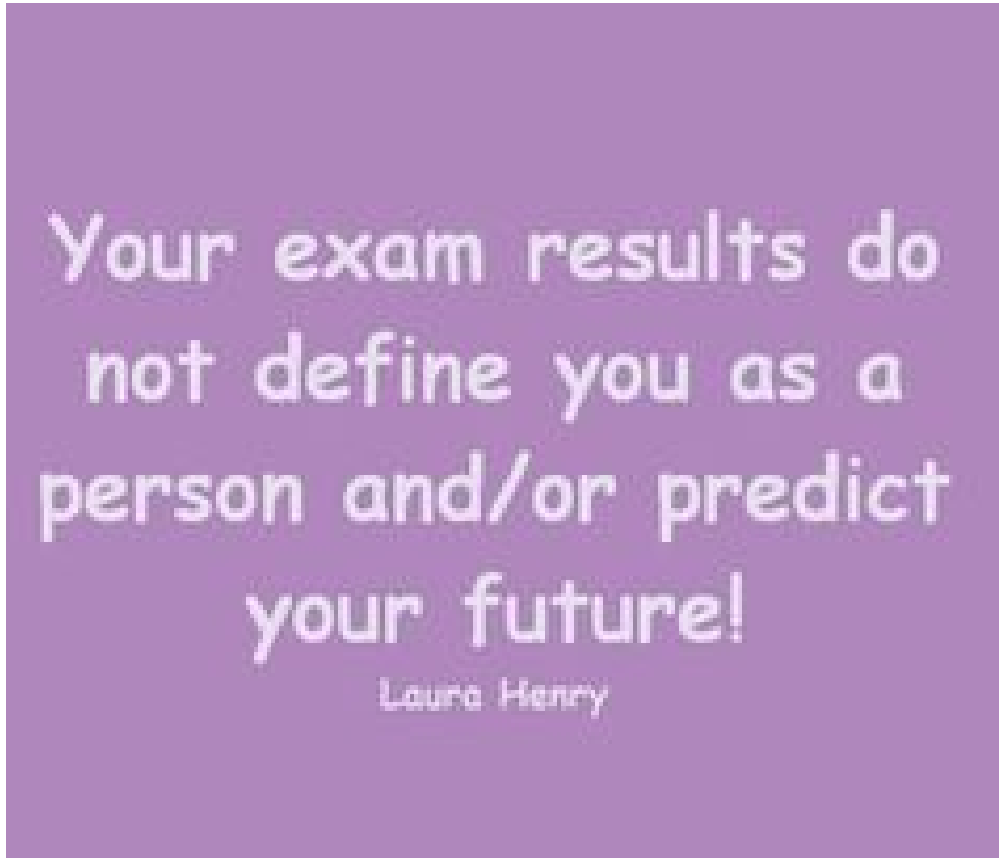
Handouts

Notes on Assignment 14

Assignment 15

No Class on Monday


Your exam results do not define you as a person and/or predict your future!
Laura Henry

