THE CLASSIC WORK
NEWLY UPDATED AND REVISED

The Art of Computer Programming

VOLUME 3

Sorting and Searching

Second Edition

DONALD E. KNUTH

This page intentionally left blank

This is not an especially simple function of m and n, in general, but by noting that $C(1,n) = n$ and that

$$C(m+1,n+1) - C(m,n) \\ = 1 + C\big(\lfloor m/2\rfloor + 1, \lfloor n/2\rfloor + 1\big) - C\big(\lfloor m/2\rfloor, \lfloor n/2\rfloor\big), \quad \text{if } mn \ge 1,$$

we can derive the relation

$$C(m+1,n+1) - C(m,n) = \lfloor \lg m\rfloor + 2 + \lfloor n/2^{\lfloor \lg m\rfloor + 1}\rfloor, \quad \text{if } n \ge m \ge 1. \quad (5)$$

Consequently

$$C(m, m+r) = B(m) + m + R_m(r), \qquad \text{for } m \ge 0 \text{ and } r \ge 0, \qquad (6)$$

where $B(m)$ is the "binary insertion" function $\sum_{k=1}^{m} \lceil \lg k\rceil$ of Eq. 5.3.1–(3), and where $R_m(r)$ denotes the sum of the first m terms of the series

$$\left\lfloor\frac{r+0}{1}\right\rfloor + \left\lfloor\frac{r+1}{2}\right\rfloor + \left\lfloor\frac{r+2}{4}\right\rfloor + \left\lfloor\frac{r+3}{4}\right\rfloor + \left\lfloor\frac{r+4}{8}\right\rfloor + \cdots + \left\lfloor\frac{r+j}{2^{\lfloor \lg j\rfloor + 1}}\right\rfloor + \cdots. \quad (7)$$

In particular, when $r = 0$ we have the important special case

$$C(m,m) = B(m) + m. \qquad (8)$$

Furthermore if $t = \lceil \lg m\rceil$,

$$R_m(r + 2^t) = R_m(r) + 1 \cdot 2^{t-1} + 2 \cdot 2^{t-2} + \cdots + 2^{t-1} \cdot 2^0 + m \\ = R_m(r) + m + t \cdot 2^{t-1}.$$

Hence $C(m, n+2^t) - C(m,n)$ has a simple form, and

$$C(m,n) = \left(\frac{t}{2} + \frac{m}{2^t}\right) n + O(1), \qquad \text{for } m \text{ fixed, } n \to \infty,\ t = \lceil \lg m\rceil; \quad (9)$$

the $O(1)$ term is an eventually periodic function of n, with period length 2^t. As $n \to \infty$ we have $C(n,n) = n \lg n + O(n)$, by Eq. (8) and exercise 5.3.1–15.

Minimum-comparison networks. Let $\hat{S}(n)$ be the minimum number of comparators needed in a sorting network for n elements; clearly $\hat{S}(n) \ge S(n)$, where $S(n)$ is the minimum number of comparisons needed in a not-necessarily-oblivious sorting procedure (see Section 5.3.1). We have $\hat{S}(4) = 5 = S(4)$, so the new constraint causes no loss of efficiency when $n = 4$; but already when $n = 5$ it turns out that $\hat{S}(5) = 9$ while $S(5) = 7$. The problem of determining $\hat{S}(n)$ seems to be even harder than the problem of determining $S(n)$; even the asymptotic behavior of $\hat{S}(n)$ is known only in a very weak sense.

It is interesting to trace the history of this problem, since each step was forged with some difficulty. Sorting networks were first explored by P. N. Armstrong, R. J. Nelson, and D. G. O'Connor, about 1954 [see *U.S. Patent 3029413*]; in the words of their patent attorney, "By the use of skill, it is possible to design economical n-line sorting switches using a reduced number of two-line sorting switches." After observing that $\hat{S}(n+1) \le \hat{S}(n) + n$, they gave special constructions for $4 \le n \le 8$, using 5, 9, 12, 18, and 19 comparators, respectively. Then Nelson worked together with R. C. Bose to show that $\hat{S}(2^n) \le 3^n - 2^n$ for all n; hence $\hat{S}(n) = O(n^{\lg 3}) = O(n^{1.585})$. Bose and Nelson published their interesting method in *JACM* **9** (1962), 282–296, where they conjectured that it was best possible; T. N. Hibbard [*JACM* **10** (1963), 142–150] found a similar but slightly simpler construction that used the same number of comparisons, thereby reinforcing the conjecture.

