Find the Pokémon
The Importance of Algorithms

Bob Brown

Copyright © 2020 by Kennesaw State University

Hello I'm Professor Bob Brown. I've taught college students about computers and computing for more than 20 years. Today we're going to learn about algorithms by playing a find the Pokémon game. Algorithms are important because they're the science in computer science. I'll say this again in a few minutes but, one definition of computer science is the study of algorithms and their realization in hardware and software.

So what is an algorithm? it's a set of steps that describe a method of solving a problem. However, not every procedure is an algorithm. We'll add some conditions that make our definition precise. If you look at the back of a shampoo bottle you'll find a procedure for washing your hair. It goes like this: wet your hair, apply the shampoo, lather the hair, rinse, repeat steps 2 to 4.

There's something missing from this. What is it?

Right! The procedure doesn't answer how many, times so it's not an algorithm. People just know that repeating one time is enough, but a computer would have to be told explicitly.

We need a more formal definition, and our definition goes like this: An algorithm is a well ordered sequence of unambiguous and effectively computable operations that produces a result and halts in a finite amount of time.

It was the last thing that made our shampoo procedure fail the test for being an algorithm. A computer, having no common sense, would just repeat forever. To really understand this we need to look at each part of that definition in more detail.

The first part was a well ordered sequence. That just means the steps are in a defined order. Let's look at an algorithm for planting a seed in a pot. We put the soil in the pot, make a small hole in the soil, put the seed in the hole, cover the seed with soil. These steps have to be in the order shown. Why?

Right! You can't make the hole until you've put the soil in the pot, you can't cover the seed until it's in the hole, and so on.

Let's finish up. We water the soil and we put the pot in the sunlight. Either order will do for the last steps, but you have to pick one.

The second part of the definition is that it must be unambiguous. All that means is that the meaning has to be clear. If you have numbers A, B, and C, and you know what the values of those numbers are, you can perform the addition of A and B, and then multiply the sum by C. That's unambiguous.

If you're just told add and multiply the numbers you don't have enough information to carry out the calculation. That's ambiguous.

The next part is "effectively computable," and that means that the computing agent whether it's a person or a machine can actually carry out the operation, can do what's needed.

Here are our two numbers A and B again. Can we check whether A is greater than B? Yes that's computable.

Can we predict the next lottery winner? No. No one knows how to do that, so it's not computable.

There's something to watch out for and that is there are problems that look computable but aren't. Computer scientists call this kind of problem *undecidable*. It's important that you know this class of problem exists so that you don't spend a lot of time on something that simply can't be done.

A famous example of an undecidable problem is called the halting problem. you can read about it on Wikipedia, but it might be more fun to search for Professor Geoffrey Pullum's poem, "Scooping the Loop Snooper."

The next part is that an algorithm produces a result. Remember our simple definition: an algorithm is a set of steps that describe a method for solving a problem. If there's no visible, result how can we tell whether our problem was solved?

Well, we can't. We need that result because without the result we can't tell whether the problem was solved.

The next part was halts a finite amount of time. Here's an "algorithm" for printing the positive integers. Positive integers are the counting numbers. "Algorithm" is in quotes because we're about to see that it isn't an algorithm at all.

So here it is: set a number to 0, add one to the number, print it, and go back to step two.

How many positive integers are there?

That's right, infinitely many. This will never end because there are infinitely many positive integers. It cannot solve the problem of printing all of them, and so it isn't an algorithm.

Here's our definition all over again. An algorithm is a well ordered sequence of unambiguous and effectively computable operations that produces a result and halts in a finite amount of time.

Remember that informal definition. An algorithm is a set of steps or a description for solving a problem. A computer program, that is the code, implements the steps of the algorithm. You have to have the algorithm before you can write the code.

Programming computers is a lot of fun, so it's tempting to just start writing code. You'll write far better programs in far less time if you plan the algorithms you're going to use before you start writing code.

I said at the beginning that algorithms are the science in computer science. Computer science is a study of algorithms and making hardware and software that can perform the algorithms. When we say hardware we mean the computers themselves. Software is the programming code that runs on the computers. When we study algorithms, we learn that there's often more than one way of performing a task or solving a problem. Some may be better than others, so a part of computer science involves learning which algorithms are best and choosing the right algorithm for the job we have at hand.

If we were in a classroom, we'd play this game in person, with partners guessing the location of each other's Pokémon. At home will have to do some pretend play unless you have a friend who can play with you. There are game sheets with the video. If you have someone who can play with you and you have a printer you should use the 1-A and 1-B game sheets. One of you uses 1-A and the other one uses 1-B.

Our pretend players are Bill and Gina. Each one has a page with 26 Pokémon characters labeled A to Z. Bill and Gina have different pages and they're careful not to let the other player see what page they have. The characters are numbered, but each player can see only his own numbers. They're hidden from the other player.

Each player picks a Pokémon and tells the other the number of the Pokémon that he picked. The other player must guess the letter that's hiding that number.

So each player picks a Pokémon. tells the other player the number, the other player can't see the sheet so can't tell which letter hides that number. Gina picks the Staryu Pokémon, number 2142, and Gina tells Bill the number. Bill also picks a Pokémon and tells Gina the number.

They're going to take turns guessing which letter is hiding the other players Pokémon. Let's go through a little bit of the game and then if you're playing at home pause the video and play until one of you wins. Bill guesses location A. Gina tells him whether he's right or wrong. In this case he's wrong. She also tells him the number that's really at location A.

