



---

Soviet-era science, translated into English

# AN OPTIMAL ALGORITHM FOR ORDERING WITHOUT A WORKING AREA

1969

SovietRxiv

---

View the original and related papers at <https://sovietrxiv.org/items/ru-196901.22975>

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.

**Abstract**

**Full Text**

UDC 518.5

**M. A. KRONROD**

## **AN OPTIMAL ALGORITHM FOR ORDERING WITHOUT A WORKING AREA**

*(Presented by Academician I. G. Petrovskii, 25 XII 1968)*

It is easy to show that an array of  $n$  numbers cannot be ordered in fewer than  $Cn \lg n$  operations. Shell constructed an ordering algorithm requiring  $Cn \lg n$  operations, but this algorithm uses a working area of length of order  $n$ . (More precisely: if a free area of length  $k$  is available,  $C(n^2/k + n \lg k)$  operations are required.) M. Ya. Vainshtein constructed an ordering algorithm requiring  $Cn \lg^2 n$  operations without a working area.

In the present paper it is shown that ordering without a working area can be carried out in  $Cn \lg n$  operations\*.

**Notation.** By the words *array of cells* we mean an array of consecutively arranged cells. Arrays of cells are denoted by  $M, L, W$ , etc.;  $M_i$  denotes the  $i$ -th cell of the array  $M$ , and  $SM_i$  the number contained in it. The segment of the array  $M$  from  $M_i$  through  $M_j$ , inclusive, is denoted by  $M[i : j]$ .

If in an array  $M$  the inequality  $SM_1 \geq SM_2 \geq \dots \geq SM_n$  holds, then we say that the contents of  $M$  are ordered.

If  $SM_1 \geq SM_2 \geq \dots \geq SM_k$  and  $SM_{k+1} \geq SM_{k+2} \geq \dots \geq SM_n$ , then we say that the contents of  $M$  are ordered for an array with boundary  $k$ .

Throughout what follows only logarithms to base 2 are used, and therefore, for brevity, we shall write  $\lg n$  instead of  $\log_2 n$ .

In each item (in each lemma or theorem) what is meant is the existence of a program using a bounded number of working cells and carrying out the assertion of the item. The factor  $C$  occurring in the formulation of each item is its own for each item.

**Lemma 1.** *Let there be two arrays  $M$  and  $L$ , each of  $n$  cells,  $SM_i = a_i$ ,  $SL_i = b_i$  ( $i = 1, 2, \dots, n$ ). Then the contents of these arrays can be interchanged in  $Cn$  operations, i.e., made so that  $SM_i = b_i$ ,  $SL_i = a_i$ .*

**Lemma 2.** *The contents of an array  $M$  of  $n$  cells can be ordered in  $Cn^2$  operations.*

**Lemma 3.** *Let there be an array  $M$  of  $n$  cells and an array  $W$ , called auxiliary, of  $k$  or more cells, and let the contents of the array  $M$  be ordered into two arrays with boundary  $k$  or  $(n - k)$ . Then the contents of the array  $M$  can be ordered in  $Cn$  operations, while, generally speaking, changing the order of arrangement of the numbers inside the array  $W$ .*

**Proof.** Let, for example, the boundary be equal to  $k$ . First, by Lemma 1, interchange the contents of the arrays  $M[1 : k]$  and  $W[1 : k]$ . The candidates for first place in the array  $M$  are the numbers lying in the cells  $M_{k+1}$  and  $W_1$ ; the larger of them we interchange with the contents of  $M_1$ . The candidates for second place are: the rejected candi-

\* Formally, one working cell is required in order to be able to interchange two numbers; in a computer that has, for example, a “swap” instruction, this interchange and, consequently, the entire ordering are carried out entirely without working cells.

put it in first place, and the number that was after the larger of the candidates for first place. The larger of these numbers is exchanged with the contents of  $M_2$ , and so on. When the contents of some  $M_i$  are exchanged with the contents of  $W_k$ , the ordering will be finished.

The case where the boundary is equal to  $(n - k)$  is treated analogously, except that one must start not with the candidates for first place but with the candidates for last place.

**Lemma 4.** *The contents of an array  $M$  of  $n$  cells can be ordered in  $C(n + k^2)$  operations if the contents of the array  $M[k + 1 : n]$  have been ordered.*

**Proof.** By Lemma 2, order the contents of the array  $M[1 : 2k]$ . Note that the largest  $k$  numbers now lie in the array  $M[1 : k]$ . Using this array as an auxiliary one, by Lemma 3 order the contents of the array  $M[k + 1 : n]$ . It remains only to order, by Lemma 2, the contents of the array  $M[1 : k]$ .