In 1964, R. W. Floyd and D. E. Knuth found a new way to approach the problem, leading to an asymptotic bound of the form $\hat{S}(n) = O\bigl(n^{1+c/\sqrt{\log n}}\bigr)$. Working independently, K. E. Batcher discovered the general merging strategy outlined above. Using a number of comparators defined by the recursion

$$c(1) = 0, \quad c(n) = c\bigl(\lceil n/2\rceil\bigr) + c\bigl(\lfloor n/2\rfloor\bigr) + C\bigl(\lceil n/2\rceil, \lfloor n/2\rfloor\bigr) \quad \text{for } n \ge 2, \qquad (10)$$

he proved (see exercise 5.2.2–14) that

$$c(2^t) = (t^2 - t + 4)2^{t-2} - 1;$$

consequently $\hat{S}(n) = O\bigl(n(\log n)^2\bigr)$. Neither Floyd and Knuth nor Batcher published their constructions until some time later [*Notices of the Amer. Math. Soc.* **14** (1967), 283; *Proc. AFIPS Spring Joint Computer Conf.* **32** (1968), 307–314].

Several people have found ways to reduce the number of comparators used by Batcher's merge-exchange construction; the following table shows the best upper bounds currently known for $\hat{S}(n)$:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$c(n)$	0	1	3	5	9	12	16	19	26	31	37	41	48	53	59	63
$\hat{S}(n) \le$	0	1	3	5	9	12	16	19	25	29	35	39	45	51	56	60

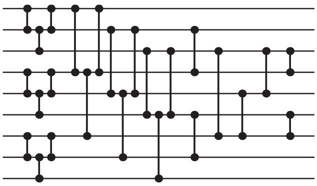
(11)

Since $\hat{S}(n) < c(n)$ for $8 < n \le 16$, merge exchange is nonoptimal for all $n > 8$. When $n \le 8$, merge exchange uses the same number of comparators as the construction of Bose and Nelson. Floyd and Knuth proved in 1964–1966 that the values listed for $\hat{S}(n)$ are *exact* when $n \le 8$ [see *A Survey of Combinatorial Theory* (North-Holland, 1973), 163–172]; M. Codish, L. Cruz-Filipe, M. Frank, and P. Schneider-Kamp [*Journal of Computer and System Sciences* **82** (2016), 551–563] have also verified this when $n \le 10$. J. Harder settled the cases $n = 11$ and $n = 12$ in arXiv:2012.04400 [cs.DS] (2020), 54 pages. The remaining values of $\hat{S}(n)$ are still not known.

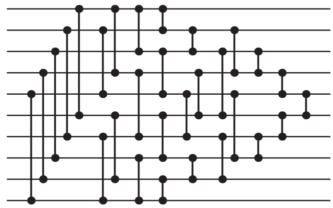
Constructions that lead to the values in (11) are shown in Fig. 49. The network for $n = 9$, based on an interesting three-way merge, was found by R. W. Floyd in 1964; its validity can be established by using the general principle described in exercise 27. The network for $n = 10$ was discovered by A. Waksman in 1969, by regarding the inputs as permutations of $\{1, 2, \ldots, 10\}$ and trying to reduce as much as possible the number of values that can appear on each line at a given stage, while maintaining some symmetry.

The network shown for $n = 13$ has quite a different pedigree: Hugues Juillé [*Lecture Notes in Comp. Sci.* **929** (1995), 246–260] used a computer program to construct it, by simulating an evolutionary process of genetic breeding. The network exhibits no obvious rhyme or reason, but it works — and it's shorter than any other construction devised so far by human ratiocination.

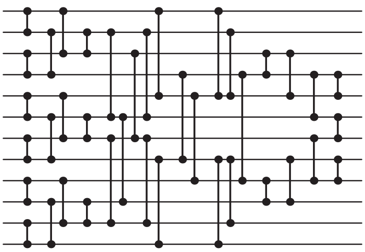
A 62-comparator sorting network for 16 elements was found by G. Shapiro in 1969, and this was rather surprising since Batcher's method (63 comparisons)