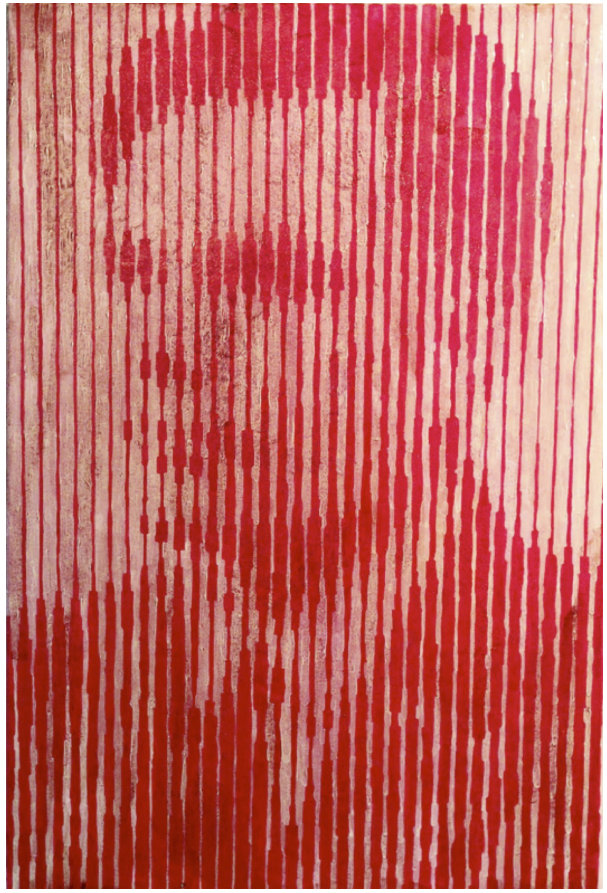
Course Grade Guidelines

	Percentages
Homework and Papers:	30%
Exam 1	20%
Exam 2	20%
Term Project:	20%
Class Participation:	10%

BuzzFeed NEWS

This Painting Made From Gay Mens Blood Make A Powerful Point About Blood Donation Rules

The painting is a protest against UK rules which ban men from donating blood within 12 months of having sex with another man.
The portrait of Alan Turing, the pioneering computer scientist and British war hero, was made from hundreds of vials of blood, all donated by gay men.
The painting was commissioned by campaign group Freedom to Donate, who unveiled it at the Houses of Parliament.
Artist Conor Collins said that almost all the blood is from medical professionals who are unable donate blood, because of their sexuality.



"With the exception of one person, all the blood in the painting is from GPs, surgeons, nurses. They're all medical professionals who, because of who they are, can't donate blood," the artist said.
"One of them was a heart surgeon and you literally trust him to open you up and hold your heart in his hands and yet we apparently, according to the law, don't trust him to donate blood."

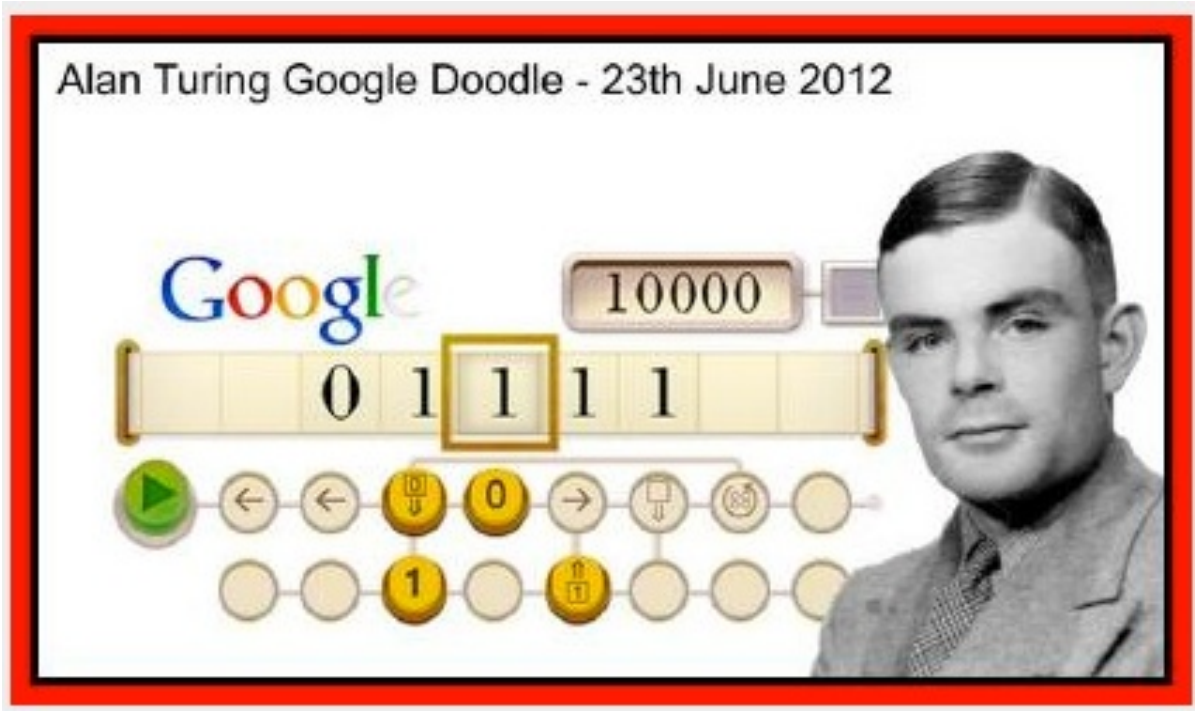
Asked why he chose Turing who was gay for the painting, Collins said there were parallels to his sacrifices and persecution.

