

УДК 004.9

Artem Polianovskyi¹, Nataliia Zhukova²

¹ student of group CST-128 ZNTU

² PhD (Philology), assistant professor ZNTU

SORTING ALGORITHMS

A sorting algorithm is an algorithm made up of a series of instructions that takes an array as input, performs specified operations on the array, sometimes called a list, and outputs a sorted array.

Sorting algorithms are often taught early in computer science classes as they provide a straightforward way to introduce other key computer science topics like Big-O notation, divide-and-conquer methods, and data structures such as binary trees, and heaps.

There are many factors to consider when choosing a sorting algorithm to use.

Here is a brief list of known used sorting algorithms: Selection Sort, Bubble Sort, Recursive Bubble Sort, Insertion Sort, Recursive Insertion Sort, Merge Sort, Iterative Merge Sort, Quick Sort, Iterative Quick Sort, Heap Sort, Counting Sort, Radix Sort, Bucket Sort, ShellSort, TimSort, Comb Sort, Pigeonhole Sort, Cycle Sort, Cocktail Sort, Strand Sort, Bitonic Sort, Pancake sorting, Binary Insertion Sort, BogoSort or Permutation Sort, Gnome Sort, Sleep Sort – The King of Laziness / Sorting while Sleeping, Structure Sorting (By Multiple Rules) in C++, Stooge Sort, Tag Sort (To get both sorted and original), Tree Sort, Cartesian Tree Sorting, Odd-Even Sort / Brick Sort, QuickSort on Singly Linked List, QuickSort on Doubly Linked List, 3-Way QuickSort (Dutch National Flag), Merge Sort for Linked Lists, Merge Sort for Doubly Linked List, 3-way Merge Sort.

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in a wrong order.

QuickSort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. There are many different versions of quickSort that pick pivot in different ways.

The key process in quickSort is partition(). Target of partitions is, given an array and an element x of array as pivot, put x at its correct position in sorted array and put all smaller elements (smaller than x) before x, and put all greater elements (greater than x) after x. All this should be done in linear time.

Merge Sort is a Divide and Conquer algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves. The merge() function is used for merging two halves. The merge(arr, l, m, r) is key process that assumes that arr[l..m] and arr[m+1..r] are sorted and merges the two sorted sub-arrays into one.

ShellSort is mainly a variation of Insertion Sort. In insertion sort, we move elements only one position ahead. When an element has to be moved far ahead, many movements are involved. The idea of shellSort is to allow exchange of far items. In shellSort, we make the array h-sorted for a large value of h. We keep reducing the value of h until it becomes 1. An array is said to be h-sorted if all sublists of every h'th element is sorted.