

Comparing Algorithms

by Devmountain Tutorials



WHAT'S COVERED

This section will explore how to compare and plan algorithms by discussing:

1. COMPARING ALGORITHMS
2. PSEUDOCODE

1. COMPARING ALGORITHMS

Keep in mind, there are many possible algorithms that can solve one problem. As we can see from the previous section, at least two (but in reality, many more) algorithms exist that can choose the top 5 products to recommend to a user on an e-commerce website. It's worth noting that writing an algorithm is a skill in and of itself. Creating a unique algorithm is something that a web developer might be asked to do.

However, what's much more likely is that a web developer would need to choose from a set of existing algorithms for a particular task, and implement one of them. Many known algorithms have already been created for the various use cases we've mentioned in this unit. To get a sense of this task, let's try to compare and contrast the two algorithms we saw in the previous section.

	Algorithm Simple	Algorithm Complex
Number of Calculations	+	+++
Accuracy	++	+++
Easy to Understand	+++	+

The chart above estimates the way that the two algorithms compare. The categories of number of calculations, accuracy, and ease of understanding are some of the common drivers for the decision about which algorithm to choose for a certain task.

With some tasks, such as mapping a route to get from one place to another, accuracy is of the highest importance. You wouldn't want to end up 1 mile west of where a birthday party is occurring, right?

With our task, accuracy isn't as important. While we want to show recommendations that are engaging to our shoppers, the ability to predict what they will want to buy next isn't necessarily going to make or break our e-commerce platform.

Another thing this chart showcases is that there is a tradeoff with different categories. Specifically, and the

number of calculations, or “runtime complexity” increases, so does the difficulty of quickly understanding an algorithm.

It might be in a web developer’s best interest to choose an algorithm that’s easy for the rest of the team to understand, especially if accuracy isn’t the top concern. Algorithms that are easier to understand translate to code that is more likely to stay bug-free in the long run.

Other times, what we gain in accuracy through a more complex algorithm isn’t necessarily worth it. For our very simple algorithm, we get a fairly accurate set of outputs with Algorithm Complex. Even though there aren’t a ton of calculations, we get a pretty good set of product recommendations. The cost of upping the accuracy level by just a little bit through using a lot of calculations just isn’t quite worth it in this case.

2. Pseudocode

Continuing with our study of the two algorithms for e-commerce, let’s talk about an important part of any coders’ languages- pseudocode! Pseudocode is both exactly what it sounds like, but also much more important than it sounds. Pseudocode is a way of notating what a set of computer code will do, step by step, section by section, but without utilizing the actual rules and grammar of a particular programming language.

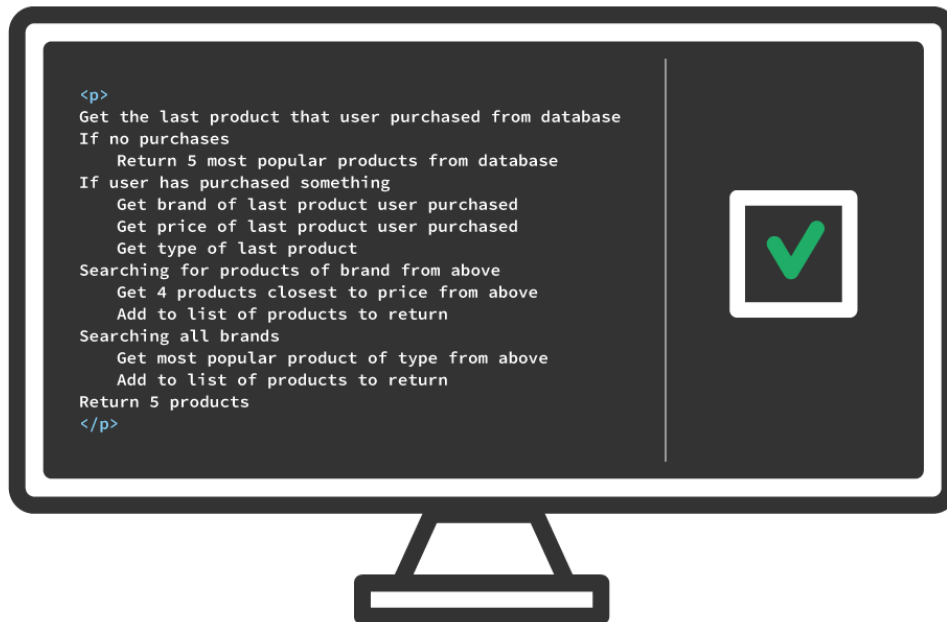
At first glance, this might seem like a tool to simply allow many programmers who don’t speak the same language communicate. While it can serve that purpose, pseudocode actually provides an even broader application.

Pseudocode allows algorithms to be documented in a *codeful*, yet language agnostic manner. Pseudocode is codeful in that the level of detail within pseudocode is fairly similar, line for line, with what you would see in a real codebase.

Pseudocode is also language agnostic; it doesn’t require the knowledge of any one programming language in order to understand it. In fact, a majority of pseudocode doesn’t even require that the reader knows how to program at all!

The other advantage that pseudocode provides is an important planning step in between understanding what the code should do and actually writing code.

Pseudocode for Algorithm A



Similar to writing effectively in English, there are a lot of important choices to make in order to write code in a clear, concise, and efficient manner. Pseudocode serves the purpose of a rough draft or outline for the actual code in a project.



TERM TO KNOW

Pseudocode

Pseudocode is a way of notating what a set of computer code will do, step by step, section by section, but without utilizing the actual rules and grammar of a particular programming language.



TERMS TO KNOW

Pseudocode

Pseudocode is a way of notating what a set of computer code will do, step by step, section by section, but without utilizing the actual rules and grammar of a particular programming language.