

# Day 1: The Cantor Set<sup>1</sup>

February 17, 2016

<sup>1</sup> Part of a year-long journey to learn one thing a day.

## Introduction

It only takes a straight line and some imagination. Today's topic will illustrate that simplicity can be an effective tool for discovery.

## Making the Set

The Cantor Set is formed by repeatedly splitting a line and removing pieces from it. A process like this is called a **finite subdivision rule**. The goal of finite subdivision is to systematically divide a shape into smaller, more interesting pieces. One of my favourite examples of this is the Sierpinski triangle.



Figure 1: Pretty things happen when you remove stuff systematically.

By applying subdivision rules, you can generate other amazing shapes that are self-similar when zooming in. These are known as fractals. The Cantor Set is one of the simplest fractals.

The steps for creating the Cantor Set are simple:

1. Take a line from 0 to 1, and split it into thirds<sup>2</sup>.
2. Remove the middle of the line, between  $1/3$  to  $2/3$ .
3. Do this again to the remaining pieces. And again, and again.

<sup>2</sup> We don't need to choose thirds, but that is a common choice. Also, we will focus on the line from 0 to 1, but any line works.

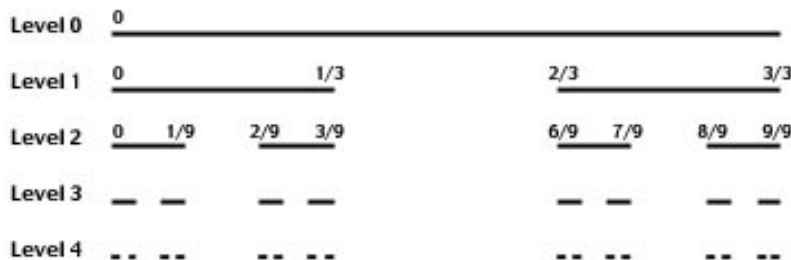


Figure 2: A few iterations. Each time you could zoom in and start all over with a smaller line.

That's it.

### *What's Left?*

We cut a line into lines, and those lines into lines. So what?

After many iterations, do we have anything left?

- We started by cutting out a length of  $1/3$ .
- Then we cut the two remaining segments by  $1/3$  of their length.  
That's two times  $1/9$ .
- Then we cut two remaining segments from those two segments.  
That's four times  $1/27$ .

See the pattern? Stuff Cut Out =  $\frac{1}{3} + \frac{2}{3^2} + \frac{2^2}{3^3} + \frac{2^3}{3^4} + \dots$

You might remember that this is a [geometric series](#). Summed together, the length of all the lines we cut out is one, which is the length of our original line!

But there isn't nothing left after infinite subdivisions. Remember we keep end points. Take  $1/3$ , for example. Once we cut the line there, it became an end. The middle third of any future line will never be that end. The same idea applies to other cutting points at any iteration.

What is less easy to see is that we also have infinitely many other points. Let's try to show that.

### *How Many Points Are Left?*

What is remarkable about the Cantor Set is that there are as many numbers left behind as there were to begin with.

Does that sound impossible?

You might have heard before that the numbers between 0 and 1 are not "countable," but what does that really mean?

The plan is to show that any number left over can be matched with a number we had to begin with.

The numbers won't be the same, since we removed some numbers. However, if we can always provide a unique number from the Cantor Set for any unique number we originally had, we must have the same count of numbers.

## Review

Before we go deeper, it is important to review how to write a number in a different base. It is normal to not have learned this. We will be working in Binary and Ternary to take advantage of how numbers are represented.

## Number Crunching Time

We are now going to use base 3 notation (ternary) to efficiently summarize what gets deleted in an iteration of building the Cantor Set.

Let's start in the first iteration. The end point  $1/3$  written in ternary is  $0.1_3$  because it is one times the first negative exponent of three. Similarly,  $2/3$  is twice that:  $0.2_3$ .

By removing the middle third in the first iteration, we are removing the numbers between  $1/3$  and  $2/3$ . In ternary, this means we are removing numbers that are bigger than  $0.1_3$  and less than  $0.2_3$ .

Here are some examples of removed numbers:  $0.11_3, 0.111_3, 0.12_3$ .

In other words, we remove any number that has 1 in its first decimal place in base 3<sup>3</sup>.

<sup>3</sup> For now we will ignore the end point  $0.1_3$  and talk about it later.

We now have two lines remaining. The lines are a third of the length of the original line from 0 to 1. This means any cutting that happens will now occur one decimal point down in ternary. That is, cutting a third into thirds creates ninths.

Let's just focus on the first line from 0 to  $0.1_3$ . Anything that is true about that line should be true for the other line from  $0.2_3$  to 1, since they are just translations of each other.

The line from 0 to  $0.1_3$  is cut at a third of a third, or  $0.01_3$  in ternary. It is also cut at the point twice that:  $0.02_3$ . We remove anything between these points. Again, by the same way of thinking, the terms with 1, now in the second decimal place, will be removed.

For the other line from  $0.2_3$  to 1, the same logic applies at the second decimal place. That is, we remove anything between  $0.2 + 0.01_3$  and  $0.2 + 0.02_3$ , since it is just the same line offset by its starting point.

This can continue forever. Try it out yourself! Our subdivision only deletes numbers with 1s in a decimal place.

### *One Last Detail*

We need to talk about end points like  $1/3$  and  $1/9$ . In ternary, these points are  $0.1_3$  and  $0.01_3$ . This can also be written without any ones as  $0.222..._3$  and  $0.0222..._3$ , just like how the number 1 can be represented as  $0.999...$  repeating forever.

Similarly, any other end point with a one is just a combination of numbers we have already looked at. For example, the first end point of the second line is  $(2/3 + 1/9)$ , which is  $0.2_3 + 0.00222..._3$ .

### *Going Back Up a Level*

We now have a convenient way of describing any number in our set: it must consist entirely of 0s and/or 2s in ternary.

With only these numbers, we can match any number we originally had in the following way:

- Take any number between 0 and 1.
- Its binary representation consists of 0s and 1s.
- Literally replace any 1 in the binary representation with a 2.  
For example, the series  $0.01101$  becomes  $0.02202$ .
- Since the Cantor Set contains every possible series made up of 0s and 2s, we can always find a matching number.
- This means there are just as many numbers, just coded differently because they are different values.

This shows that there are as many points in the original line as there are in its sliced up set. Since this does not agree with common sense, the numbers in the Cantor Set and between 0 and 1 are both **uncountable**.

### *Conclusion*

The Cantor set has as many points as the interval it was made from, yet it contains no interval with length!

Thanks for joining me for Day 1.