What is a computer algorithm?

To make a computer do anything, you have to write a computer program. To write a computer program, you have to tell the computer, step by step, exactly what you want it to do. The computer then "executes" the program, following each step mechanically, to accomplish the end goal.

When you are telling the computer ỹ Ḏ Ấto do, you also get to choose Ḏỷ ỹ it's going to do it. That's where **computer algorithms** come in. The algorithm is the basic technique used to get the job done. Let's follow an example to help get an understanding of the algorithm concept.

Let's say that you have a friend arriving at the airport, and your friend needs to get from the airport to your house. Here are four different algorithms that you might give your friend for getting to your home:

The taxi algorithm:

1. Go to the taxi stand.
2. Get in a taxi.
3. Give the driver my address.

The call-me algorithm:

1. When your plane arrives, call my cell phone.
2. Meet me outside baggage claim.

The rent-a-car algorithm:

1. Take the shuttle to the rental car place.
2. Rent a car.
3. Follow the directions to get to my house.

The bus algorithm:

1. Outside baggage claim, catch bus number 70.
2. Transfer to bus 14 on Main Street.
3. Get off on Elm street.
4. Walk two blocks north to my house.

All four of these algorithms accomplish exactly the same goal, but each algorithm does it in completely different way. Each algorithm also has a different cost and a different travel time. Taking a taxi, for example, is probably the fastest way, but also the most expensive. Taking the bus is definitely less expensive, but a whole lot slower. You choose the algorithm based on the circumstances.

In computer programming, there are often many different ways -- algorithms -- to accomplish any given task. Each algorithm has advantages and disadvantages in different situations. **Sorting** is one place where a lot of research has been done,
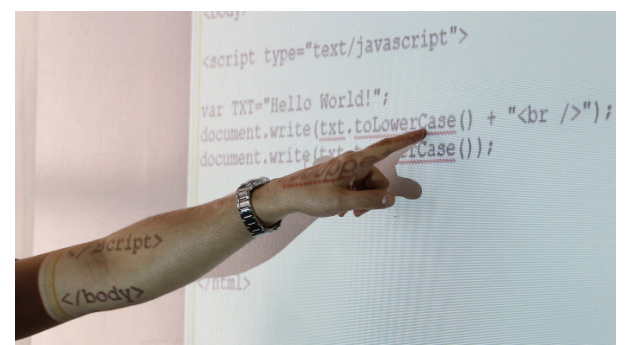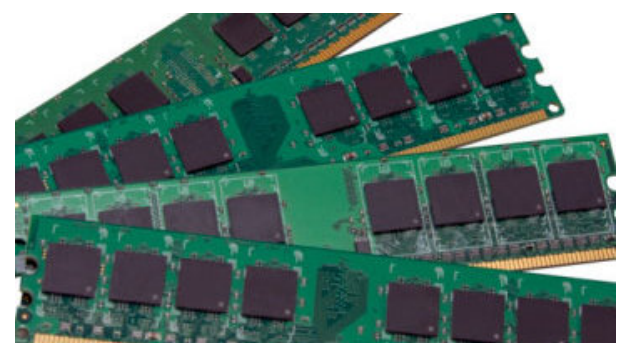
because computers spend a lot of time sorting lists. Here are five different algorithms that are used in sorting:

- **Bin sort**
- **Merge sort**
- **Bubble sort**
- **Shell sort**
- **Quicksort**

If you have a million integer values between 1 and 10 and you need to sort them, the **bin sort** is the right algorithm to use. If you have a million book titles, the **quicksort** might be the best algorithm. By knowing the strengths and weaknesses of the different algorithms, you pick the best one for the task at hand.

Here are some interesting links:

- [whatis.com: Algorithm](#)
- [How Java Works](#)
- [How C Programming Works](#)
- [How PCs Work](#)
- [How Encryption Works](#)
- [How File Compression Works](#)

[Print](#) | [Citation & Date](#) | [Reprint](#)