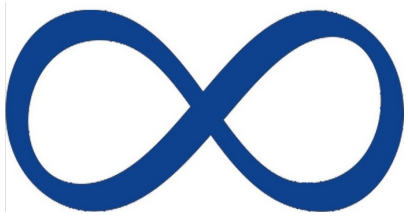
The first thing is that he anonymously through his actions saved the lives potentially of millions and millions of people, he said.

And when you donate you anonymously save the life of someone you've never met and who you'll maybe never meet.

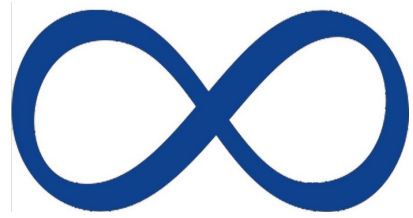
The second thing is that Alan Turing, despite all the actions and great things he did, if he was alive today he wouldn't be allowed to donate blood either.

Turing and Numbers III

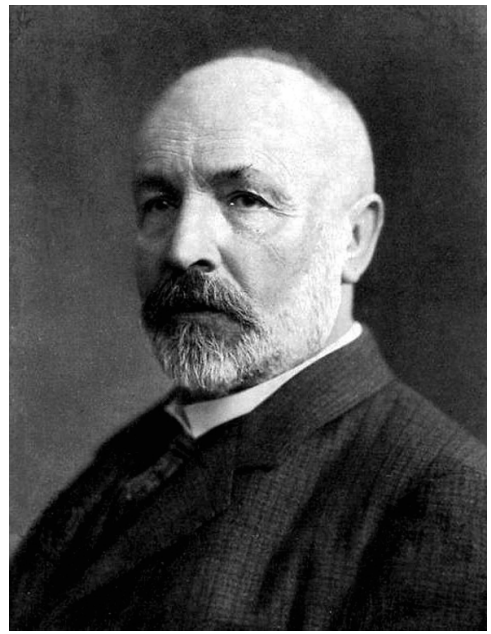




INFINITY



Georg Ferdinand Ludwig Philipp Cantor



Born: March 3, 1845 in St Petersburg, Russia
Died: January 6, 1918 in Halle, Germany
Cantor Biography Link

According to Cantor:

A collection is **infinite** if it can be put into a one-to-one correspondence with one of its proper subsets.

A collection is **finite** if it is not infinite.

Two collections **have the same size** if there is a one-to-one correspondence between the elements of each collection.

Examples of Infinite Sets

Counting Numbers = Positive Integers

Even Positive Integers = 2,4,6,8,10,...

Odd Positive Integers = 1,3,5,7,9,11,?

Primes = 2, 3, 5, 7, 11, 13, 17, 19, 21, 23, 29,?