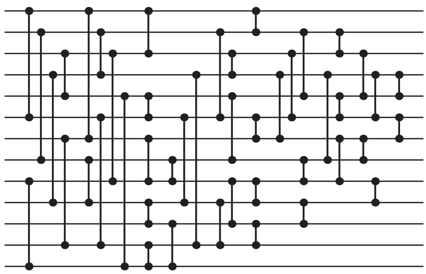
$n=9$ 25 modules, delay 9



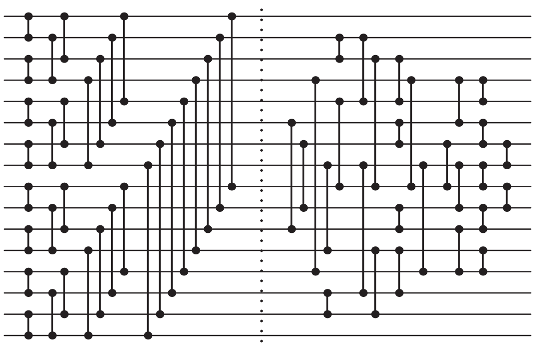
$n=10$ 29 modules, delay 9



$n=12$ 39 modules, delay 9



$n=13$ 45 modules, delay 10



$n=16$ 60 modules, delay 10

Fig. 49. Efficient sorting networks.

would appear to be at its best when n is a power of 2. Soon after hearing of Shapiro's construction, M. W. Green tripled the amount of surprise by finding the 60-comparison sorter in Fig. 49. The first portion of Green's construction is fairly easy to understand; after the 32 comparison/interchanges to the left of the dotted line have been made, the lines can be labeled with the 16 subsets of $\{a, b, c, d\}$, in such a way that the line labeled s is known to contain a number less than or equal to the contents of the line labeled t whenever s is a subset of t. The state of the sort at this point is discussed further in exercise 32. Comparisons made on subsequent levels of Green's network become increasingly mysterious, however, and as yet nobody has seen how to generalize the construction in order to obtain correspondingly efficient networks for higher values of n.

Shapiro and Green also discovered the network shown for $n = 12$. When $n = 11$, 14, or 15, good networks can be found by removing the bottom line of the network for $n+1$, together with all comparators touching that line.

The best sorting network currently known for 256 elements, due to D. Van Voorhis, shows that $\hat{S}(256) \le 3651$, compared to 3839 by Batcher's method. [See R. L. Drysdale and F. H. Young, *SICOMP* **4** (1975), 264–270.] As $n \to \infty$, it turns out in fact that $\hat{S}(n) = O(n \log n)$; this astonishing upper bound was proved by Ajtai, Komlós, and Szemerédi in *Combinatorica* **3** (1983), 1–19. The networks they constructed are not of practical interest, since many comparators were introduced just to save a factor of $\log n$; Batcher's method is much better, unless n exceeds the total memory capacity of all computers on earth! But the theorem of Ajtai, Komlós, and Szemerédi does establish the true asymptotic growth rate of $\hat{S}(n)$, up to a constant factor.

Minimum-time networks. In physical realizations of sorting networks, and on parallel computers, it is possible to do nonoverlapping comparison-exchanges at the same time; therefore it is natural to try to minimize the delay time. A moment's reflection shows that the delay time of a sorting network is equal to the maximum number of comparators in contact with any "path" through the network, if we define a path to consist of any left-to-right route that possibly switches lines at the comparators. We can put a sequence number on each comparator indicating the earliest time it can be executed; this is one higher than the maximum of the sequence numbers of the comparators that occur earlier on its input lines. (See Fig. 50(a); part (b) of the figure shows the same network redrawn so that each comparison is done at the earliest possible moment.)

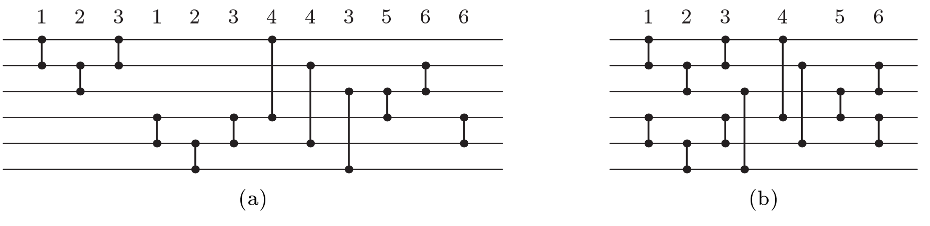


Fig. 50. Doing each comparison at the earliest possible time.

