

Bubble Sort

An simple algorithm for sorting data.

Bubble sort works on the idea of “bubbling” a value up to the surface/“end”.

Given these starting values:

[5, 8, 2, 1]

It is obvious to a human that the sorted version would be:

[1, 2, 5, 8]

But, our goal is to write an algorithm to perform this job, so we need some very specific step-by-step instructions of how to do it.

Imagine then, that instead of 4 numbers, we had:

[6, 58, 31, 7, 9, 5, 42, 7, 36, 18, 5, 8, 92, 50, 12, 4, 3, 67, 8, 9, 6, 44, 6]

Sorting those is not something a human could do just by looking quickly. So how could we go about it?

Well, what if we, one number at a time, find the largest number, and put that at the end of the list?

[6, 58, 31, 7, 9, 5, 42, 7, 36, 18, 5, 8, 92, 50, 12, 4, 3, 67, 8, 9, 6, 44, 6]

[6, 58, 31, 7, 9, 5, 42, 7, 36, 18, 5, 8, 50, 12, 4, 3, 67, 8, 9, 6, 44, 6, 92]

And then did that same thing again?

[6, 58, 31, 7, 9, 5, 42, 7, 36, 18, 5, 8, 50, 12, 4, 3, 67, 8, 9, 6, 44, 6, 92]

[6, 58, 31, 7, 9, 5, 42, 7, 36, 18, 5, 8, 50, 12, 4, 3, 8, 9, 6, 44, 6, 67, 92]

If we just repeated that process, again and again, we would eventually have a sorted list.

The one issue is that

“find the largest number”

and

“put that at the end of the list”

might look like reasonable easy things to do, but are actually multi-step processes that a computer can't just do immediately.

In order to create a proper algorithm, we need steps that a computer is easily capable of.

With Bubble sort, we consider that the computer is capable of two simple tasks:

- a. Comparing two values.
- b. Swapping two values in a list.

As it turns out, with just those two processes, we can completely implement the previous behaviour, and so create an algorithm that sorts any information!

Going back to the easy numbers:

[5, 8, 2, 1]

Let's compare the first two numbers:

5, 8

We will simply ask if the first of those two numbers is larger, if it is, we will swap the numbers; if not, we will do nothing.

[5, 8, 2, 1]

In this case, 5 is not greater than 8, so we do nothing. Let's move on to the next two numbers:

8, 2

Same process, is the first greater than the second?

In this case, yes, so we will swap them.

[5, 8, 2, 1]



[5, 2, 8, 1]

And again, the same process:

8, 1

8 is greater, so we will swap them:

[5, 2, 8, 1]

[5, 2, 1, 8]

If you look closely, we have achieved

“find the largest number”

and

“put that at the end of the list”

The only thing left, is to repeat the process!