Hi, welcome to Algorithm implementation: Sorting in Python.

☼ We will discuss how to code bubble and merge sort algorithms in Python, In general, a good software engineer understands the algorithms they're using. The study of algorithms is a key part of computer science and software engineering, and by understanding those, a developer can build upon and modify them.

☼After this lesson, you should be able to:

Use sorted function in Python

Explain the importance of coding sort algorithms in Python

Code bubble sort in Python

Code merge sort in Python

Read and understand other sorting algorithms written in Python

Apply your knowledge & understanding of sorting algorithms to your problem solving in Python

☼

In this lesson, we will cover a few topics. Let's go through them in details one by one.

☼Let's recall the application: After Find the Distance to N Locations, sort them by distance.

☼Most general-purpose programming languages have access to standard library functions, templates, subroutines, or methods that are available to perform sorting. In Python, The sorted function (not a method, a function) will sort any collection. It will:

- Separate the collection into individual elements.
- Sort those elements.
- Return the elements in a sorted list.
- Return a new sorted list without changing the original list

☼ Normally, Sorting a list of lists will sort on the first item in each list. Python also has tools to sort on values other than the first. Let's see how it works to Sort coffeeshops by distance. Suggested code is given. Now you already have the knowledge of function, list. It is easy for you to understand the code.

Python provides a module named operator that contains the useful item getter for sorting values other than the first and allows sorting on multiple values.

☼Python provides convenience functions to make accessor functions easier and faster. Using those functions, the program is simple and easy. By changing the parameter of the function, we can easily Sort the list by any field. You can try to run the program.

☼Looks great? Yes. The sort algorithm is efficient. What algorithm does python's sorted() use? Python uses an algorithm called Timsort:

Timsort is a hybrid sorting algorithm

- It is derived from merge sort and insertion sort and designed to perform well on many kinds of real-world data.
- It was invented by Tim Peters in 2002 for use in the Python programming language.
- The algorithm finds subsets of the data that are already ordered, and uses the subsets to sort the data more efficiently. This is done by merging an identified subset, called a run, with existing runs until certain criteria are fulfilled.
- Timsort has been Python's standard sorting algorithm since version 2.3.
- It is now also used to sort arrays in Java SE 7, and on the Android platform.

☼You might wonder, since Python has such efficient sort algorithm, why do we still need code sort algorithm by ourselves?

The sorting algorithms in programming languages cover a subset of possible situations that a software engineer is likely to encounter across the course of a career. If you don't know when they work well and when they don't, you're likely to misuse them.
In addition, sorting algorithms are quite easy, and if a student doesn't understand those, the graph algorithms that come later may be too difficult. It's training for the future.

They are fundamental problems that illustrate well the algorithmic complexity , which you will learn in later lesson. You will have the ability to tackle higher level problems, and to make wise decisions about how to reduce the algorithmic complexity of you code with your grounding in those simpler kinds of problems . Let's start from the simplest sort algorithm.

☼As we have learned, Bubble sort, repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass go through the list is repeated until no swap is needed, which indicates that the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller or larger elements "bubble" to the top of the list.

The algorithm, pseudocode, flowchart of bubble sort is given here.

Note that this sorting algorithm involves a pair of nested loops over the list size (blue and red), meaning that the calculation cost will go as the square of the input size (here, an N-element list);

☼The suggested code is given. The flow chart is converted to python code.

Is it efficient?

Any possible improvement?

☼Let's recall: The number of items that are to be sorted decreases while the number of pass increases. n-th pass finds the n-th largest element and puts it into its final place.

☼ **The computational cost can be halved by adjusting the inner loop conditional to be "range(len(alist)-passnum-1)", as one more element reaches its final position after one pass.** So, the inner loop can avoid looking at the last n − 1 items when running for the n-th time: **Inner loop executes n-1 times at first, linearly dropping to just once. On average, inner loop executes about n/2 times for each execution of the outer loop**

Any possible further improvement? Recall: Since each pass places the next largest value in place, the total number of passes necessary will be n − 1. So, for simplest implementation, the outer loop needs to run n − 1 passes.

☼Quick check: What is the output of the following code?

☼After n-1, that is 5 passes, get the sorted list. However, in this example, the input is the best case, which is a sorted list, if the program is intelligent and efficient, it should stop immediately after it realizes that the list is sorted.

☼How can we know if the list is sorted or not? The pass go through the list but no swaps are needed, which indicates that the list is sorted. How to implement? As we have learned, the break statement is useful for stopping computation when the "answer" has been found or when continuing the computation is otherwise useless. Now, the array is already sorted, but the algorithm does not know if it is completed. The algorithm needs one whole pass without any swap to know that it is sorted. So swapped is set to be the flag to indicate if swapping happens or not.

☼Run the improved code with same example. For best case, one pass is sufficient.

☼As we know, we need to learn different sort algorithms as there is no single best one. We have learned that Merge sort is quite different from the bubble sort. It is an example of a divide-and-conquer style of algorithm, in which a problem is repeatedly broken up into sub-problems, often using recursion , until they are small enough to solve; the solutions are combined to solve the larger problem. The idea behind merge sort is to break the data into parts that can be sorted trivially, then combine those parts knowing that they are sorted.

☼The algorithm is described clearly here.

Step 1: If the input sequence has fewer than two elements, return.

Step 2: Partition the input sequence into two halves.

Step 3: Sort the two subsequences using the same algorithm.

Step 4: Merge the two sorted subsequences to form the output sequence.

Let's start coding.

☼

"Sort the two subsequences using the same algorithm. " A logical way to write the program is to use recursion , where each recursive call splits the data in two parts until there is only one element. Once the data has been split into individual components, the merge stage creates sorted pairs. That is, the lowest level of recursion combines the individuals into sorted pairs, and returns to the next level where the pairs are combined into fours, then eights, and so on until at the highest level the list is completely sorted.

☼In merge function, There are now two sorted sublists of length N//2. We need to Merge them into one list of length N. A beginning is made from the starting parts of both lists

Step 1. Compare the first elements of both lists one by one.

Step 2. Add this value to the list of "sorted items" and move the smaller element out of the list it was found in.

Step 3. Repeat the Process Until Only a Single List is Left Remaining

Step 4. One list should still contain elements. This list is sorted. Move its contents into the result list.

☼The complete merge function code with comments is given here.

☼You can try to run the program after adding application part.

☼Quick Summary:

In Python, The sorted function will sort any collection.

With the module named operator, sorted function can sort values other than the first and allows sort on multiple values.

The sorting algorithms in programming languages cover a subset of possible situations that a software engineer is likely to encounter across the course of a career. If you don't know when they work well and when they don't, you're likely to misuse them. We have learnt coding bubble sort and merge sort in Python in this lesson. It's training for the future.

I hope you enjoy this lecture.

And, I will see you next time.