

INTRODUCTION TO SORTING ALGORITHMS

1. Definitions

- **Sorting** can be defined as any process of systematically arranging items of any kind.

- Sorting either means:

1. **Ordering**: arranging items in a sequence ordered by some criterion;
2. **Categorizing**: grouping items with similar properties and or behaviors.

In computer science, arranging in an ordered sequence is called "**sorting**". And it is a common operation in many applications.

The most common uses of sorting in computing :

- Lookup or searching : Finding a record in a data base
- Merging of sequences: Copy and pasting, file transfers .
- Data processing in a defined order: Request handling by **queue** and **priorities**

The opposite of sorting or rearranging a sequence of items is a way of arranging data in a random or meaningless order and is called **shuffling**.

A Sorting Algorithm is used to rearrange a given array or list elements according to any defined comparison operator of any kind on the elements from the same collection. The comparison operator is used to decide the new order of the current element in the respective **data structure** or collection.

2. Some Common Sorting Algorithms

- Selection Sort (**Our focus**)
- Bubble Sort (**Our focus**)
- Insertion Sort(**Our focus**)
- Merge Sort
- Quick Sort
- Heap Sort
- Counting Sort

- Radix Sort
- Bucket Sort

3. Sorting Terminology

. In-place sorting

An in-place sorting algorithm uses constant extra space for producing the output (modifies the given array only). It sorts the list only by modifying the order of the elements within the list.

Known in-place sorting algorithms are :, **Insertion Sort** and **Selection Sorts** as they do not use any additional space for sorting the list and in the same way a typical implementation of **Merge Sort** and **counting sort** are not in-place, not in-place sorting algorithms.

. Internal and External Sorting

When all data that needs to be sorted cannot be placed in-memory at a time, the sorting is called **external sorting**. External Sorting is used for massive amount of data. Merge Sort and its variations are typically used for external sorting. Some external storage like hard-disk, CD,...are used for external storage, on the other hand When all data is placed in-memory for processing, then the applied sorting algorithm is called **internal sorting**.

. Stable sorting

Stability is mainly important when we have key value pairs with duplicate keys possible (like people names as keys and their details as values). And we wish to sort these objects by keys, then the sorting algorithm is said to be **stable** if two objects with equal keys appear in the same order in sorted output as they appear in the input array to be sorted

Example : {

[“**domicique**”, “Maried-1.7metres,male”],

[“**domicique**”, “Mukeru-Maried-1.7metres-male”],

[“**atanase**”, “Maried-1.7metres=male”],

```
[“mugunga”,”Maried-1.7metres-male”]  
}
```

If you can sort this algorithm in a way that the first element will continue to be before the second element then your sorting is stable :

4. Things to sort

A number of things can be sorted depending on the application area:

- **In farming:** you may sort your crops potatoes,bananas or your animals
Pigs,cows,dogs
- **In the kitchen:** various types of cereal, dry pasta, beans,...
- In Business: money, notes,coins
- **information architecture (Our focus) :**
Alphabetically — simply, in alphabetical order
- Chronologically — by time; for instance by when the document was created or what year was the report concerned
- Magnitude — some kind of ranking; for example from the lowest price to the highest, from the shortest nails to the longest

Sorting containers:

- Open space,Trash beans, baskets,**Data Structures (Arrays,Lists,nodes,...)**

Every day applications of sorting:

- In Cash processing
- School result proclamation
- While eating
- Toy clean-up

Usual Sorting criteria:

- color,shape , weight, size ,test,....

Array Sorting Example:

```
var myArray={29, 64, 73, 34, 20}; // A collection of 5 numbers
```

Let us sort this array using an algorithm that divide the array into two sub arrays: sorted and unsorted then take the first element of the unsorted array and swap it with the Smollett element of the unsorted array producing the sorted array.

Step 0. 29, 64, 73, 34, 20.

Step 1. 20, 64, 73, 34, 29.

Step 2. 20, 29, 73, 34, 64.

Step 3. 20, 29, 34, 73, 64.

Step 4. 20, 29, 34, 64, 73.

Note :It took us exactly 5 steps as the number of elements to complete the sorting of the entire collection

As you can see sorting is not only fun, but it's also building those important early mathematical and scientific reasoning skills.

Why Sorting Algorithms are Important

Since sorting can often reduce the complexity of a problem, it is an important algorithm in Computer Science. These algorithms have direct applications in searching , database management, divide and conquer methods, data structure efficient usage, and ...

Array Sorting Example:

```
var theArray={29, 64, 73, 34, 20};
```

Let us sort this array using an algorithm that divide the array into two sub arrays: sorted and unsorted then take the first element and swap it with the Smollett element of the unsorted array producing the sorted array.

Step 0. 29, 64, 73, 34, 20.

Step 1. 20, 64, 73, 34, 29.

Step 2. 20, 29, 73, 34, 64.

Step 3. 20, 29, 34, 73, 64.

Step 4. 20, 29, 34, 64, 73.

Note :It took us exactly 5 steps as the number of elements to complete the sorting of the entire collection