**Theorem 1.** *Let an array  $M$  of  $n$  cells be given, and suppose that the contents of the array  $M$  have been ordered into two arrays with boundary  $l$ . Then the contents of  $M$  can be ordered in  $Cn$  operations.*

**Proof.** Put  $k = \lceil \sqrt{n} \rceil$ ;  $p \equiv l \pmod{k}$ ;  $q \equiv (n - l) \pmod{k}$ . We shall call zones the arrays of  $k$  cells each:

$$Z^{(1)} = M[p + k + 1 : p + 2k]$$

—the first zone,

$$Z^{(2)} = M[p + 2k + 1 : p + 3k]$$

—the second zone, and so on. Thus we have divided the array  $M$  into  $t$  zones, the array  $M[n - q + 1 : n]$ , and the array  $M[1 : p + k]$ , which we shall use as an auxiliary one. The contents of each zone are ordered.

We shall call the first number of each zone the *mark* of the zone. Find the zone with the maximum mark and interchange its contents with the contents of the

first zone; then find the zone with the second-largest mark and interchange its contents with the contents of the second zone, and so on.

**Remark.** Now for each zone  $Z^{(i)}$  there are no more than  $k$  numbers smaller than the mark of  $Z^{(i)}$ , and lying in zones with numbers smaller than  $i$ .

By Lemma 3, order the contents of the array  $M[p + k + 1 : p + 3k]$ , using the array  $M[1 : p + k]$  as an auxiliary array; then, by the same lemma, order the contents of  $M[p + 2k + 1 : p + 4k]$ . From the remark it follows that now the contents of  $M[p + k + 1 : p + 4k]$  will be ordered. Then order the contents of  $M[p + 3k + 1 : p + 5k]$ , and so on.

Thus we order the contents of the array  $M[p + k + 1 : n - q]$ . Since the contents of the array  $M[n - q + 1 : n]$  are ordered, we can now, by Lemma 3, order the contents of the array  $M[p + k + 1 : n]$ , and then, by Lemma 4, the whole array  $M$ .

**Theorem 2 (main).** *The contents of an array  $M$  of  $n$  cells can be ordered in  $Cn$  operations.*

**Proof.** In the proof we shall use Theorem 1. First order the contents of the arrays  $M[1 : 2]$ ,  $M[3 : 4]$ , and so on. (All these arrays, except possibly the last, have length equal to 2, and the last has length less than or equal to 2.) Then order the contents of the arrays  $M[1 : 4]$ ,  $M[5 : 8]$ , and so on; then the contents of arrays of 8 cells, of 16 cells, and so on.

**Theorem 3 (generalization of Theorem 2).** *Let an array  $M$  of  $n$  cells be given, and suppose that the contents of each of the arrays  $M[1 : i_1]$ ,  $M[i_1 + 1 : i_2]$ , ...,  $M[i_{k-1} + 1 : n]$  have been ordered. Then the contents of the array  $M$  can be ordered in  $Cn \lg k$  operations.*

The proof of Theorem 3 is analogous to the proof of Theorem 2. By Theorem 1, first the contents of the arrays  $M[1 : i_2]$ ,  $M[i_2 + 1 : i_4]$ ,  $M[i_4 + 1 : i_6]$ , etc., are ordered; then the contents of the arrays  $M[1 : i_4]$ ,  $M[i_4 + 1 : i_8]$ ,  $M[i_8 + 1 : i_{12}]$ , etc.

For the algorithms whose existence is proved by Theorems 1-3, large constants  $C$  are obtained when the algorithm is implemented on a computer. Su-

there exist other algorithms with smaller constants. The basis for all of them is Lemma 3. With its help one can prove the following lemma (generally speaking, this lemma is a special case of Theorem 2, but the algorithm given in its proof has a smaller constant  $C$ ).

**Lemma 5.** *Let there be an array  $M$  of  $n$  cells and an array  $W$ , called auxiliary, of  $n$  or more cells. Then, in  $Cn \lg n$  operations, it is possible to order the contents of the array  $M$ , while changing, generally speaking, the order of the numbers in the array  $W$ .*

The **proof** of Theorem 2 now looks, for example, as follows. Put

$$k_1 = \left\lceil \frac{n+1}{2} \right\rceil.$$

By Lemma 5, order the contents of the array  $M[k_1 + 1 : n]$ , using  $M[1 : k_1]$  as auxiliary. Put

$$k_2 = \left\lceil \frac{k_1 + 1}{2} \right\rceil.$$

Using  $M[1 : k_2]$  as auxiliary, first order the contents of  $M[k_2 + 1 : k_1]$  by Lemma 5, and then the contents of  $M[k_2 + 1 : n]$  by Lemma 3. Put

$$k_3 = \left\lceil \frac{k_2 + 1}{2} \right\rceil$$

and so on. Thus we order the contents of the entire array  $M$  in  $Cn \lg n$  operations. The constant  $C$  here is considerably smaller than in the first proof. But it can also be reduced by a factor of 2 if one carries out only  $p = \lg(\lg n)$  steps and then, for example, applies Theorem 2 (with the first proof) to the array  $M[1 : k_p]$  and then orders the contents of the entire array  $M$  by Theorem 1.

Moscow State University  
named after M. V. Lomonosov

Received  
14 XII 1968

*Note: Figure translations are in progress. See original paper for figures.*

*Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.*