Algorithms and what you should know about them

(by Jana Varečková)

Algorithms. You've heard the word before, this one nerdy friend of yours who works for an IT company always talks about them. But do you know what he means? Maybe you can sometimes perceive this word differently than computer scientists. Your understanding can be blurry and we want to show you there is no reason to be afraid of algorithms. And power comes with knowledge ... so please read carefully.

What is an algorithm?

Although there is no generally accepted formal definition of an algorithm, there are a few that try to define it. But they are still not correct, because with time the word algorithm adds new meanings and everyone perceives details differently.

When first learning about algorithms, our teacher compared them to recipes. Because a recipe is a collection of simple instructions. When you accomplish one task, you can move to the next. Imagine you have some vegetables, rice and meat as an input and you want to cook curry, which would be your output. This can be your problem to solve and you have to come up with a solution (with a recipe - an algorithm - how to do it).

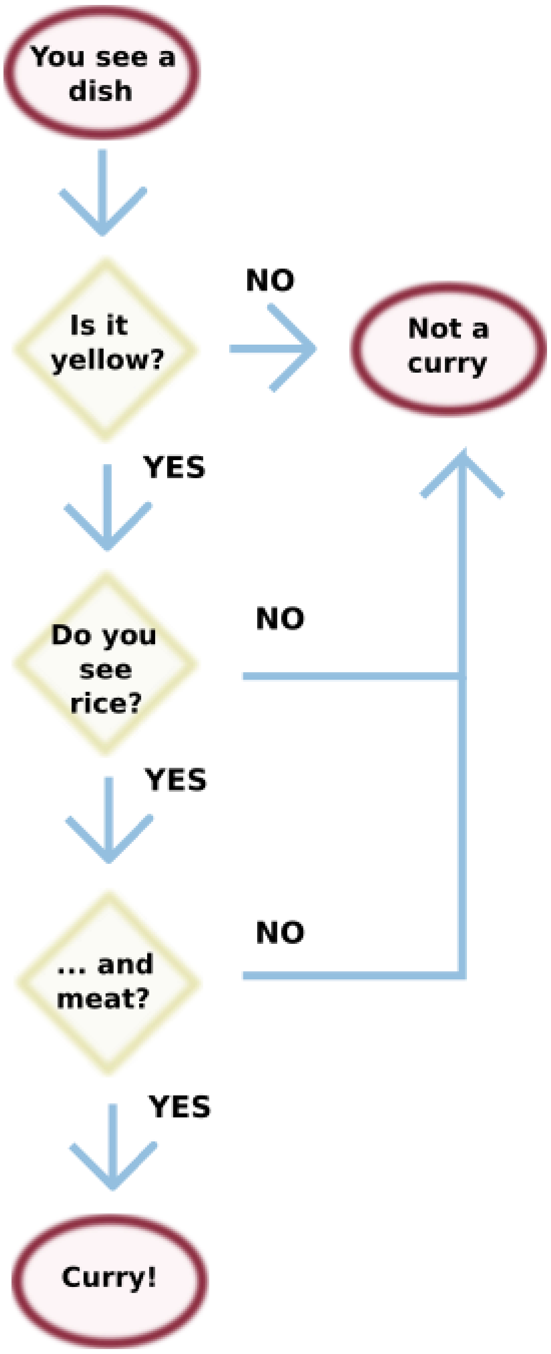
One solution can be: The first step is to prepare the meat (get rid of some parts you don't like, use salt, pepper, etc.). Second - peel the vegetables. Third - put the vegetables into the frying pan. And so on. But maybe you are someone who likes peeling the vegetables first and preparing the meat as the second step! This would be a different recipe (algorithm), but you would still accomplish the task. You have to keep in mind that at the end you would serve your amazed friends curry. During the process you don't want to spend a lot of time in the kitchen, you want to be as efficient as possible, so you won't start cooking your rice as the last thing, because you know it takes quite long to cook it. Which could ruin your end-meal.

Even if you write some steps on how to cook the curry, you still have some freedom, for example which knife to use. The knives represent different programming languages. The recipes (algorithms) are independent of them because they are written as a list of steps (set of instructions in pseudocode). We will come back to the pseudocode later.

Why are we telling you all this things that are clear? We want to make an analogy on how the computer scientists make the algorithms and then implement them. So … mostly we want recipes (algorithms) to have ingredients like meat, rice, vegetables (the input) that is somehow cooked (processed). You have to finish your meal at some point (the process needs to be finite) and result in a curry (output), you probably don't want to spend too much time in the kitchen (the process should be efficient) and obviously you want curry, not something else (the process should be definite).

Now we come to the above mentioned pseudocode. For an example what a pseudocode is, we will use another algorithm determining whether you cooked the curry:

```
if the the color of the dish is yellow and you see rice and meat
        the food is probably curry
else (if it doesn't fulfill the above mentioned conditions)
        we are sorry, maybe it was cooked with love, but it is not the curry we wanted
```



If you are more of a visual person, you will like a flowchart (diagram) of an algorithm (shown on the right side). So you can see that according to some conditions (Is it yellow? Do you see rice and meat?), the algorithm can have different output (either you decide it is curry or not).

What do programmers do with an algorithm?

That is an easy one. We implement them. We use our imagination (we can be creative too!) and program.

