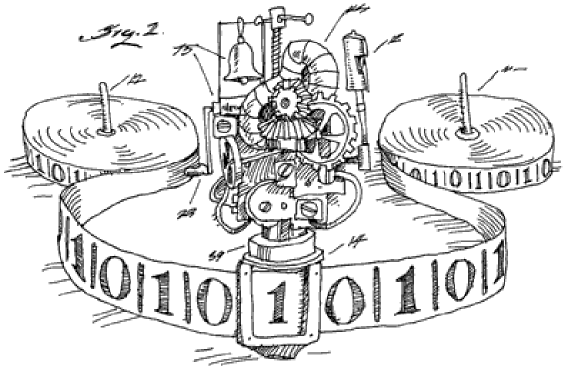
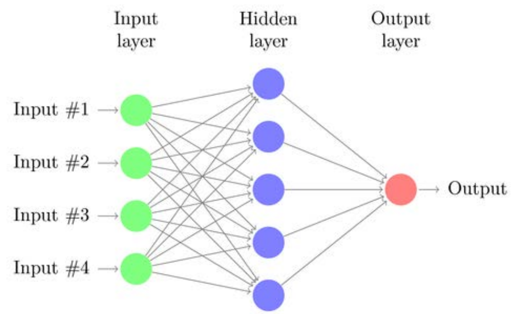
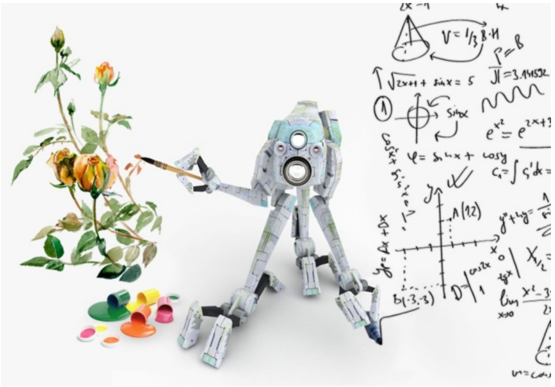
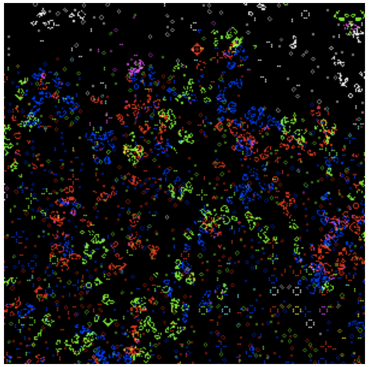


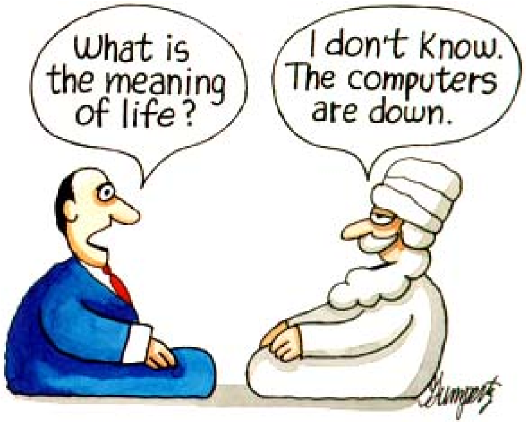


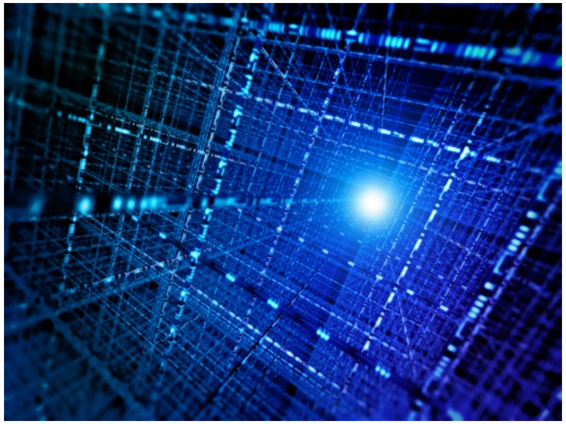


Limits of Computation

Antonina Kolokolova



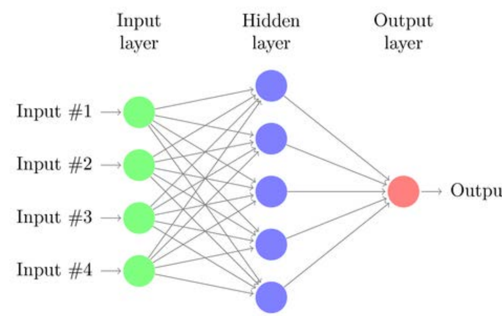




- What is computation?

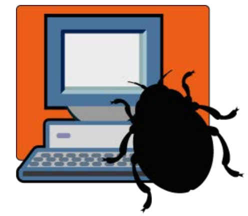
- What is information?

- What is learning?



- Are there any limits of our ability to solve problems?

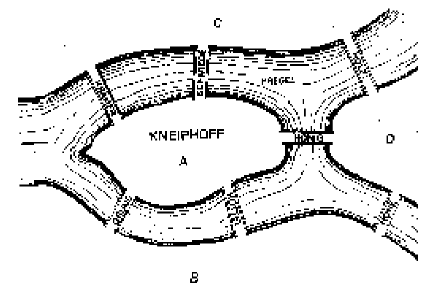
Theoretical Computer Science



- Is there a perfect antivirus?

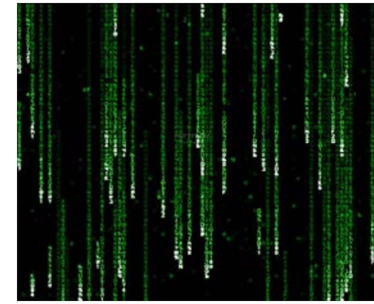
- Can computers be creative?

- Why some problems are easier than others?



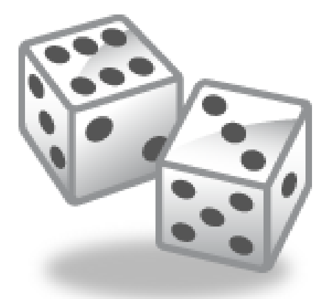
- Is it possible to have secure information and communication?

What is information?

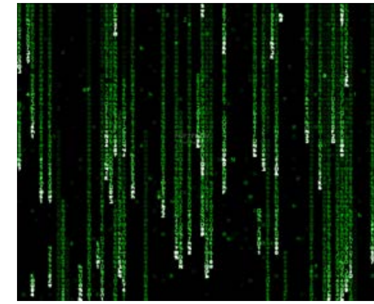


- Does string 11111111111 contain more information than the string 10010110100?

- Do you learn more from a coin toss of a fair coin or a roll of dice?



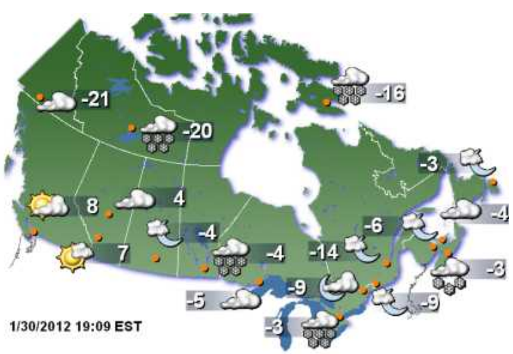
What is information?

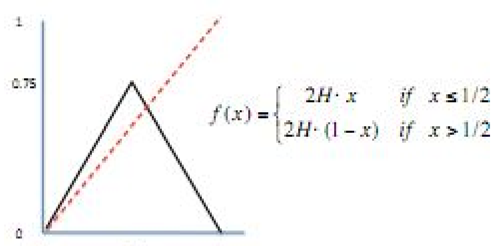


The less you can predict an outcome

The more you learn from it:

The more information you get.





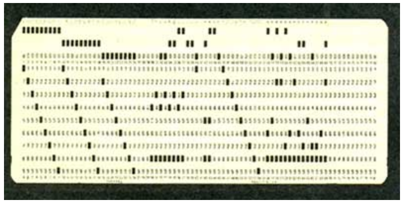
The science of information

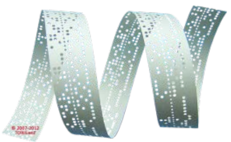
- In many languages the word for “Computer Science” is derived from the word for information
  - French: Informatique
  - Spanish: Informática
  - German: Informatik
  - Russian: Информатика

- The information comes in and we process it.
- So do computers. So do living cells, etc, etc.