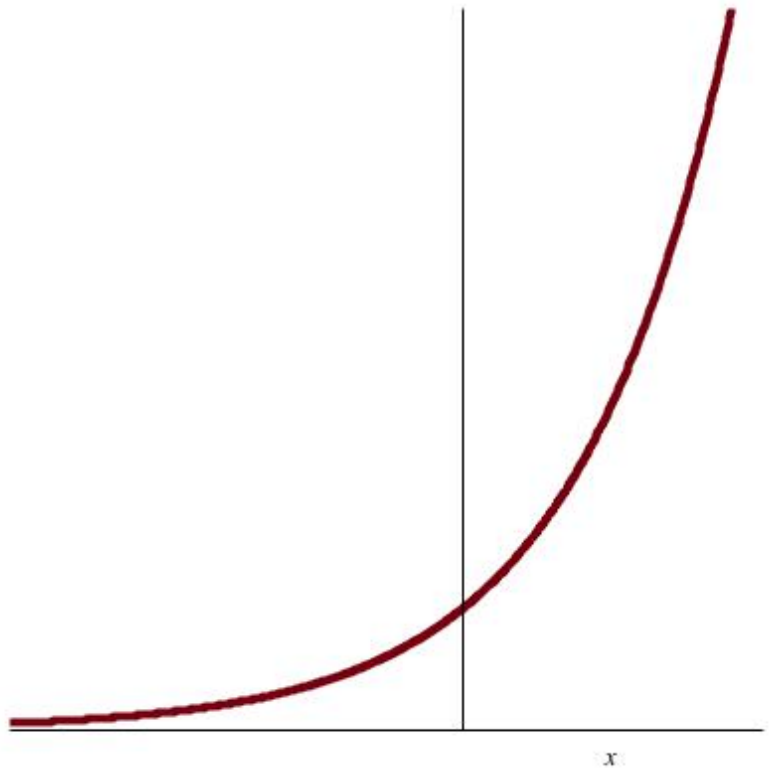
Integers: 0, +1, -1, +2 , -2, +3, -3, . . .

Rational Numbers

Real Numbers

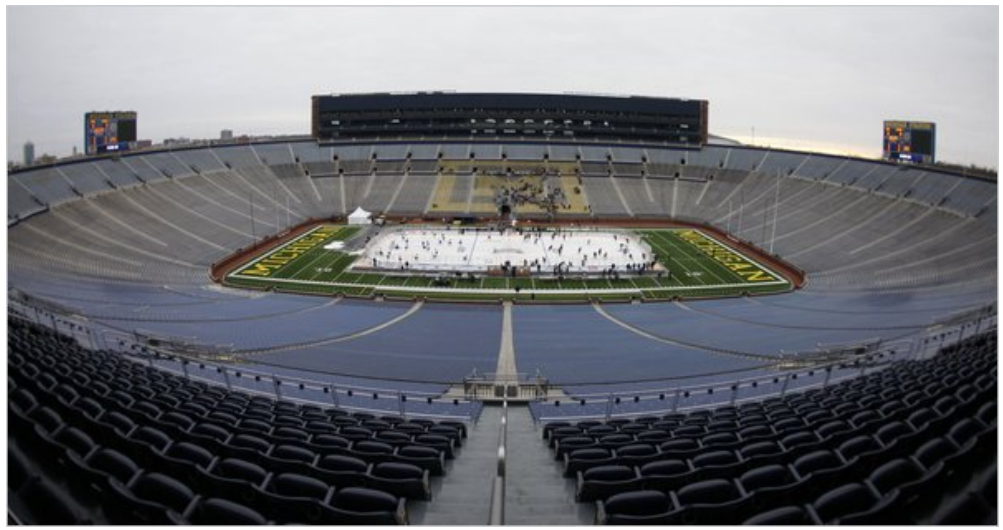
The Real Numbers Form an Infinite Set

A function such as $f(x) = 10^x$ sets up a one-to-one correspondence between the set of all real numbers and the subset of all positive real numbers.



Two collections **have the same size** if there is a one-to-one correspondence between the elements of each collection.

More Fans?



More Seats?


MICHIGAN STADIUM
MICHIGAN
16
HOME SCHEDULE
DEVITA

Two collections **have the same size** if there is a one-to-one correspondence between the elements of each collection.

