Name _____

Let's explore algorithms

Edison's line tracking sensor can detect if the surface below the robot is reflective or non-reflective. You can use this sensor to get Edison to behave in different ways, such as driving until it detects a black line, then stopping. You can also use this sensor to program Edison to follow a black line, even if you don't know what that black line looks like. To do this, you first need to create an **algorithm.**

Jargon buster

An **algorithm** is a broad set of instructions to solve a set of problems. An algorithm lays out a process or a set of rules to be followed in order to solve any problem in the set.

Computer programs often use algorithms, but programs and algorithms are not the same thing. Remember, a computer program is a collection of instructions that tell a computer to perform a specific task. An algorithm lays out the logic for how to solve a whole set of problems, not just one specific task. You can write a computer program that uses an algorithm, but not all computer programs are algorithms.

Algorithms are really helpful because using an algorithm lets a person, or a computer, solve a whole set of problems, even if you don't know every detail.

Why is that?

Let's say you want to teach your friends how to make fruit pies. If you know that all of your friends have apples, you can just write down one recipe for apple pie.

Not all of your friends might have apples, however. What if one of your friends has blueberries, another has cherries, and a third has apples? They cannot all follow the apple pie recipe. You would need to write a separate recipe for each different fruit.

What if you don't know what fruit each of your friends has? How could you teach them to make fruit pies?

No matter what fruit they have, all of your friends need to follow the same basic instructions: make the dough, fill the pie with the fruit, then bake the pie.

This new set of instructions is an example of an algorithm.

In computer programming, we often want to create instructions for a computer to follow in order to solve a whole set of problems. By using an algorithm, we can write a program that will let the computer solve any problem in the set. Without an algorithm, we would need to write a new program for every single problem individually.

Name _____

Follow a black line

You know that you can use Edison's line tracking sensor to detect black (non-reflective) and white (reflective) surfaces. We can create an algorithm that uses this sensor to get Edison to follow any black line.

Why is that?

Let's say you draw a black line for Edison to follow. To get Edison to follow your line, you could write a program which makes Edison drive the exact path of the line. If you make a new line, however, you will need to write a whole new program for that new line.

Instead, you can create an algorithm.

This algorithm will solve a set of problems: 'follow any black line'. Any specific line you make for Edison to follow is a new problem inside this set.

Using the algorithm to guide the logic, you can then write a program which will work for all of the problems in the set. This way, a whole new program isn't needed for each new problem.

You can plan out an algorithm using pseudocode, just like you do when you plan out a program. Here is an algorithm in pseudocode that will allow Edison to follow any black line:

Turn line tracker on

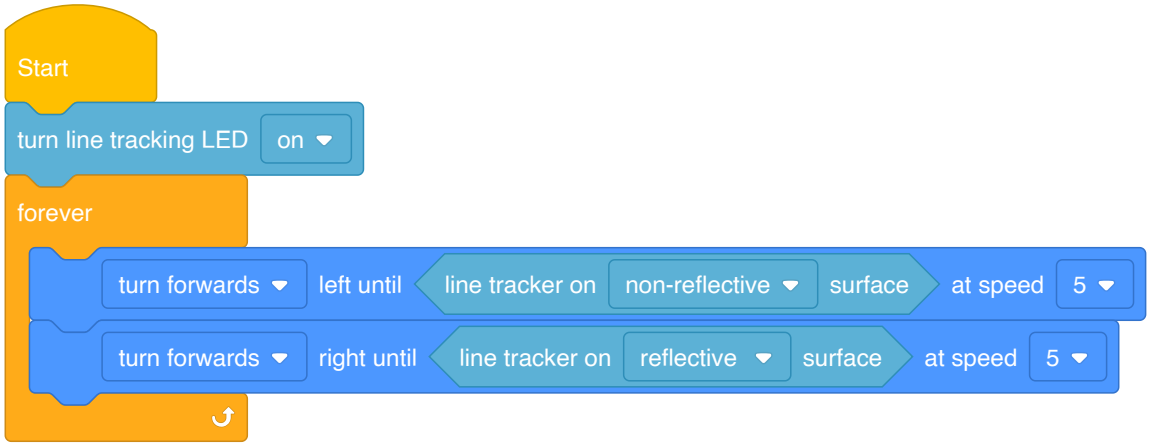
Loop forever

Drive forwards left until the robot detects a black surface

Drive forwards right until the robot detects a white surface

This algorithm says that the robot should drive forwards to the left until the line tracking sensor is on a non-reflective (black) surface. Then the robot should drive forwards to the right until the line tracking sensor is on a reflective (white) surface. The robot should keep doing this behaviour forever.

This algorithm can be used to write a program in EdScratch: Can you see how the algorithm's logic is inside the program?



Write the line-following program in EdScratch and download it to your Edison robot. Use the activity sheet on page 133 to test the program. Remember to start Edison with the line tracker on a white surface, not directly on the black line.

How does the robot move when you run the program? Look at the program and think about the logic in the algorithm. Why does the robot move that way?

Name _____

Follow a different black line

The program you wrote uses an algorithm that is designed to solve any problem in the set of problems: 'follow any black line'. That means this same program should let Edison follow any black line you make!

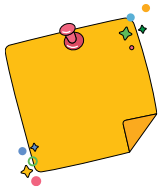
Make your own line to test. Use a black marker on white paper or make a line using black tape on the floor or a desk. Run the program in Edison to test out your line. Can Edison follow your line?

Was Edison able to follow your line? If you had any problems, describe them.
What do you think caused the problems?

Challenge up: There's more than one way to follow a line

To get your Edison robot to follow any line, you need to use an algorithm.

Think about the logic in the algorithm that will allow Edison to follow any black line. At its simplest, what is it saying?



Don't forget

An **algorithm** is a broad set of instructions to solve a set of problems. An algorithm lays out the logic for how to solve a whole set of problems, not just one specific task.

The broad instructions to solve the 'follow any black line' set of problems are: using the line tracker, while moving forward, go one way if on black and the other way if on white. Here's what this very basic algorithm looks like in pseudocode:

```
Use the line tracker
forever
If the robot detects black,
drive forward one way (left
or right)
If the robot detects white,
drive forward the
other way (right
or left)
```

If you were planning out an algorithm to then code, you might write this same logic slightly differently, and you would probably add in some of the details, like in this example:

```
Turn line tracker on
Loop forever
Drive forwards left until
the robot detects a black
surface
Drive forwards right until
the robot detects a
white surface
```

Because the logic is the same in both examples, any programs you make using either example will follow the same logic. Even if the programs look different, they will still solve the 'follow any black line' set of problems.

Just how many different ways could you write a program that will solve the 'follow any black line' set of problems? More than you might think!

What to do

Your challenge is to write at least two different programs that use the basic logic of the 'follow any black line' algorithm.

You will need to plan each program using pseudocode, then code it in EdScratch.

Hint

Pseudocode helps us plan, comments help us debug. Adding comments as you code will help you check your logic and keep track of your thinking, so you can find and fix any issues you encounter when you test your program.

Think about how you can apply the logic of the algorithm in EdScratch. Look at different conditionals, including **until** blocks, **if** blocks and **if-else** blocks. Don't forget about the **Events** category too!

Download each of your programs one at a time. Test each program using the activity sheet on page 133 or make your own line to use as a test space. Both of your programs need to use Edison's line tracking sensor and should be able to follow any black line.

What do your two programs look like? Either write pseudocode, download and share your program files or write an explanation of each of your programs **(use image boxes on next page for photos)**.

Program 1:

Program 2:

Name _____

Program 1:

Program 2:

Activity sheet: Non-reflective border