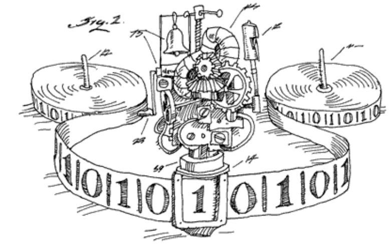
On the other side of iron curtain

- In Soviet Union, in particular in Ukraine, PCs were not around till 1990^{th}
- First photo: “MIR” computer (from 1969). Developed in Kiev by Glushkov and his group.
- Were still in use in 1980s.
- Programmable calculators for personal use





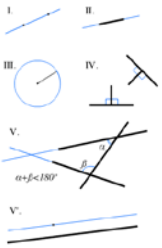
What is computation?



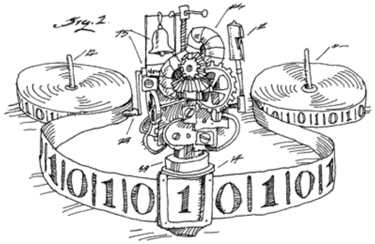
- We process information by doing a "computation on it". Changing it from one representation to another.

- But what is computation?
  - What does your smartphone compute when you are playing Pokemon Go?
  - How does DNA "compute"?

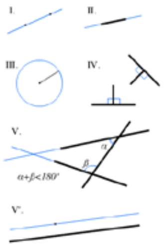
- Is there a limit to what can be computed?



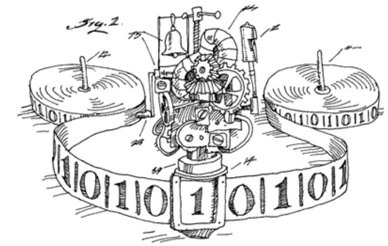
Limits of computation



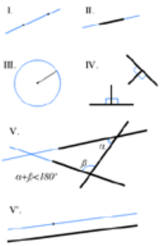
- In 1900, at the International Congress of Mathematicians in Paris, David Hilbert posed a list of 23 problems. Problem 2 asked to prove that mathematics contains no self-contradictions.
- In 1920, Hilbert extended it to what is now known as “Hilbert’s program”



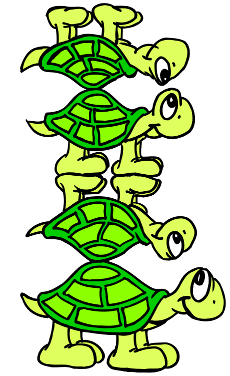
Hilbert’s program



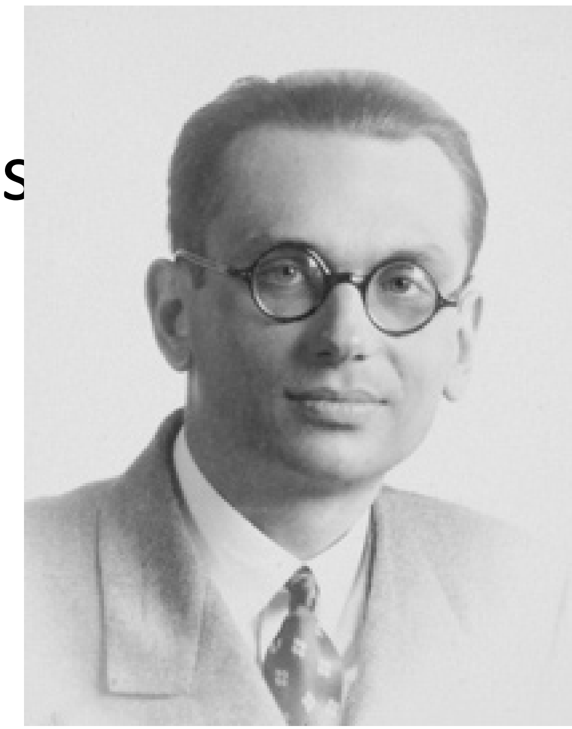
- Express all mathematics in a precise way
- Allowing a formal proof of all true statements
- With a proof, inside mathematics, that there is no self-contradiction
- And a procedure (an algorithm) for deciding, for any given mathematical statement, whether it is true or false.

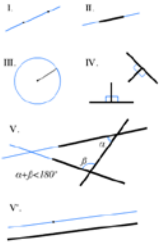


Gödel Incompleteness Theorem

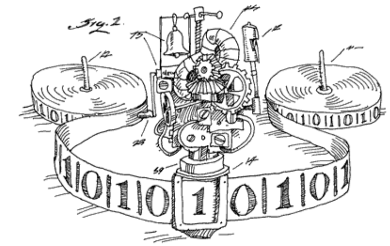


- If mathematics is not self-contradictory...
- Then there are true statements that can't be proven!
- Such as "I am not provable"
- A paradox!





Church and Turing:

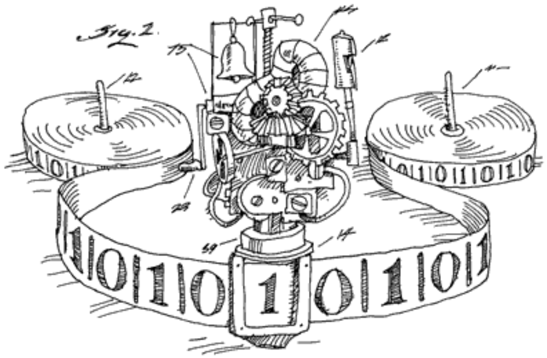


- Moreover,
- there is no procedure
- to decide if a given statement is true or false!
- And to decide many other things...

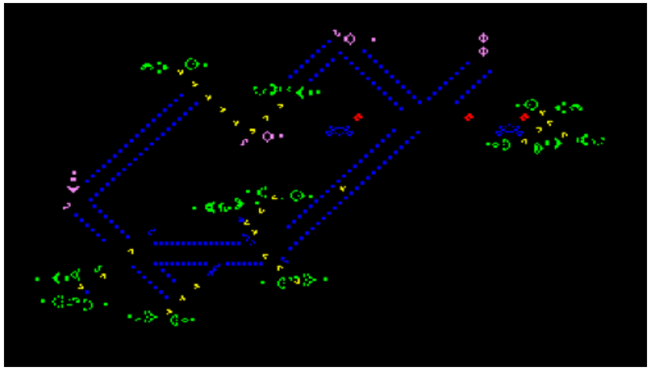
- But what do we mean by a “procedure”?

Models of computation

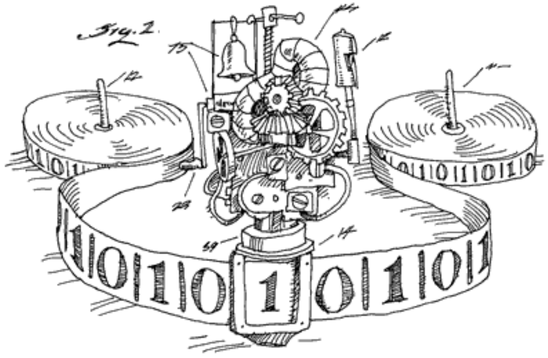
- Let me show you two models of computation.
- The first one is the Turing machine
  - Our modern-day computers are based on this model



- The second is the Game of Life
  - Looks nothing like a computer, and yet has the same power.

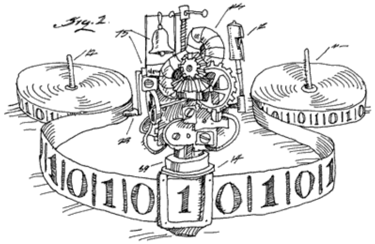


Turing machine

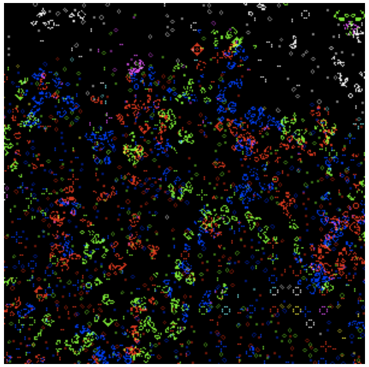


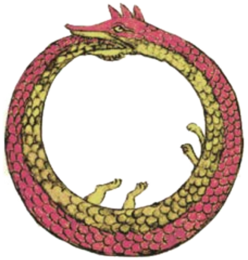
- A Turing machine has an (unlimited) memory, visualized as a tape
- Or a stack of paper
- And takes very simple instructions:
  - Read a symbol
  - Write a symbol
  - Move one step left or right on the tape
  - Change internal state.

