2 inputs $X = \{x_1,x_2,\ldots,x_{rg^2}\}$ to $\hat{T}$ be numbered so that the g sorted multisets of inputs are

$$C_j = \bigcup_{1\le i\le rg} \{x_{(i-1)g+j}\}, \qquad 1\le j\le g. \tag{9}$$

If we consider X to be an rg x g array, with $X_{(i,j)} = x_{(i-1)g+j}$, then the g columns $X_{(*,j)} = C_j$ are each ordered.

It is convenient to partition the rg rows of X, given by

$$X_{(i,*)} = \bigcup_{1\le j\le g} \{X_{(i,j)}\}, \qquad 1\le i\le rg, \tag{10}$$

* Theorem 1 is a generalization of the following theorem proved by R. W. Floyd [3]: $M_2(4n) \ge 2M_2(2n) + n$.

into g partitions containing r rows each. We define these partitions according to

$$P_{\mu} = \bigcup_{(\mu-1)r<i\leq\mu r} X_{(i,*)}, \qquad 1\leq\mu\leq g, \tag{11}$$

so that P1 consists of the first r rows, . . . , and P_g contains the last r rows of X. These partitions are illustrated in Fig. 5 forthecase $r = 3$, $g = 5$.

The comparators in $\hat{T}$ may be divided into two classes, according to whether the two elements compared are in the same partition or in different partitions. Now each partition, which contains r rows of X, may be considered to be an $r \times g$ array with ordered columns. Therefore, $\hat{T}$ must contain at least $M_g(rg)$ comparators within each of the g partitions, which explains the first term in the **right-hand-side** of (8). The second term in the right-hand-side of (8) is a bound for the number of comparators that join elements in different partitions; the derivation of the term follows the proof of Lemma 1.

Q.E.D.

We may use Theorem 1, with $r = g^{k-1}$, to obtain the recurrence relation

$$M_g(g^{k+1}) \geq g\ M_g(g^{k}) + a_g\ g^{k+1}, \tag{12}$$

where

	x_1	x_2	x_3	x_4	x_5
P_1	x_6	x_7	x_8	x_9	x_{10}
	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
	x_{16}	*17	x_{18}	x_{19}	x_{20}
P_2	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}
	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}
	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}
P_3	x_{36}	x_{37}	x_{38}	x_{39}	x_{40}
	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}
	x_{46}	x_{47}	x_{48}	x_{49}	x_{50}
P_4	x_{51}	x_{52}	x_{53}	x_{54}	x_{55}
	x_{56}	x_{57}	x_{58}	x_{59}	x_{60}
	*61	x_{62}	x_{63}	x_{64}	x_{65}
P_5	x_{66}	x_{67}	x_{68}	x_{69}	x_{70}
	x_{71}	x_{72}	x_{73}	x_{74}	x_{75}

Fig. 5. Inputs to $\hat{T}$.

$$a_g = \sigma(g,g)/g^2. \tag{13}$$

With the boundary condition

$$M_g(g) = S_g(g) = \eta, \tag{14}$$

Equations (12) and (2) lead to

$$M_g(g^{k+1}) \geq [a_g k + (\eta/g)]\, g^{k+1}, \tag{15}$$

$$S_g(g^k) \geq [\tfrac{1}{2}a_g k^2 + ((\eta/g) - \tfrac{1}{2}a_g)]\, g^k. \tag{16}$$

From (16) we observe that $S_g(N)$ is bounded by $L(N)$, where

$$\begin{aligned} L(N) &\sim \tfrac{1}{2}a_g N(\log_g N)^2 \\ &= \tfrac{1}{2}a_g (\log_2 g)^{-2}\, N(\log_2 N)^2. \end{aligned} \tag{17}$$

From (7) and (13) we can easily verify that $a_g > 0$, $g \geq 2$. Therefore, the minimum number of comparators required by an N-sorter network that uses the g-way divide-sort-merge strategy grows asymptotically as $N(\log_2 N)^2$.

REFERENCES

[1] D.E. Knuth [1969]: Seminumerical algorithms. *The Art of Computer Programming*, **2**, Addison-Wesley Publishing Company,

[2] R.C. Bose and R.J. Nelson [1962]: A sorting problem. *J. Assoc. Comp. Mach*. **9**,282-296.

[3] R.W. Floyd and D.E. Knuth [1970]: The Bose-Nelson sorting problem. CS Report 70-177, Stanford University, Stanford, California; November 1970.

[4] D.C. Van Voorhis [1971]: An improved lower bound for the Bose-Nelson sorting problem. Technical Note no. 7, Digital Systems Laboratory, Stanford University, Stanford, . California; February 1971.

[5] D.C. Van Voorhis [1971]: A generalization of the divide-sort-merge strategy for sorting networks. Technical Report no. 16, Digital Systems Laboratory, Stanford University, Stanford, California; August 1971.

ACKNOWLEDGMENTS

The author would like to express his appreciation to Dr. Harold S. Stone for his prompt and careful attention to this paper, and for his many helpful comments and suggestions.