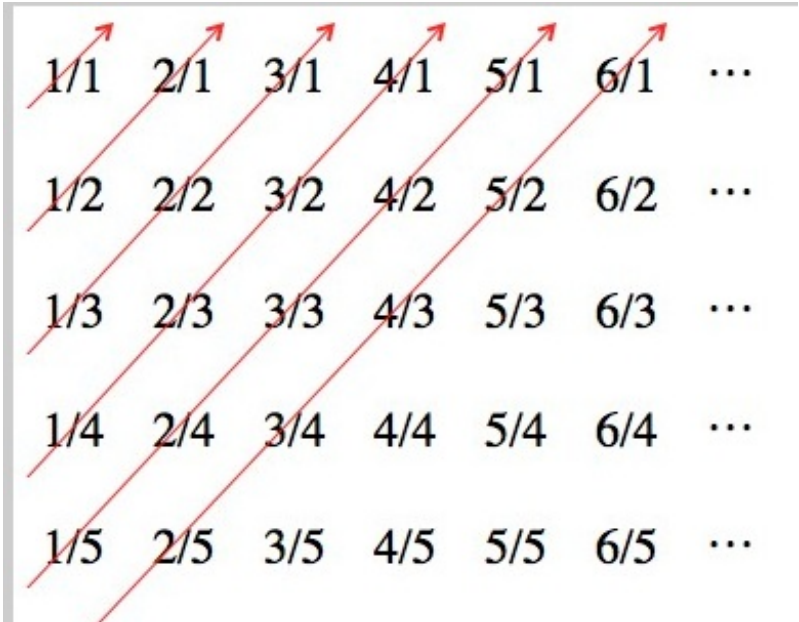
Galileo's Paradox: The Natural Numbers and the Perfect Squares are the same size

1	2	3	4	5	6	7	8	9	. . .
↕	↕	↕	↕	↕	↕	↕	↕	↕	
1	4	9	16	25	36	49	64	81	. . .

Cantor's First Big Surprise

The Set of Rational Numbers and The Set of Integers Have the Same Size

1/1	2/1	3/1	4/1	5/1	6/1	7/1	8/9	9/1	10/1	11/1	. . .
1/2	2/2	3/2	4/2	5/2	6/2	7/2	8/2	9/2	10/2	11/2	. . .
1/3	2/3	3/3	4/3	5/3	6/3	7/3	8/3	9/3	10/3	11/3	. . .
1/4	2/4	3/4	4/4	5/4	6/4	7/4	8/4	9/4	10/4	11/4	. . .
1/5	2/5	3/5	4/5	5/5	6/5	7/5	8/5	9/5	10/5	11/5	. . .
1/6	2/6	3/6	4/6	5/6	6/6	7/6	8/6	9/6	10/6	11/6	. . .
1/7	2/7	3/7	4/7	5/7	6/7	7/7	8/7	9/7	10/7	11/7	. . .
1/8	2/8	3/8	4/8	5/8	6/8	7/8	8/8	9/8	10/8	11/8	. . .
1/9	2/9	3/9	4/9	5/9	6/9	7/9	8/9	9/9	10/9	11/9	. . .
1/10	2/10	3/10	4/10	5/10	6/10	7/10	8/10	9/10	10/10	11/10	. . .
1/11	2/11	3/11	4/11	5/11	6/11	7/11	8/11	9/11	10/11	11/11	. . .



The Set of Real Numbers and The Set of Integers Do NOT Have the Same Size

Cantor's Diagonal Argument

Consider any listing of the real numbers in interval [0,1]

1		.	3	6	7	0	0	2	4	5	9	. . .
2		.	2	5	0	0	0	0	0	0	0	. . .
3		.	9	6	7	8	7	1	2	1	5	. . .
4		.	2	4	8	6	1	3	5	7	6	. . .
5		.										...
6		.										...
7		.										...
8		.										...
. . .		.	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .

Form a number **a** by taking as its *n*th digit

5 if the *n*th digit of number in row *n* is 3

3 if the *n*th digit of number in row *n* is not 3

Claim: a does not appear in the list

David Hilbert

Born: January 23, 1862 in Königsberg, Prussia
(now Kaliningrad, Russia)
Died: February 14, 1943 in Göttingen, Germany

Hilbert Biography Link

"No one shall expel us from the Paradise that Cantor has created."

Turing Machines Algorithms, and Computing

Turing machines (TMs) can compute any function normally considered "computable." In fact, we can define **computable** to mean "computable by a TM".

At the International Congress of Mathematicians (Paris, 1900), David Hilbert identified 23 problems in Mathematics and posed them as a challenge for the coming century

Hilbert's 10th Problem

Devise an *algorithm** that determines whether a given polynomial (with one or more variables) with integer coefficients has an integral root.

* In Hilbert's words: "... a process according to which it can be determined by a finite number of operations."

For example $p(x, yz) = 6x^3yz^2 + 3xy^2 - x^3 - 10$ has four terms over the three variables x, y, z and has a root at $(x, y, z) = (5, 3, 0)$.

The polynomial $p(x) = x^2 - 3x + 2$ over the single variable x has two roots $\{1,2\}$ which can be found by factoring $p(x) = x^2 - 3x + 2 = (x - 1)(x - 2)$ or by using the quadratic formula.

The polynomial $p(x) = x^2 - 2 = (x - \sqrt{2})(x + \sqrt{2})$ over the single variable x has **no** integer roots.

Perhaps Hilbert assumed that an algorithm must exist (similar to the quadratic formula for single variable polynomials). Someone only needed to find it.

It soon became clear that mathematics lacked a clear definition for just what an "**algorithm**" was as well as just what it meant for something to be "**computable function**". At the turn of the century, these two terms (algorithm and computable) were not seen as essentially being the same.

Not having a precise definition for important terms is generally a problem for mathematicians!

During the 1930s, mathematicians were still trying to come to grips with the notion of just what was meant by the term "**computable function**" (as well as the notion of "**algorithm**" itself). Several alternative notions had been proposed each with its own peculiarities and all very different.

Here is a list of the most well known attempts:

λ-calculus	Alonzo Church, Stephen Kleene (1936)
μ-recursive functions	Kurt Gödel (1933)
combinatory logic	Haskell Curry (1930)
Post systems	Emil Post (1928)
Turing machines	Alan Turing (1936)

All these systems had a notion of "computation" in one form or another. They deal with various types of data, however. For example, Turing machines manipulate strings over a finite alphabet, μ-recursive functions manipulate the natural numbers, the lambda calculus manipulates "lambda" terms, and combinatory logic manipulates strings of "combinatory symbols".

It is easy to encode just about anything as strings in $\{0,1\}$, after all, we do this all the time when using modern day computers. Surprisingly, with suitable encodings as strings, it turns out that all the above formalisms can simulate each other. So despite their superficial differences, they are all equivalent models of "computation".

Of the classical systems listed above, the one that most closely resembles modern computers is the Turing machine.

The Church – Turing Thesis

When it was discovered that all these researchers were essentially talking about the same thing, Alonzo Church went out on a limb and declared that the intuitive notion of "computable by some algorithm", which mathematicians had sought to capture formally for some time, is captured precisely by Turing machines.

This assertion is known as **Church's Thesis** or **The Church – Turing Thesis**. It is not a theorem but rather a declaration that the formalism exactly captures our intuition of what it means to "compute" something

The Church – Turing Thesis

Intuitive notion of what it means to be computable via an algorithm

$\cong$ (is equivalent to) computable via a Turing machine

or

intuitive notion of algorithm $\cong$ TM

or

Any computation that can be carried out by mechanical means can be performed by some Turing machine.

Also, since there is a "**Universal Turing Machine**" that can simulate all other TMs, one can take the liberty of viewing a TM as being an algorithm and a universal TM as being a computer that "executes" algorithms (all other TMs).

Algorithm $\cong$ TM $\cong$ computer program (say in Java or Python)

computer $\cong$ universal TM