Church-Turing thesis

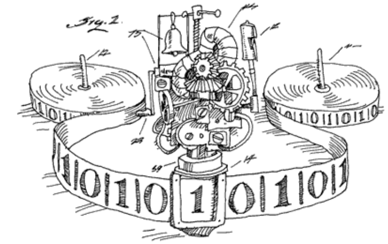


***Everything we can call "computable" is computable by a Turing machine*.**

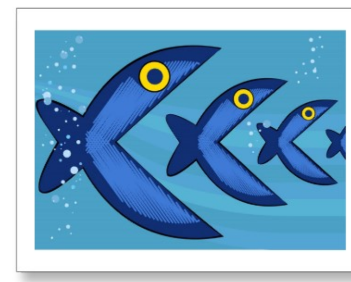




“Will this ever stop?”



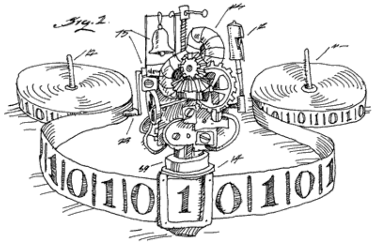
- A code for a Turing machine program is a string.
- Any string can be an input to a program.
- Imagine there is a machine that always does the opposite…
  – From the machine which code is its input
    - On a string encoding it



- What will it do on its own code?
  – Yes?... No?... Yes?… No?... Paradox!

- So no such machine can exist… Some problems are unsolvable, with self-reference to blame.

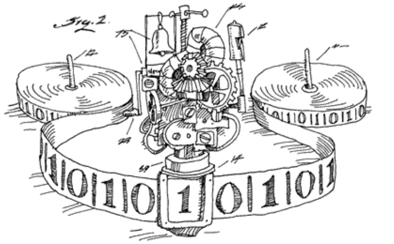
Complexity of computation



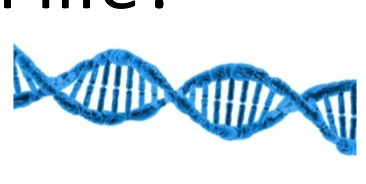
- Would you still consider a problem really solvable if it takes very long time?
  - Say 10^n steps on an n-symbol string?
  - At a billion (10^9) steps per second (~1GHz)?
  - To process a string of length 100…
  - will take $10^{100}/10^9$ seconds, or ~$3x10^{72}$ centuries.

  - Age of the universe: about $1.38x10^{10}$ years.
  - Atoms in the observable universe: 10^{78}-10^{82}.

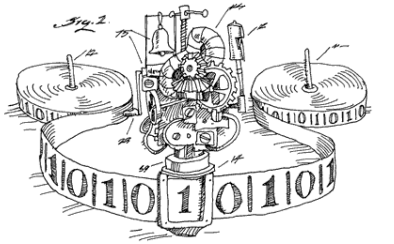
Complexity of computation



- What strings do we work with in real life?
  - A DNA string has 3.2×10^9 base pairs
  - A secure key in crypto: 128-256 bits
  - Number of Walmart transactions per day: 10^6.
  - URLs searched by Google in 2012: $3 \text{x} 10^{12}$.



Efficient computation



- What can be computed in our universe?
  - We could only work with very short strings…
    - But we want to work with our DNA string!

  - We can try being efficient in solving problems.
    - What does it mean to be efficient?
    - And what kinds of problems can be solved efficiently?

A million-dollar question!

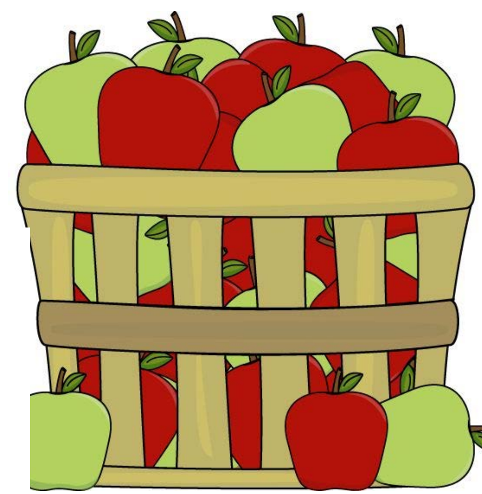
The million dollar question

- In Russian, called “perebor” problem.
  - “perebor” translates as “exhaustive search”.
  - Question: is it always possible to avoid looking through nearly all potential solutions to find an answer?
    - Combinations of letters to guess your password?
    - Combinations of numbers to solve an equation?

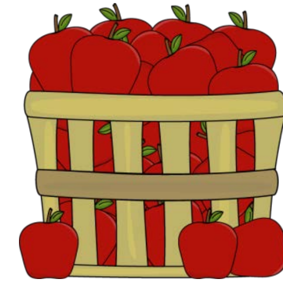
Are there situations when exhaustive search is unavoidable?

The million dollar question

- Suppose you have a basket of apples.
- Can you check that all apples are good without looking at (essentially) every single one?
  – Is there a way that would work for every possible basket of apples?
- **Smell test?**



The million dollar question



- In English, most known as "P vs. NP" problem
  - P stands for "polynomial time computable".
  - NP is "polynomial time checkable"
    - non-deterministic polynomial-time computable
    - (fancy word for "guess and check")

  - **Question: is everything efficiently checkable also efficiently computable?**
    - In particular, is there a "smell test" for every problem that has easy-to-check solutions?

Colouring maps

- How many colours needed so that neighbouring countries do not get the same colour?
- For a picture like that – no more than 4.
- (A theorem famous for being proven with a help of a computer)
- Can it be done with 3?

Colouring maps

- Can it be done with 3 colours?
- How do we find out?
  - Look at neighbours of Austria. There are 7 of them… 3 colours not enough.
- In general, nobody knows a good way!

Question

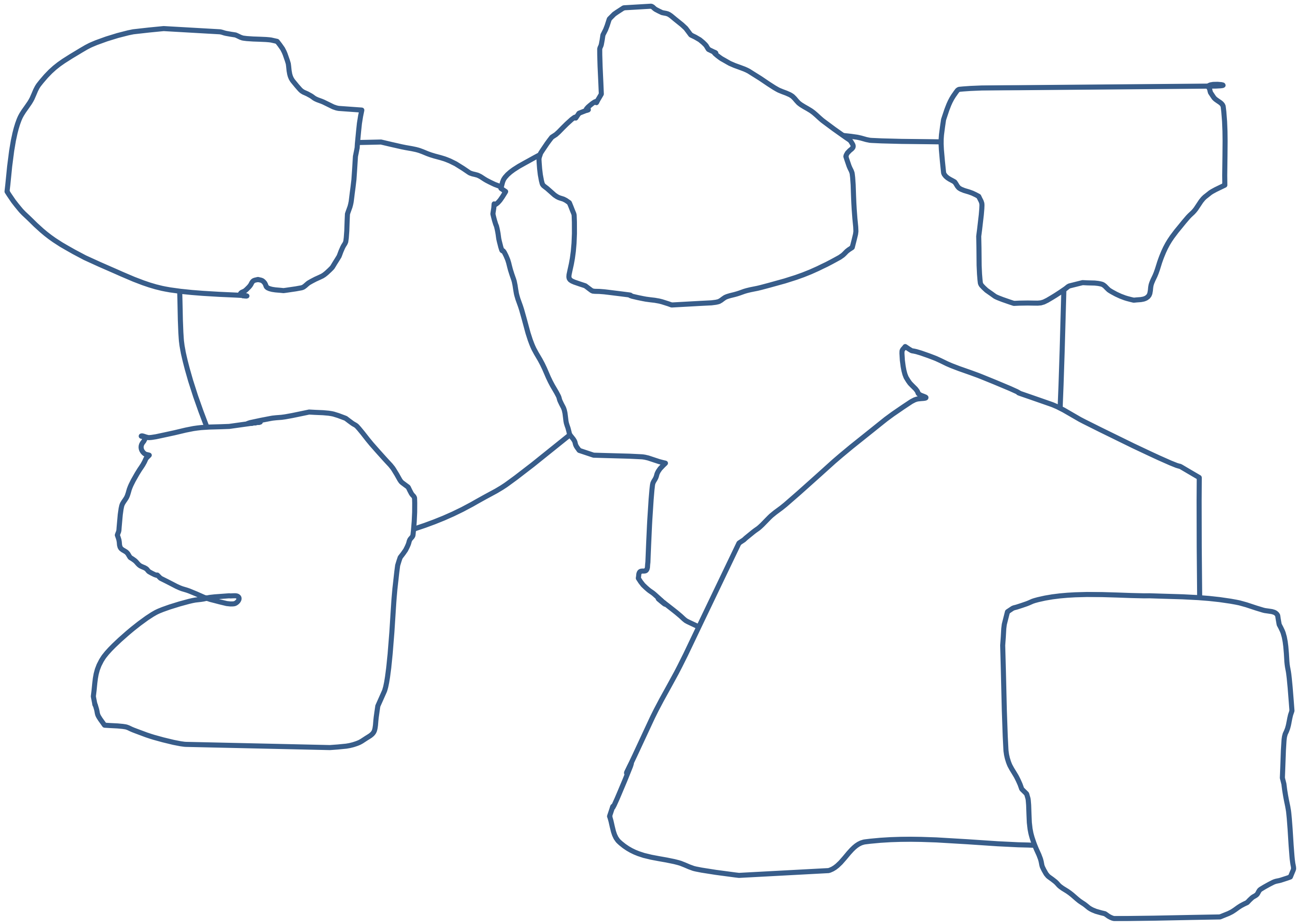
- Can this map be coloured with three colours?