The game continues for a while with Bill and Gina taking turns making guesses. Eventually Bill guesses location T and Gina tells him he's right.

If you're with a friend, you can play the game. Use worksheets 1-A and 1-B because we're going to play the game again in a few minutes with the other worksheets. Pause the video while you're playing. If you're alone or you didn't print the game sheets please pause the video anyway and take a moment to think about how you would approach this game.

So, we're back from our pause. How did you approach the problem? If we were in a classroom I'd ask you how many guesses you took and what you did. Different students tried different ways including starting with the vowel letters first and starting with the letters in the other player's name first. Sometimes someone gets it right on the first guess. Sometimes someone goes through all 26 possibilities. Mostly the number is somewhere in between, and on the average It should have taken 13 guesses. How did we get 13? Right! That's half the possibilities.

What you've just done is a search algorithm. Some players checked locations one at a time until they found the right one. The one-at-a-time algorithm is called a linear search. You go down the line looking until you hit the right Pokémon. If there's no order to the Pokémon numbers a linear search is the best way to program this problem on a computer. If the size of the list doubled it would take twice as many guesses on the average.

Searching and search algorithms are very important when I was putting this together I did some searching on Google to find the Pokémon characters and one of my results said “About 1,320,000 results in .59 seconds.”

Let's think about that. Google searched a zillion items, got over a million results, and did it in less than one second. Depending on the data there are algorithms that work better than the linear search.

Let's play the game again but with a difference; this time the numbers are in order. Besides telling you the numbers are in order, if we were in class I'd give you a hint to start with the middle of the alphabet. What's the middle letter of the alphabet? Right! M is the middle of the alphabet.

Knowing that the numbers are in order and thinking about my hint, can you think of a way that will let you find the Pokémon in fewer guesses?

There are more printable game sheets. This time, use 2-A and 2-B. If you're going to play at home, pause the video. If you're not going to play at home, still think about whether there's a way to find the Pokémon with fewer guesses, and when you're ready come back from the pause and we'll watch Bill and Gina play the game.

Bill and Gina took my hint. They're going to do a binary search. The binary search algorithm is a divide and conquer algorithm in which we divide the number of possibilities in half with each guess. Gina has picked the Pokémon Vulpix at number 1321. Bill knows the number because Gina told him but he can't see the letters because Gina won't let him see her game sheet. Bill guesses location M.

Why? Right! I gave the hint to start in the middle. M is the middle of the 26 letters. Gina says no, M is 5027. Bill has learned something. What is it?

Right! Not only does Bill know that M isn't right, he knows that the Vulpix he's looking for is before M Why? Because he knows that the numbers are in order and 1321 is less than 5027. That means the Vulpix is between A and L. Bill is going to pick the middle of the letters between A and L.

In his next turn Bill guesses F. Why? Because F is in the middle of A to L. Gina tells Bill the Vulpix isn't at location F, either F is 1910. Bill knows the Vulpix must be before F because he's looking for 1321, less than 1910 and so it has to be between A and E. In his next turn Bill guesses C. Why? Because C is in the middle of A to E.

Gina tells Bill the Vulpix isn't at location C. C is 722 and Bill has gone too far; he has to go the other way. Bill knows the Vulpix must be at D or E. There's no middle between D and E. Bill guesses D and he's right! He got it in four guesses.

If he had been wrong the only other possibility would have been E and it would have taken Bill five guesses. Five guesses is the most it could have taken using this algorithm because the smallest power of 2 greater than or equal to 26 is 5. We'll talk about that more shortly.

This algorithm is a binary search. I'm going to go over how the binary search works but the important thing for you to remember is that it exists. If you need to search things that are in order you can look up how the binary search works and refresh your memory, but you have to remember that it exists to be able to look it up.

The binary search starts with the middle element of the items that we're trying to search. If the value of the middle element is more than the target, the target has to be in the top half. Otherwise, if the value at the middle is less than the target, then the target has to be in the bottom half.

In either case we cut the number of items for the next guess in half, or we could get lucky and hit the target with fewer than the maximum number of guesses.

The binary search takes at most $\log_2(n)$ checks. N is the number of items in our list. It's 26 in our Pokémon game.

Log_2 is the power of two that it takes to get the number of items to be searched. That's the N on the slide. Two to the fourth power is 16, which is not enough to cover 26 items. Two to the fifth power is 32. That's the smallest power of two that is more than or equal to 26, so to search 26 letters, we need at most five guesses using a binary search.

How cool is that?

Each check cuts the number of remaining items in half, so doubling the number of items adds only one more check. The bad news is that this only works if the items to be searched are in order.

A few minutes ago I said there's often more than one way of performing a task or solving a problem. Some may be better than others, so a part of computer science involves learning which algorithms are best so that we can choose the best one.

We've taken a look at two of the algorithms for searching. There are others and there are also algorithms for other tasks.

To search an unordered list, and you only need to search it once, use that linear search. Start at the beginning and go down the line until you find the item you're looking for or you come to the end of the list. If the list is in order, you can use the binary search. If you're going to do many searches, it might pay you to do some pre-work like sorting the item so you can do a binary search or trying a different algorithm. That's one of the things that makes Google so fast.

There are other algorithms for searching. Many other tasks like sorting can have more than one algorithm. It's not the code it's the algorithm!

Our friend Mister Natural has some advice about that: "At home or at work get the right tool for the job." Learn about algorithms and for each program you write choose the best algorithm for the job.

Thank you! I enjoyed making this video. I hope you've enjoyed watching it and that you learned a lot.