

Sorting problem

What is sorting?

Very often we need to organize a sequence of elements. The required order can be ascending or descending. Usually, the ascending order is taken by default.

What can we sort?

It is possible to sort data of different types:

- numbers in accordance with the arithmetic order;
- Unicode characters in accordance with their order in the Unicode character table;
- strings (**lexicographically** or by size);
- dates and times in accordance with the chronological order.

Also, it's often possible to sort data of more complex types if we know how to compare items. As a rule, such data has one or more fields called **sorting keys**, by which sorting is performed.

Key features of sorting algorithms

- **Time efficiency.** The size of an array we want to sort is very important for efficiency. If we want to sort an array consisting of a few dozen elements, we can use any sorting algorithm. But what if the array contains a lot of data? In that case, we should use more effective sorting algorithms, otherwise, the results might take too long.
- **Stability.** An array may contain several elements with equal values. Stable sorting algorithms always sort such elements in the same order as they appear in the input. Otherwise, the sorting algorithm is unstable. Stability is important when we sort complex structures such as objects, strings, or something else.
- **In-place/out-of-place sorting.** An algorithm performs **in-place** sorting if it requires only a constant amount of additional space, otherwise, the algorithm performs **out-of-place** sorting. The larger the size of the array, the more additional memory **out-of-place** algorithms require.
- **Internal or external sorting.** An algorithm performs **internal** sorting if the data is kept entirely within the main memory of the computer. In turn, we need **external** sorting when the data does not fit into the main memory of the computing device. In such a case, we keep it in the slower external memory (usually a hard drive).

Many sorting algorithms compare array items during sorting, but some algorithms use other techniques to sort. Such algorithms are also known as **non-comparison sorting algorithms**.