Question

- Can this map be coloured with three colours?

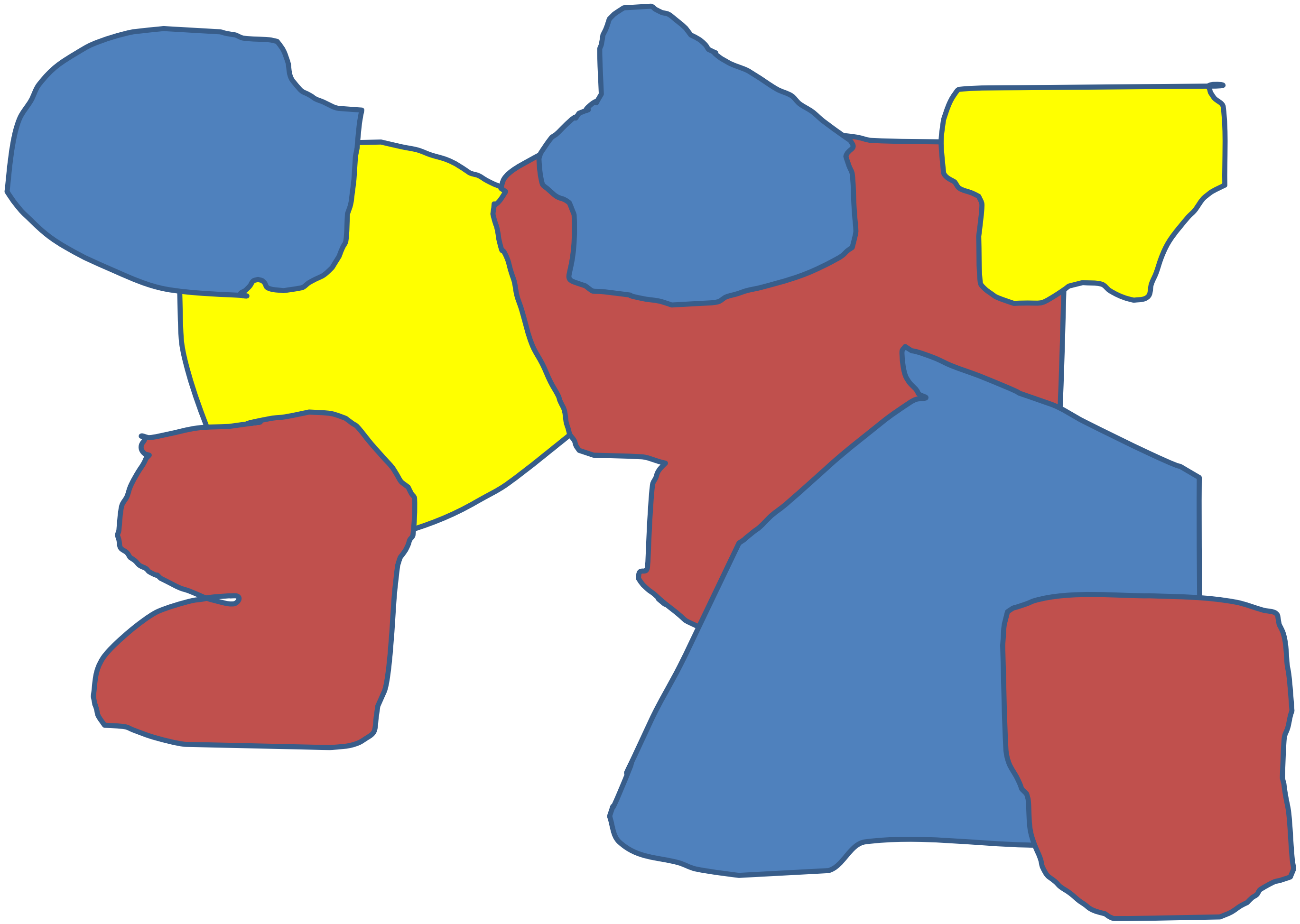
Question

- Can this map be coloured with only 3 colours?



Question

- Can this map be coloured with only 3 colours?



Question

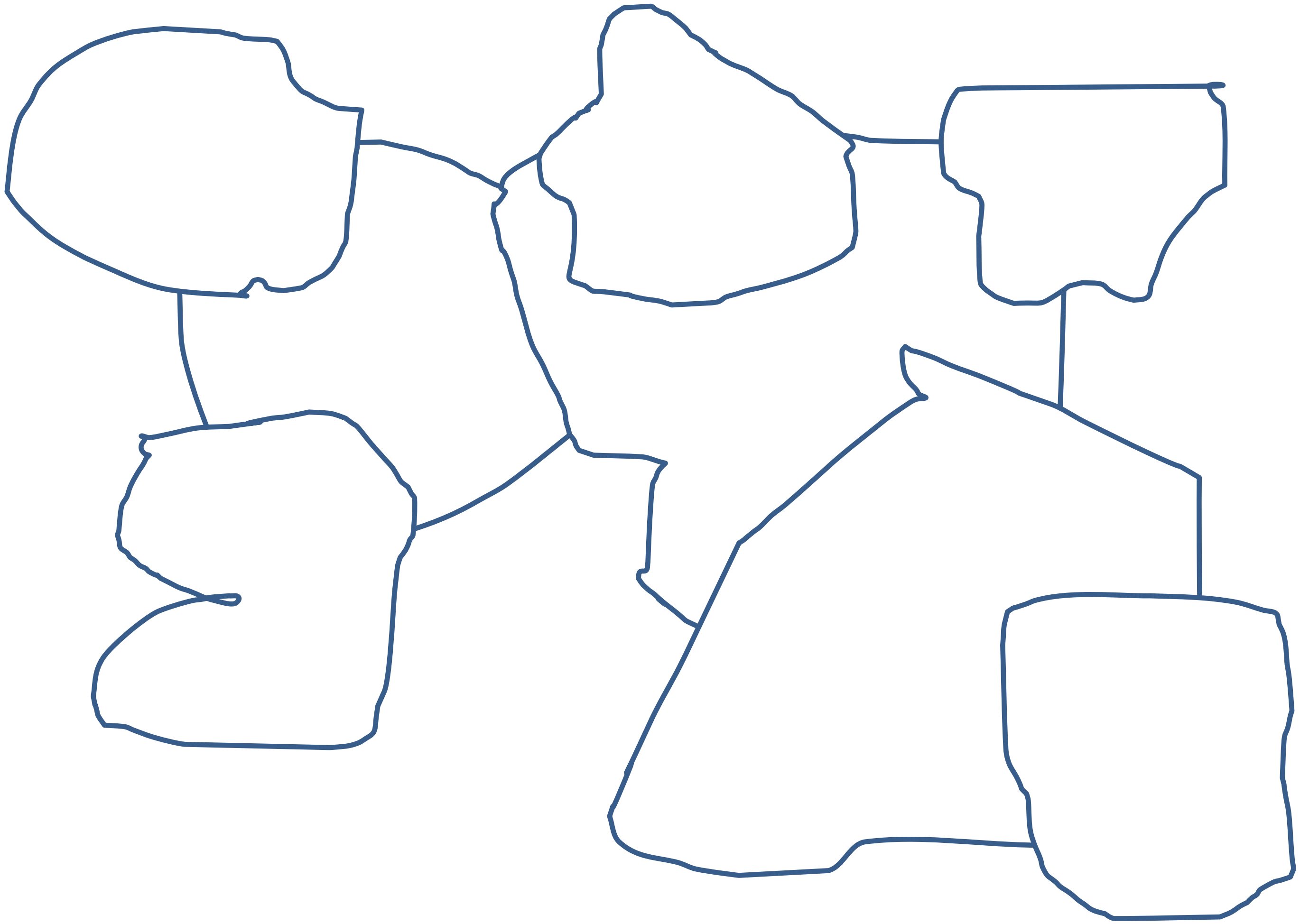
- Can this map be coloured with two colours?

