Sorting Algorithms

The first section

Peace, mercy and blessings of God. Welcome to this lesson.

My name is Amer Ahmed Jose, a computer software developer; please accept my sincere greetings from Sultan bin Abdul Aziz Center for Science and Technology (Scitech), Al-Khobar, Kingdom of Saudi Arabia.

This lesson is about a very important topic. As you know, we live in the era of information technology and computing, where data increases steadily and enormously, increasing the demand for sorting and processing data in order to benefit from this data .

Arranging a set of elements is one of several problems faced by programmers, for example, arranging students in alphabetical order or according to their scores in a certain subject, or by their GPA to recognize the outstanding students.

Have you ever wondered how the order of a set of elements is?

Is it possible to write a systematic method of arrangement as steps or programming codes that can be fed to the computer to do such a process?

Think about this issue and I will get back to you after a few minutes.

The second section

Welcome again, I think you have found one way or more.

Finding a systematic method to solve a problem is called the solution algorithm.

An algorithm is a step-by-step mathematical and logical procedure, needed to solve a problem.

It is called an algorithm, after the Muslim mathematician, Muhammad ibn Musa al-Khwarizmi, who lived during the Abbasid Caliph al-Ma'mun rule, and administered the House of Wisdom in the ninth century AD. He was the author of *"Hisab al-jabr w'al-muqabala"*, and by translating it into Latin languages, the words Algebra and zero "cipher" have been added to those languages. He also was a major participant in solving quadratic equations. That is when the word "Algorithm" was added to Latin languages.

Let us introduce one of the Sorting Algorithms:

Bubble Sort Algorithm:

Suppose we have a set of numbers you need to arrange:

1	2	3	4	5
8	7	5	1	2

This method involves the following steps:

1. Compare the first element with the second one.

2. Switch the two elements if the first is greater than the second.

3. Compare the second element to the third, and swap if the second is larger than the third.... Repeat this process up to the last two elements in the series.

4. By applying this process, we get the largest element at the end of the list.

5. The process is repeated for the rest of the elements.

Example:

Let us consider the following set of numbers:

1	2	3	4	5
8	7	5	1	2

Try to arrange them in this way and I will get back to you.

The third section

Welcome again. Let us solve the previous problem together:

Our example:

1	2	3	4	5
8	7	5	1	2

Compare 8 and 7, and swap them because 8 is greater than 7

7	8	5	1	2

Compare 8 and 5, and swap them because 8 is greater than 5

7	5	8	1	2

Compare 8 and 1, and because 8 is greater than 1, we swap them.

7	5	1	8	2

Compare 8 and 2, and also we swap them because 8 is greater than 2.

7	5	1	2	**8**

Thus, we have obtained the largest element 8 at the end of the set.

The process is repeated to arrange the rest of the elements through the following steps

Compare 7 and 5, and swap them because 7 is greater than 5

5	7	1	2	**8**

Compare 7 and 1, and swap them because 7 is greater than 1

5	1	7	2	**8**

Compare 7 and 2, and swap them because 7 is greater than 2

5	1	2	**7**	**8**

By this, we have placed 7 in the right order.

Back again, compare 5 and 1, and swap them because 5 is greater than 1

1	5	2	**7**	**8**

Compare 5 and 2, and swap them because 5 is greater than 2

1	2	**5**	**7**	**8**

By this, we have placed 5 in the right order.

Go back to the beginning and compare 1 and 2 and since 2 is greater than 1, there is no need for swapping. Thus, our list of elements has been arranged in the right order.

1	**2**	**5**	**7**	**8**

This method is the Bubble Sort Algorithm and it is named so because the sorted element is moved until it reaches the right place in the set just like air bubble moves up in the water.

Now … Do you think that there is a better way to order?

Let us introduce another method called:

Selection Sorting Algorithm:

Perhaps it was one of the ways in which you came up with in the first activity.

This method includes the following steps:

1. Find the smallest element in a list of elements

2. Swap it with the first element in the list.

3. Repeat the previous two steps with the remaining elements (Starting from the second element, followed by the third one and so on)

Try to sort the previous set of elements using this method and I will get back to you after a few minutes.

The fourth section

Welcome again. Now, we sort the elements according to selection sorting algorithm:

1. We choose the smallest element in the list which is 1 in this case.

2. We swap it with the first element in the list (8)

1	2	3	4	5
1	7	5	8	2

Repeat the previous steps 1, and 2 to sort the rest of the elements

Swap 2 with 7

1	2	5	8	7

The element 5 keeps its place because it is smaller

1	2	5	8	7

Swap 7 with 8

1	2	5	7	8

By doing the last step, we have the list in order.

Note that the sorting of the elements was done in the opposite direction, from the smallest element to the largest, and that the elements were switched only to put them in their final position, and this is why selection sorting algorithm is faster than bubble sorting algorithm.

Now, would you be surprised if I told you that there is even a faster way than the previous methods?!

But it is a little more complicated.