When a programmer picks an algorithm (the cook chooses the recipe) they still have a lot of freedom with how to implement it in a programming language (which knife to use).

How do programmers decide which algorithm to use?

Mostly it depends on the efficiency of the algorithm, its correctness and memory load (low efficient food processor which cuts the vegetable into too big pieces and use too much energy can represent an inefficient, incorrect and high memory load algorithm). Sometimes the correctness can be an issue. When you need your output to be really precise (try cutting meat into 2x2x2 cm^3 cubes) it takes a lot of time (and memory) to count it. If you want the algorithm to be efficient, you loose a bit of precision (the cubes will not be precise, but it will take less time).

Where can you find algorithms?

Everywhere! Really! Ok, maybe not everywhere, but when you use your smartphone, you will be definitely surrounded by them.
Algorithms can be simple, for example when you order your files alphabetically, or much more complex. What we see mostly is composed of more than one algorithm.
One really good example is Google. They wrote a story on how searching on Google works. One algorithm autocompletes what you write, another one checks your spelling. Even though this story is nicely done, it doesn‘t show everything that is underneath.

The most common application of algorithms is sorting. Google sorts web pages, Netflix recommends music, your email is sorted into spam or important folder. Advertisement is personalized after an algorithm analyzes your demographics and online behaviour.

How do algorithms shape the world around us and why you should know about them?

Do you know that Facebook's News Feed is filtered? That some posts your friends wrote are never shown on this page? If yes, congratulation, you are one of the minority (why are you even reading this? ;). We think it is important to know about the algorithms shaping what we see, how we perceive the world.

Facebook filters the content of home page. At first it wasn’t necessary, but with increasing number of users and their interaction, it became. In one study the majority of participants at first not knowing (and therefore annoyed) about the News Feed Algorithm began to like it after the end of the study. At first when they realized they don’t see everything their friends post (they saw filtered list of stories), they were dissatisfied, but with time they realized mostly what they don’t see, they are not interested in (for example how other people interact with each other).

Should I be concerned that algorithms rule my world?

Well … It is a hard question.
It was shown that when Facebook changed its news feed algorithm to study how social media posts affect people's emotions, they found out that people tend to reflect the emotions of their friends. So when the algorithm is showing your friends posts where they write how happy they are, you will most likely be happier than when they show you how miserable they are.

But on the other hand algorithms are making our life a lot easier. Imagine a world without Google! Or without you GPS navigation, how would you come to a shop with organic food?

Are algorithms error-free?

Just like recipes can always be improved, we, as computer scientists have to admit, that the programs we write are not perfect. We try to program them as flawlessly as possible, but algorithms are mostly really complex and people make mistakes, therefore you can find bugs (the programmers slang for error) probably in all programs. But don‘t worry! When we look for bugs (a process called debugging), we try to correct them as quickly as possible. Because programs are not flawless, many people stop using programs after seeing them err as they think that when it made a mistake once, they are totally worthless and they can‘t learn. But this is not true. Sometimes your GPS navigation chooses a longer way, but you have no reason to stop using it! Maybe it didn‘t have enough information about the traffic, or it had a bug, but the development team have just found it and after the next update, it won‘t make the same mistake again.

Why are algorithms invisible?

Thanks to good design, we are actually not bothered by many unnecessary things. It would be too complex to show the algorithm (or more precisely a lot of algorithms somehow combined), how could you possibly show all the complexity? There are programming teams where members are glad to know how their own code works.

One of the main reasons why algorithms are hidden from our sight is intellectual property protection. Can you imagine working on something for a few years and then you have to show it to the public and everyone can “steal” your ideas?

Another reason is tracking. (Almost) Every click you make is recorded, the same with where your mouse is moving, how long you stayed on that page. They can use this information for targeted advertising or sell this information. These pieces of information can be misused, but on the other hand improve applications you use.

……

Thank you for taking the time to read this article. This article was written on behalf of the *Critical algorithm studies* seminar, TU Wien. If you have any questions or you have found any flaws, please send an email to janavareckova@yahoo.com

References:

http://social.cs.uiuc.edu/papers/pdfs/paper188.pdf
http://www.cyberpsychology.eu/view.php?cisloclanku=2014093001&article=2
http://median.newmediacaucus.org/art-infrastructures-information/seeing-the-sort-the-aesthetic-and-industrial-defense-of-the-algorithm/
http://www-personal.umich.edu/~csandvig/research/Eslami_Algorithms_CHI15.pdf
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2466040

http://culturedigitally.org/2014/06/algorithm-draft-digitalkeyword/
https://en.wikipedia.org/wiki/Algorithm_characterizations
http://www.cnet.com/news/facebooks-mood-study-how-you-became-the-guinea-pig/
http://research.microsoft.com/en-us/um/people/gurevich/Opera/209a.pdf
http://www.math.ucla.edu/~ynm/papers/eng.pdf
http://research.microsoft.com/en-us/um/people/gurevich/Opera/164.pdf
http://ac.els-cdn.com/S0315086015000725/1-s2.0-S0315086015000725-main.pdf?_tid=36a65f52-5f07-11e6-b846-00000aab0f6c&acdnat=1470839706_648342ebf1ad230a29a796f7c025708b
https://en.wikipedia.org/wiki/Computer_program