Question

- Can this map be coloured with two colours?
- No…
- Western Australia, Northern territory and South Australia should all be different colours.

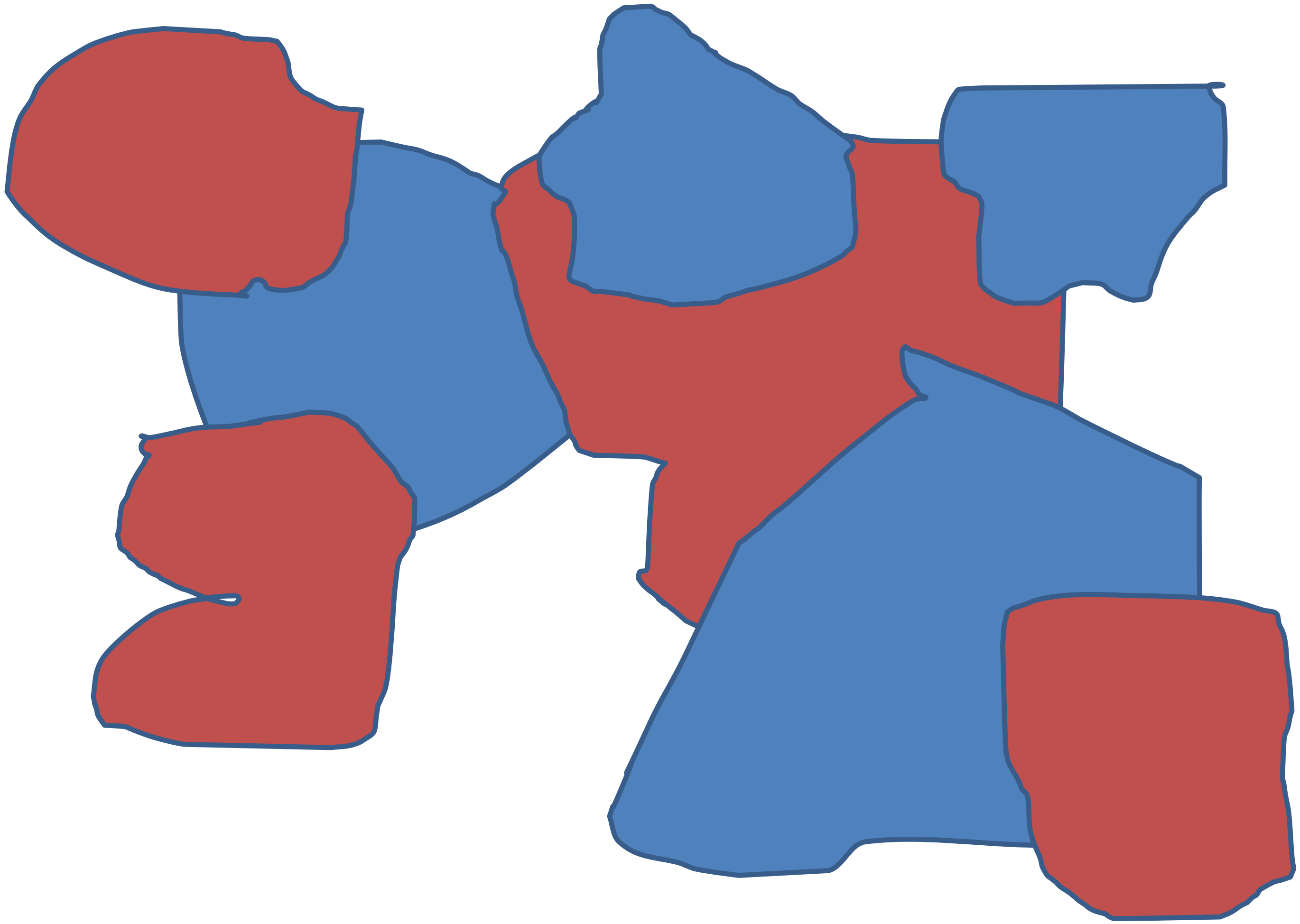
Question

- Can this map be coloured with only 2 colours?



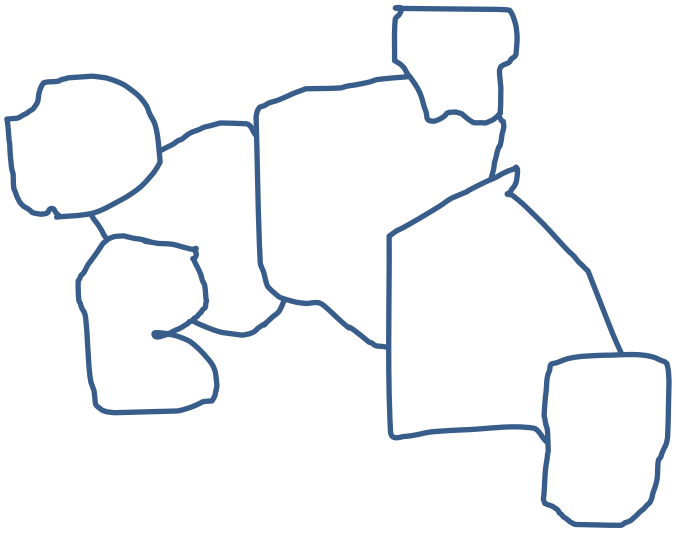
Question

- Can this map be coloured with only 2 colours?



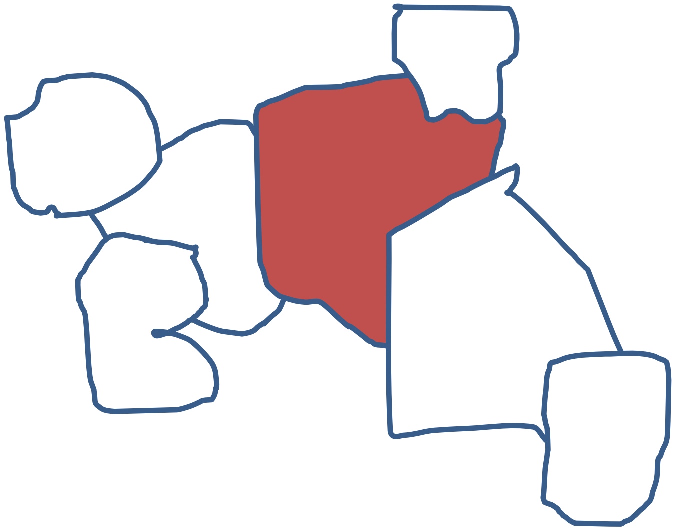
Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour



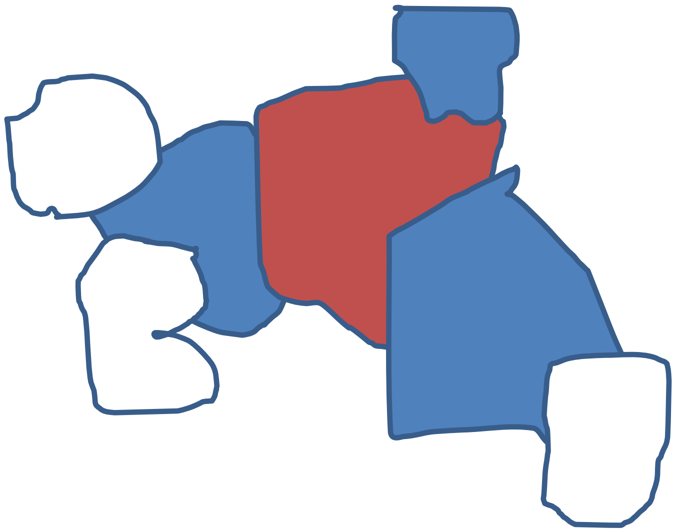
Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour



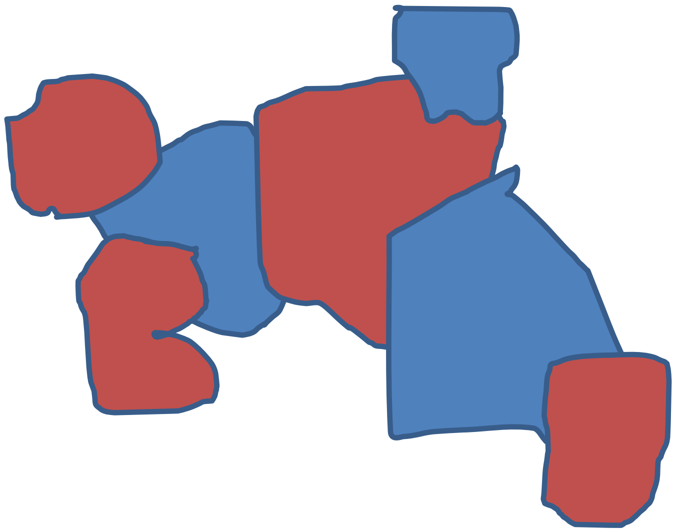
Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour



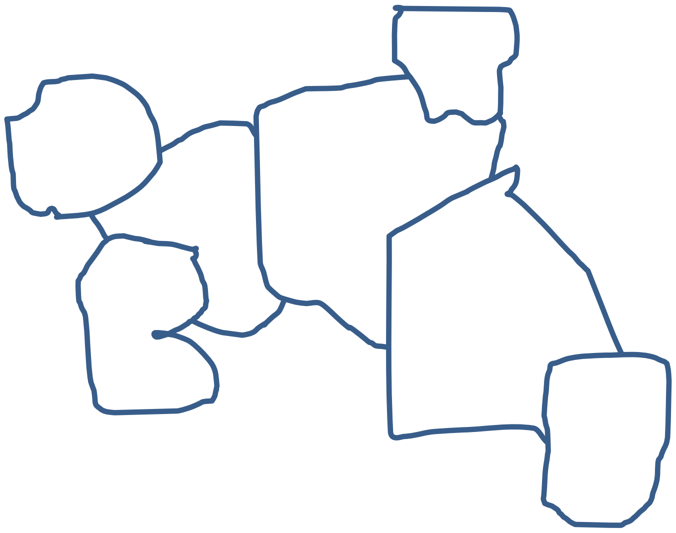
Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour



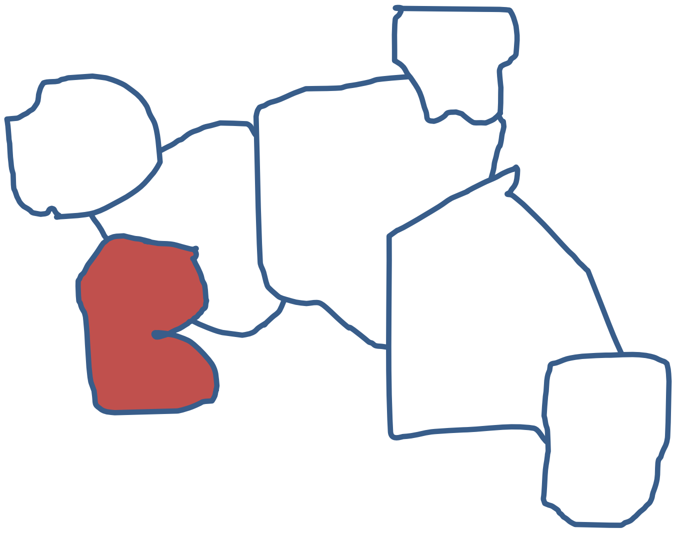
Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour



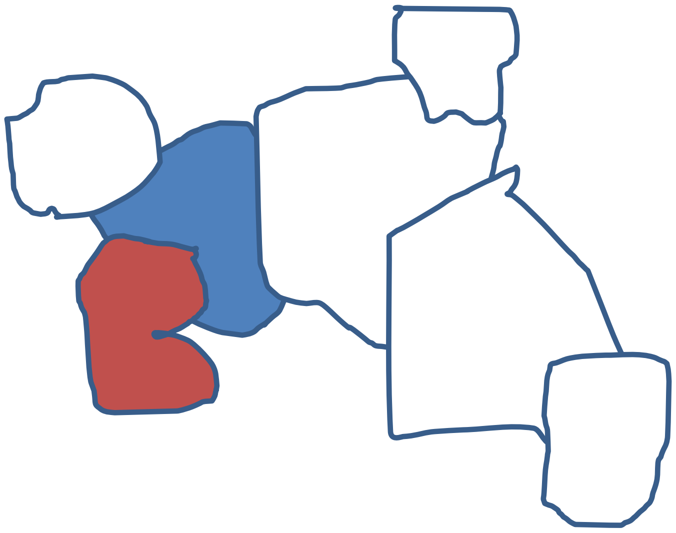
Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour



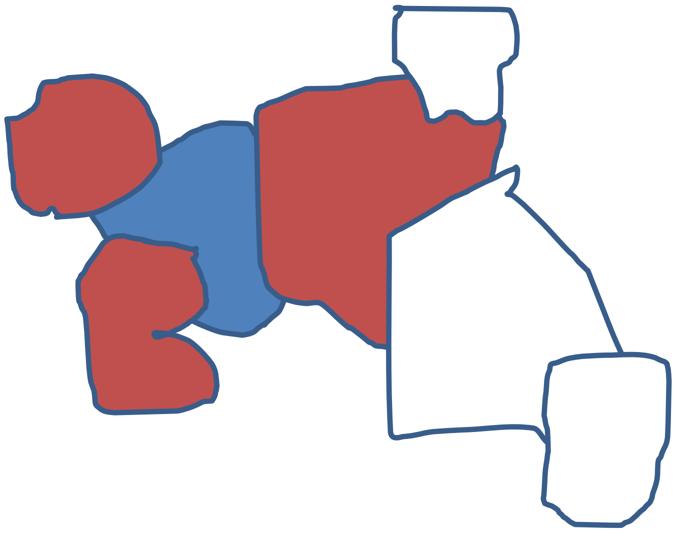
Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour



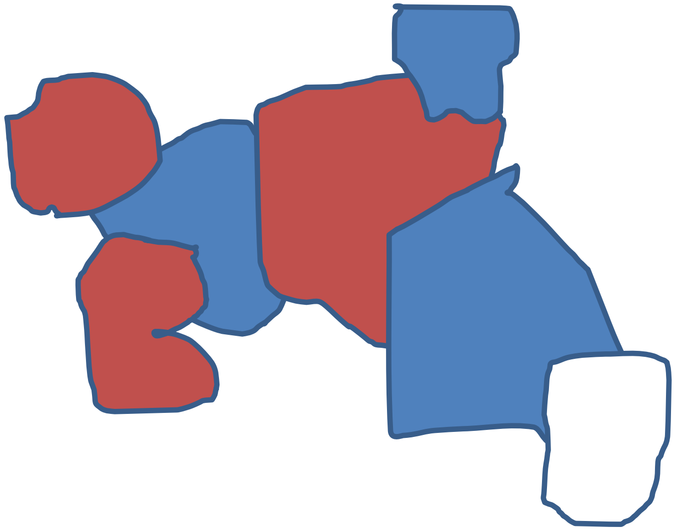
Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour



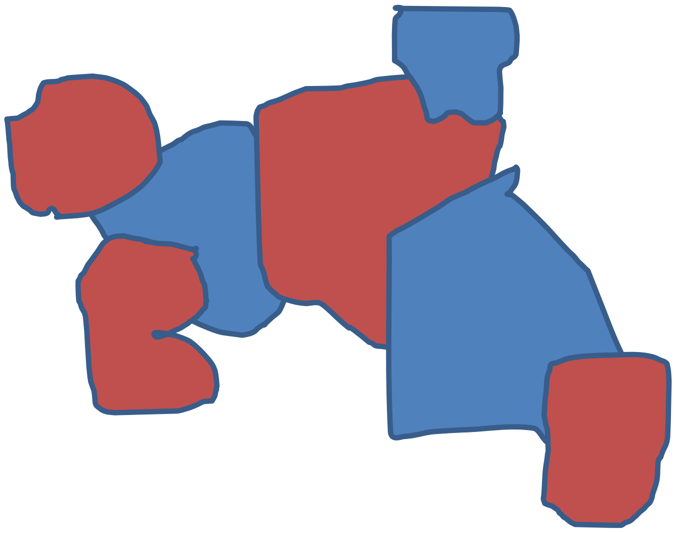
Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour



Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour

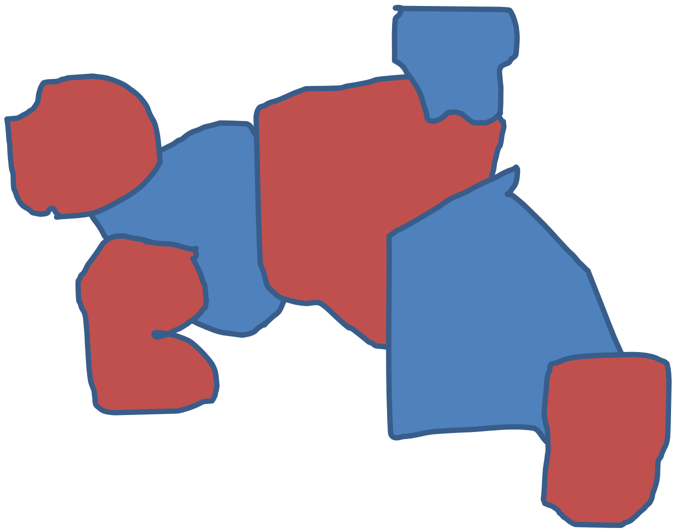


Colouring with 2 colours

- How do you check if a map is colourable with 2 colours?
  - Start anywhere.
  - Colour a region red
  - Colour its neighbours blue
  - Colour their neigbours red again…
  - Continue until either done, or found a region would border one of the same colour

Colouring with 2 colours

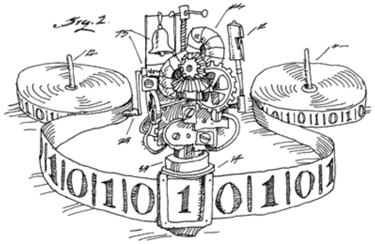
- So checking if a map can be coloured with 2 colours is an easier problem than with three!
- And any map can be coloured with 4 colours.



Colouring maps

- If somebody gives you a coloured map, easy to check.
  - Check that there are 3 colours overall
  - Check that each country is different from its neighbours.
  - Done!
- Finding a colouring seems much harder…

Polynomial-time computable



- Efficiently solvable:
  - On an input string of length n
  - Produce a solution roughly in time at most
    - n, or n^2, or n^3, or… n^{const}.
  - So a DNA string can be processed in about 3.2×10^9 steps. At 1GHz, it is 3.2 seconds.
- Concept dates back to 1960s, Jack Edmonds, and also Alan Cobham.
  - Edmonds arguing why his "blossom algorithm" is better than what was known before.
- Checking that a given map is 3-colourable is polynomial time computable.
  - So is figuring out if a given map is 2-colourable.

NP-completeness

- Is it possible to eliminate exaustive search?
- **NP-completeness:** enough to answer for the problem of map colouring!
- A map is like a basket of apples.

  A map is colourable with 3 colours

  =

  There is a bad apple in the basket.

Stephen Cook

Leonid Levin

- Can “disguise” every problem that has efficiently checkable solutions as map colouring!
  - The concept of **NP-completeness** was invented by Stephen Cook (and independently Leonid Levin) in 1971
  - Made its way into popular culture, often as a synonym to “hard”… though we do not know for sure!

Maps vs. teams

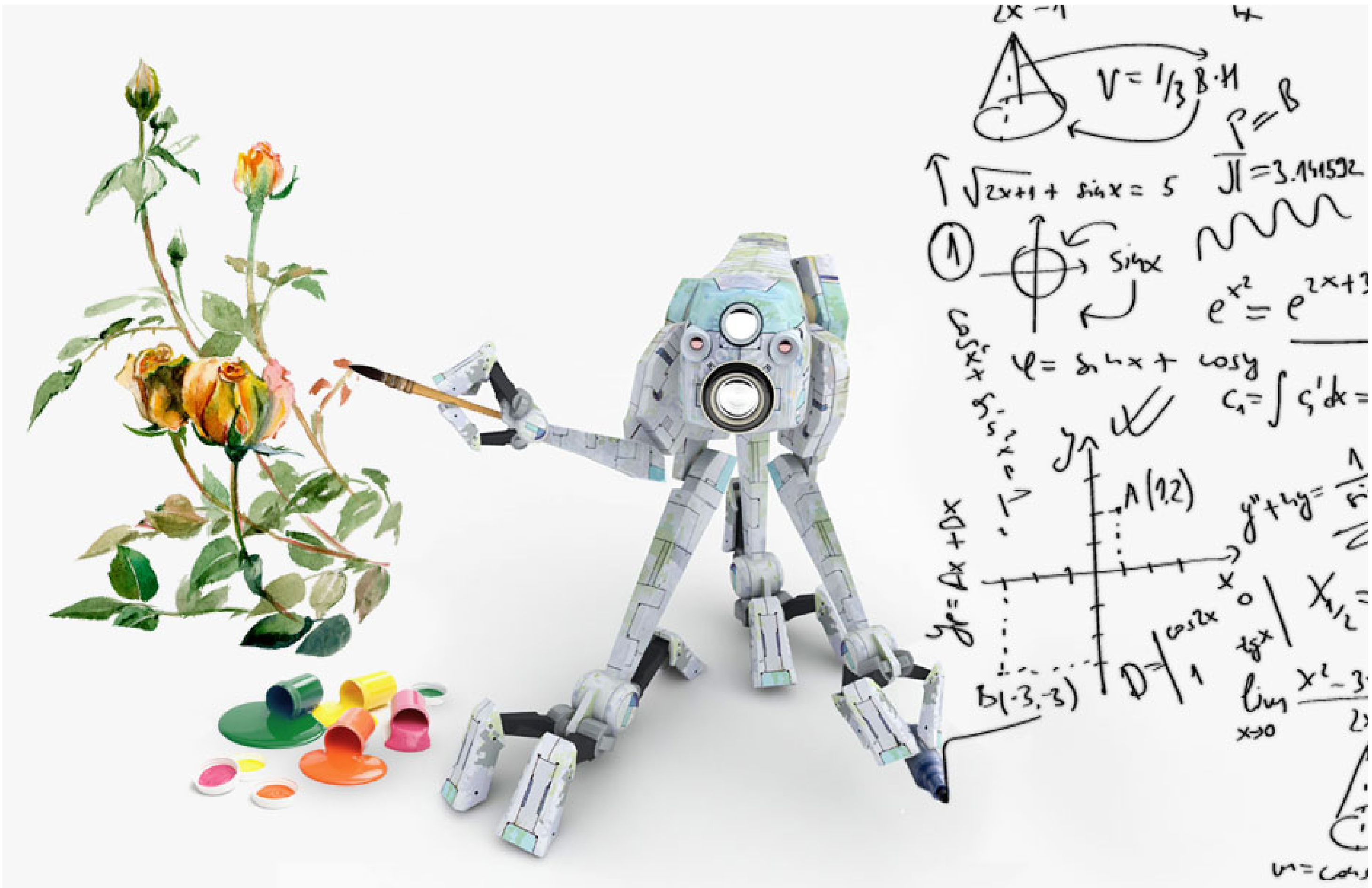
- Colouring a map with 3 colours is the same problem as splitting into three teams – in disguise.
- Call them TeamRed, TeamGreen and TeamYellow.
- Require that neighbours cannot be on the same team.
- If can split into three teams, colour countries by their team's colour.
- So if can split into teams efficiently, can colour maps just as efficiently.
- So splitting into two teams seems easier than into three!