It's quicksort algorithm:

It is based on the following steps:

1. Select an item from the list randomly .This element is called the pivot (for example, the element in the middle)

2. Put smaller elements in a sub-list called smaller elements (L) to the right of the pivot.

3. Put larger elements in a sub-list called larger elements (G) to the left of the pivot.

(And thus ensure putting the pivot in the right place)

4. Repeat the steps for the two sub-lists L and G are then recursively* sorted

* Recursively means that the repeated process of dividing a problem into similar and simpler divisions, to find to the solution.

You can refer to the Towers of Hanoi prepared by Dr. Richard Larson on Blossoms for more information about this important and interesting lesson.

Now, I'm going to use the quicksort method to sort the previous list, but after a short break

(Break for five seconds)

Section 5

Let's solve the example using the quicksort method

1	2	3	4	5
8	7	5	1	2

1. I will choose an element randomly and let it be the third element (5) as a pivot.

8	7	5	1	2

2. Put the elements smaller than 5 (1 and 2) in the sub-list of the smaller elements (to the right of the pivot)

1	2

3. Put the elements greater than 5 (7and 8) in the sub-list of the larger elements (to the left of the pivot)

8	7

The list becomes as follows:

1	2	5	8	7

4. We repeat the previous steps for the two sub-lists:

A – Sort the small sub-list: We select (1) as a pivot.

B – A smaller list does not exist in this case, and the large list consists of only one element so it is already sorted.

C – The sub-list becomes:

1	2

Sort the large sub-list as follows:

A -We select (7) as a pivot.

B - A smaller list does not exist in this case, either. The large list consists of only one element (8) so it is already sorted, and we place it to the left of the pivot.

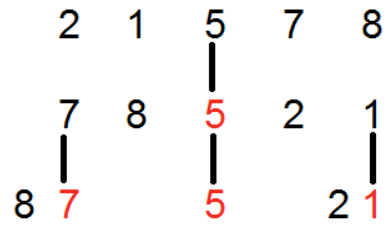
C – The sub-list becomes:

7	8

We combine the small sub-list, pivot, and large sub-list, to have the final list in order:

1	2	5	7	8

I will repeat the sequence of steps graphically and briefly



The elements are 8, 7, 5, 1, & 2.

Pivot is 5.

1 and 2 is the small sub-list.

8 and 7 is the large sub-list.

Sort the small sub-list: 1 is the pivot and 2 is the large list to left of the pivot.

Sort the large sub-list: 7 is the pivot and 8 is the large list to left of the pivot.

The final list will be in order: 12 578

Perhaps you think that this method is not better than the previous two algorithmic methods, and for that I suggest that you sort the following in cooperation with your teacher:

1	2	3	4	5	6	7	8	9	10
1	3	7	8	4	9	5	2	6	0

Meet you after a few minutes

Section VI

Welcome again. I hope you have enjoyed sorting the example according to the quicksort algorithm, let's sort this list together:

1	2	3	4	5	6	7	8	9	10
1	3	7	8	4	9	5	2	6	0

We choose the pivot to be (4) for example, (the fifth element in the list)

First, we form the small sub-list to the right of the pivot, which is

1	3	2	0

And then, we form the large sub-list to the left of the pivot, which is

7	8	9	5	6

So that the following list is produced:

1	3	2	0	4	7	8	9	5	6

Sort the small sub-list and the large sub-list in the same way by choosing 2 as a pivot for example for the small sub-list and 8 for the large one.

1	0	2	3	4	7	5	6	8	9

3 is only element so it is already sorted, and so is the element 9.

1	0	**2**	**3**	**4**	7	5	6	**8**	**9**

Now, sort the remaining two sub-lists:

The sub-list:

1	0

And the sub-list:

7	5	6

In the same way

1	0	**2**	**3**	**4**	7	5	6	**8**	**9**

So that the list becomes finally in order:

0	1	2	3	4	5	6	7	8	9

But..... What if I asked you to sort a list of more than 100 elements for example, or perhaps a thousand or ten thousands?

Is it possible to compare these algorithms to tell which one is the fastest algorithm and the best to use?

Think about a solution, and I will come back to you shortly.

Section VII

How can sort a list of 100 elements for example?

How can we even sort a list of perhaps thousands or millions of records to speed up the search in the databases?

Yes, here comes the role of the computer to accomplish these tasks on our behalf.

As you have noticed, the algorithmic steps are simple and few. They can be easily programmed in any programming language you have already studied, if you have knowledge required for dealing with lists or matrices.

This is an example of a program that performs the sorting according to the three algorithms which we have reviewed together. It orders three identical sets of random lines of different lengths simultaneously to show us which algorithm is faster as well as a visual representation of the sorting steps.

We run the program. As we can see, there are three lists which will be sorted according to the three algorithms and I added extra time to delay the process so that we can see the sorting steps.

We press the Start button to start the sorting process. We find that the first list has been sorted very quickly by the quicksort algorithm and then followed by selection sorting algorithm while the bubble sort algorithm is still working slowly and raising each element gradually resembling a bubble of air moving up in water to reach the top.

So quicksort algorithm is the fastest method.

I hope you have enjoyed this lesson and that it motivates you to learn more about mathematics, programming, and writing algorithms that can be useful to solve other problems that you may confront in your daily life.

If you need any help or if you have any questions, you may e-mail me using the following e-mail address:

amer[at] salaamsoft.com

Good-bye and peace be upon you and God's mercy and blessings.