Batcher's odd-even merging network described above takes $T_B(m,n)$ units of time, where $T_B(m,0) = T_B(0,n) = 0$, $T_B(1,1) = 1$, and

$$T_B(m,n) = 1 + \max\bigl(T_B(\lfloor m/2\rfloor, \lfloor n/2\rfloor), T_B(\lceil m/2\rceil, \lceil n/2\rceil)\bigr) \quad \text{for } mn \ge 2.$$

We can use these relations to prove that $T_B(m,n+1) \ge T_B(m,n)$, by induction; hence $T_B(m,n) = 1 + T_B\bigl(\lceil m/2\rceil, \lceil n/2\rceil\bigr)$ for $mn \ge 2$, and it follows that

$$T_B(m,n) = 1 + \lceil \lg \max(m,n) \rceil, \qquad \text{for } mn \ge 1. \tag{12}$$

Exercise 5 shows that Batcher's sorting method therefore has a delay time of

$$\binom{1 + \lceil \lg n \rceil}{2}. \tag{13}$$

Let $\hat{T}(n)$ be the minimum achievable delay time in any sorting network for n elements. It is possible to improve some of the networks described above so

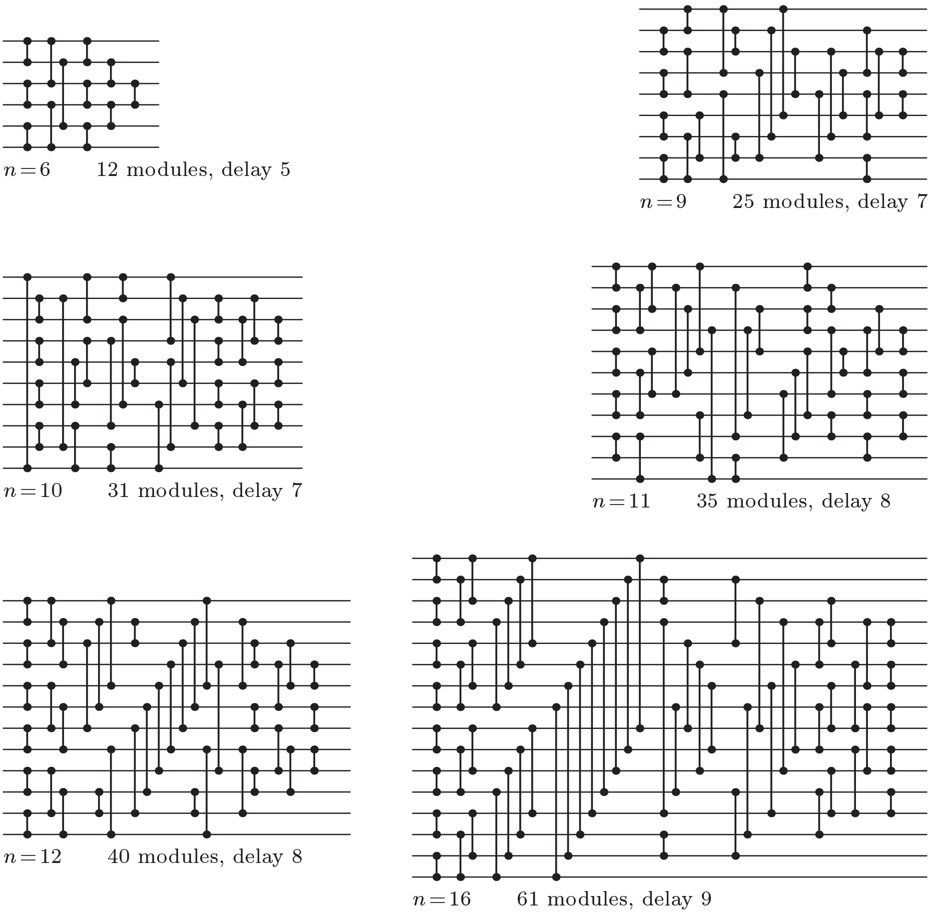


Fig. 51. Sorting networks that are the fastest known, when comparisons are performed in parallel.