P vs. NP

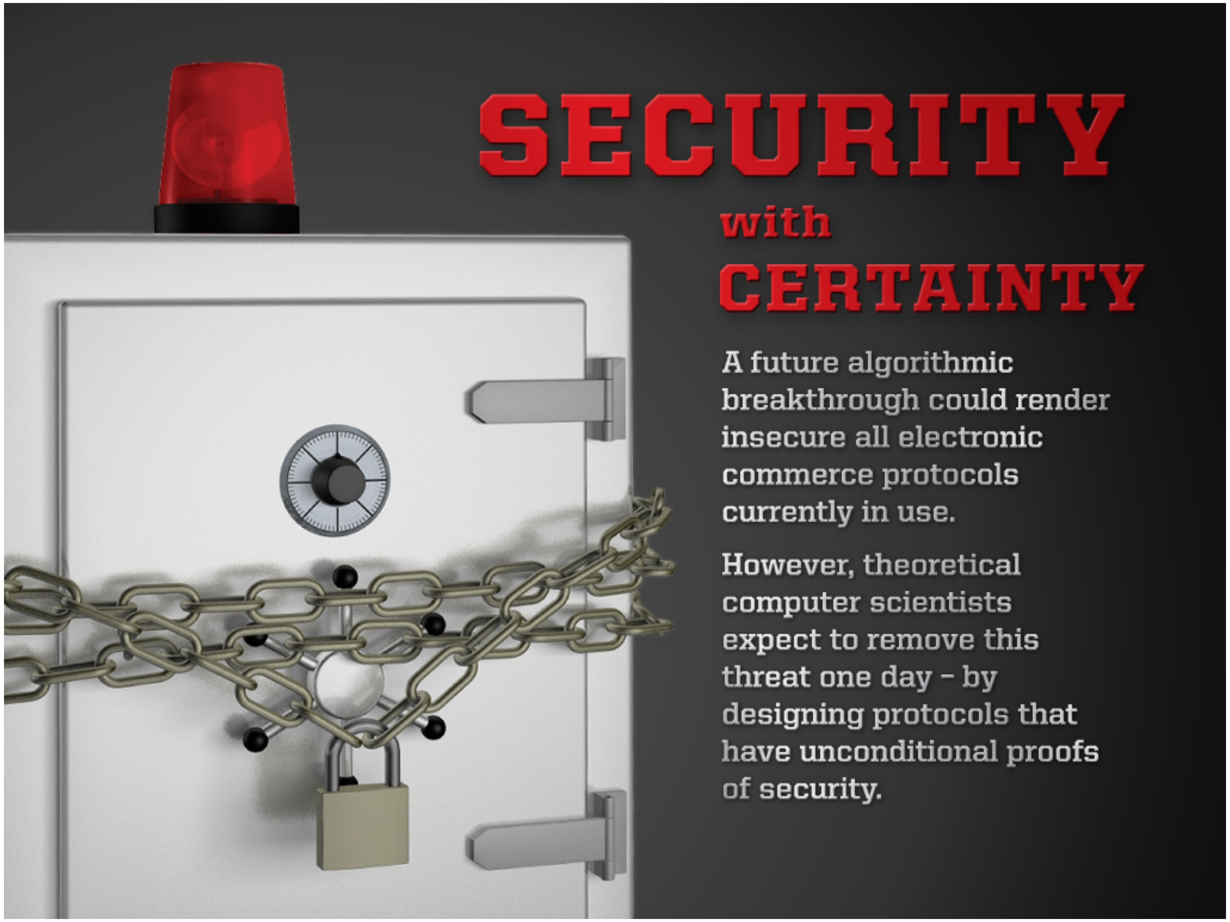
- If somebody finds a way to solve 3-colouring efficiently, then we will live in a very different world, where
  - Creativity and problem-solving are automated.
  - Not much security left on the internet.
  - Every theorem has a short proof…

- So most scientists believe that solving 3-coloring is impossible, but nobody so far can prove it.

If P=NP… creativity is automated

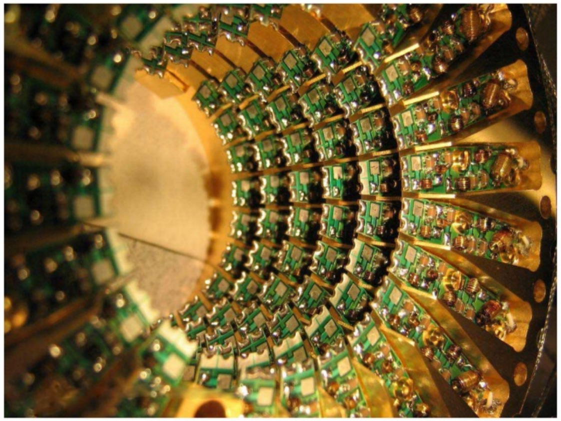


If P=NP… there is no security

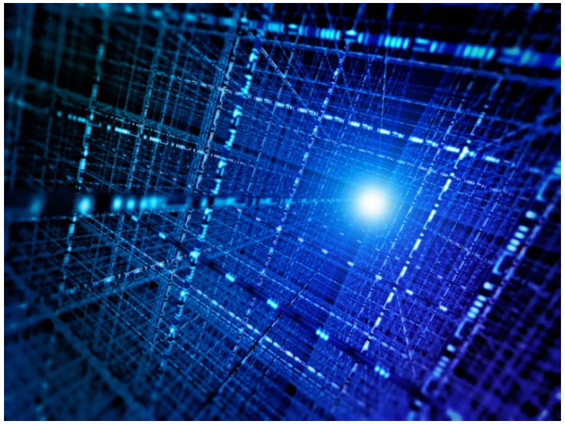


Quantum computers

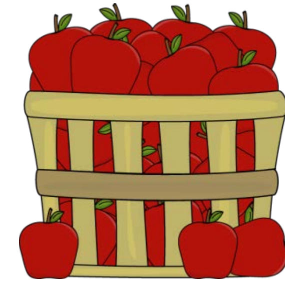
- Can quantum computers colour maps efficiently?
- We don't know… but don't think so.
- Although they can factor numbers, which we do not know how to do on a usual computer fast.
- A real scalable quantum computer would require changing much of security on the internet.
  - RSA cryptosystem assumes factoring is hard.



Adiabatic Quantum Computer Component Array

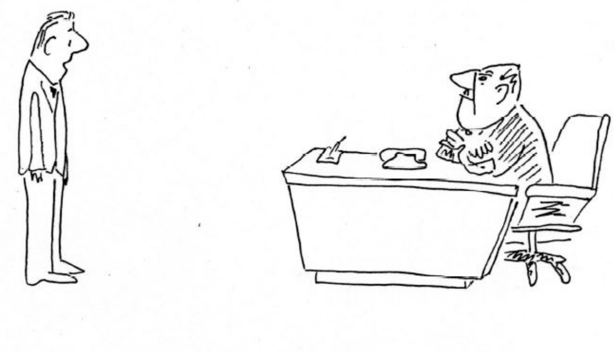


Exhaustive search beyond P vs. NP

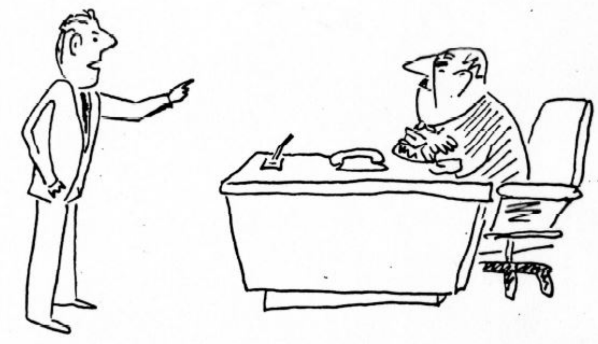


- Consider your friends on Facebook.
- Now look at the groups that they form
  - Belonging to the same Facebook group
  - Following the same person
  - Etc…
    - There can be many more groups than people.
- How hard is it to check if there are two groups that do not have any person in common?
  - Easy! Just compare any two groups.
    - Time to compare two groups times square of the number of groups.
  - Wait, but that's exhaustive search!... Can we do faster?..
    - If there are *lots* of groups, too slow in practice.
  - If so, can eliminate exhaustive search from map colouring…
    - Not quite solving P vs. NP, but getting there!

P vs. NP (David Johnson's cartoons)



"I CAN'T SOLVE IT - I GUESS I'M JUST TOO DUMB."



"I CAN'T SOLVE IT — BECAUSE NO SOLUTION EXISTS!"

"I CAN'T SOLVE IT - BUT NEITHER CAN ALL THESE FAMOUS PEOPLE!"

WE MAY NOT BE ABLE TO SOLVE IT...
BUT WE SURE CAN GET CLOSE!