UDC 004.451:811.111

Tulchak L.V.
Senior Teacher
of Foreign Languages Department, VNTU
Noskovenko Y.P.
Undergraduate of VNTU

DATA SORTING IN INFORMATION TECHNOLOGY

Nowadays new information technologies occupy a very important place not only in specialized but also in everyday aspects of life. Computers are used in business, management, ccommerce, education, medicine and many other spheres of human activity. Because of this, we have an excellent opportunity to work with incredible amounts of information. Moreover, it can cause some difficulties and requires further improvement of technologies and approaches to solving problems.

For example, the need to streamline any value by a certain feature: record some numbers in ascending order, the words - in alphabetical order - occurs in programming very often. If you can compare any two subjects from the given set, then this set is always possible to arrange. The process of organizing information is called "sorting".

Sorting is rearranging data in accordance with a given pattern. For example, the input data is not ordered, and our program is more convenient to treat an ordered sequence. There are situations where pre-sorting the data can reduce a substantial part of the algorithm significantly, and the time - dozens of times. Generally, it is known that in any field of activity, uses the computer for recording, processing and storage of information, all data is stored in databases that also need sorting. A certain orderliness is very important, because the user is much easier to work with data that has a particular order.

The task of sorting in programming is not yet complete. After all, although there are a large number of sorting algorithms, still for the purpose of programming is not only to develop algorithms to sort the elements, but also the development of efficient sorting algorithms. It is known that the same problem can be solved using various algorithms and every time a change of the algorithm leads to new, more or less effective solutions to the problem.

The main requirements for the efficiency of sorting algorithms is, above all, time efficiency and low memory usage. According to these requirements, simple sorting algorithms (e.g. selection sort and sort inclusion) are not very effective.

The sorting is applied in all areas of programming, be it a database or math programs. So, the same method can be implemented based on different structures, methods and use of various programming techniques. That is, we can use conventional linear approach to problem solving, and procedural, and the same algorithm implemented using recursion [1].

The choice of the algorithm depends on the structure of data being processed. In the case of sorting, this dependence is so great that these methods were divided into

two groups - sorting arrays and sorting files (sequences). Sometimes they are called internal and external sorting because arrays are stored in a fast internal memory (random access) and files are stored in slower but more voluminous external memory.

The sorting method is called stable, if in the process of normalizing the relative position of elements with equal values is not changed. Stability of sorting is often desirable, if we are talking about the items that are already ordered according to some secondary keys (i.e., the signs) that do not affect the primary key. The main condition: the selected sort method of the arrays needs to make efficient use of available memory. This means that the permutation of the leading items in the order must be carried out "in the same place". That is, methods where the elements of array a are transferred to the result array b, have no practical value.

Evaluation criteria sorting methods are:

- the number of operations of comparison of key pairs;
- the number of permutations of elements;
- economical use of memory.

Sorting algorithms apart from the criterion of memory saving will be classified by speed, that is, their running time. In accordance with the entered criteria the performance of the sorting algorithms are divided into two types - direct and quick [2].

Direct methods are convenient for explanation and analysis of the basic traits of most sorts that are easily programmable and the program is short, which is also important to save memory. They are based on the repetition of N stages of processing array with a decrease on each of them the number of the compared elements. The efficiency of these algorithms is the value of the order of O(N2). Such methods are convenient to use the so-called "short" arrays.

Quick methods require small amounts of processing stages; however, these stages are quite complicated. Of course the gain in efficiency for such algorithms is obtained on long arrays. The average number of comparison operations depends on the method of sorting and for a rational choice of a method reaches a certain minimum, depending on n - the size of the array (array size - the number of contained entries).

Internal sorting methods can be divided into two groups:

- methods that do not require memory resources;
- methods that require memory resources.

The first group includes techniques such as sampling method, bubble sort, insertion, Shell sorting. The second group includes the method of quadratic selection, the method of fusion, and others. Simple sorting methods (selection, exchange, insertion) require approximately n 2 comparisons. Algorithms that are more complex usually provide a result of nlog2(n) comparisons on average: the sorting method of the Shell, merge, "quick sort"[3].

However, the optimum in any case the sort does not exist, because their effectiveness depends significantly on the type of keys in the array and their preorder.

Thus, the existing algorithms of sorting of arrays differ greatly in terms of complexity, speed, stability, memory requirements, and other parameters. However,

almost every algorithm is the most convenient in any given situation. And even very slow algorithms are used because of their simplicity, for educational purposes.

Literature:

1. Coxon, Anthony Peter Macmillan. Sorting data: Collection and analysis / A.P.M Coxon. p. cm. – (A Sage university papers series. Quantitative applications in the social sciences; no. 07-127) ISBN – 0-8039-7237-7 C# 2010.
2. All-in-One for Dummies. Published by Wiley Publishing, Inc. – 111 River Street, Hoboken NJ 07030- 5774.
3. Herbert Schildt. C++: The Complete Reference, 4th Edition. - McGraw Hill Professional, 2002ISBN – 0071502394.