that they have smaller delay time but use no more comparators, as shown for $n = 6$, $n = 9$, and $n = 11$ in Fig. 51, and for $n = 10$ in exercise 7. Still smaller delay time can be achieved if we add one or two extra comparator modules, as shown in the remarkable networks for $n = 10$, 12, and 16 in Fig. 51. These constructions yield the following upper bounds on $\hat{T}(n)$ for small n:

$$\begin{array}{rcccccccccccccccc} n & =1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ \hat{T}(n) & \leq 0 & 1 & 3 & 3 & 5 & 5 & 6 & 6 & 7 & 7 & 8 & 8 & 9 & 9 & 9 & 9 \end{array} \qquad (14)$$

In fact all of the values given here are known to be exact (see the answer to exercise 4). The networks in Fig. 51 merit careful study, because it is by no means obvious that they always sort. Some of these networks were discovered in 1969–1971 by G. Shapiro ($n = 6$, 12) and D. Van Voorhis ($n = 10$, 16); the others were found in 2001 by Loren Schwiebert, using genetic methods ($n = 9$, 11).

Merging networks. Let $\hat{M}(m,n)$ denote the minimum number of comparator modules needed in a network that merges m elements $x_1 \le \cdots \le x_m$ with n elements $y_1 \le \cdots \le y_n$ to form the sorted sequence $z_1 \le \cdots \le z_{m+n}$. At present no merging networks have been discovered that are superior to the odd-even merge described above; hence the function $C(m,n)$ in (6) represents the best upper bound known for $\hat{M}(m,n)$.

R. W. Floyd has discovered an interesting way to find *lower* bounds for this merging problem.

Theorem F. *For all $n \ge 1$, we have $\hat{M}(2n,2n) \ge 2\hat{M}(n,n) + n$.*

Proof. Consider a network with $\hat{M}(2n,2n)$ comparator modules, capable of sorting all input sequences $\langle z_1,\ldots,z_{4n}\rangle$ such that $z_1 \le z_3 \le \cdots \le z_{4n-1}$ and $z_2 \le z_4 \le \cdots \le z_{4n}$. We may assume that each module replaces (z_i,z_j) by $\bigl(\min(z_i,z_j),\max(z_i,z_j)\bigr)$, for some $i<j$ (see exercise 16). The comparators can therefore be divided into three classes:

a) $i \le 2n$ and $j \le 2n$.

b) $i > 2n$ and $j > 2n$.

c) $i \le 2n$ and $j > 2n$.

Class (a) must contain at least $\hat{M}(n,n)$ comparators, since $z_{2n+1}, z_{2n+2}, \ldots, z_{4n}$ may be already in their final position when the merge starts; similarly, there are at least $\hat{M}(n,n)$ comparators in class (b). Furthermore the input sequence $\langle 0,1,0,1,\ldots,0,1\rangle$ shows that class (c) contains at least n comparators, since n zeros must move from $\{z_{2n+1},\ldots,z_{4n}\}$ to $\{z_1,\ldots,z_{2n}\}$. ▮

Repeated use of Theorem F proves that $\hat{M}(2^m,2^m) \ge \frac{1}{2}(m+2)2^m$; hence $\hat{M}(n,n) \ge \frac{1}{2}n\lg n + O(n)$. We know from Theorem 5.3.2M that merging *without* the network restriction requires only $M(n,n) = 2n-1$ comparisons; hence we have proved that merging with networks is intrinsically harder than merging in general.

The odd-even merge shows that

$$\hat{M}(m,n) \le C(m,n) = \tfrac{1}{2}(m+n)\lg\min(m,n) + O(m+n).$$

P. B. Miltersen, M. Paterson, and J. Tarui [*JACM* **43** (1996), 147–165] have improved Theorem F by establishing the lower bound

$$\hat{M}(m,n) \ge \tfrac{1}{2}\bigl((m+n)\lg(m+1) - m/\ln 2\bigr) \qquad \text{for } 1 \le m \le n.$$

Consequently $\hat{M}(m,n) = \frac{1}{2}(m+n)\lg\min(m,n) + O(m+n)$.

The exact formula $\hat{M}(2,n) = C(2,n) = \lceil \frac{3}{2}n \rceil$ has been proved by A. C. Yao and F. F. Yao [*JACM* **23** (1976), 566–571]. The value of $\hat{M}(m,n)$ is also known to equal $C(m,n)$ for $m = n \le 5$; see exercise 9.

Bitonic sorting. When simultaneous comparisons are allowed, we have seen in Eq. (12) that the odd-even merge uses $\lceil \lg(2n) \rceil$ units of delay time, when $1 \le m \le n$. Batcher has devised another type of network for merging, called a