

0. Scratch

First of all, what is computer science?

David Melan defines it as problem solving. However, problem solving is part of everything. An input and output cycle is basically every problem solving “algorithm” your brain thinks of every time a problem occurs.

In the world of computers, you need to need to decide how to define these inputs and outputs. Computers only know what language so to speak.

Binary

Computers generally use binary digits - 0's and 1's. So, why that particular pattern? At the end of the day, representing one thing or the other is really easy for computers. This is known as a boolean. In our human world, if we think of the number 157 for example, we think of it instinctively of course as the number 157, but looking further, it is actually $100 \times 1 + 10 \times 5 + 7 \times 1$. We assign each place of the digits a value; the ones place, tens, hundreds, thousands, etc. These are all in the power of ten and are called “decimals”. Coincidentally, this is quite similar for computers, except it is in binary, where it is in powers of two.

So far, binary seems to look like a very useful thing, but only for things like number crunching and excel and stuff like that. How would it represent things like A or Z?

Computers standardise it, just like how us humans standardise that A is the first letter, computers standardise it as 65. It uses what is known as ASCII. It becomes sort of an inception because the 1 gets represented as another number but thankfully it ends there.

However, if you want to represent things such as emojis, you'd need unicode, which just uses another byte to store them. Nowadays, it does not really cost much to actually do this.

RGB

It stands for RGB. The computer represents how much red it should show, green, and blue. For example, with the numbers 72 73 33, your phone might interpret it as “Hi!”, however, photoshop, it might be seen as a colour. To make an image, it is tiny little pixels of these colours.

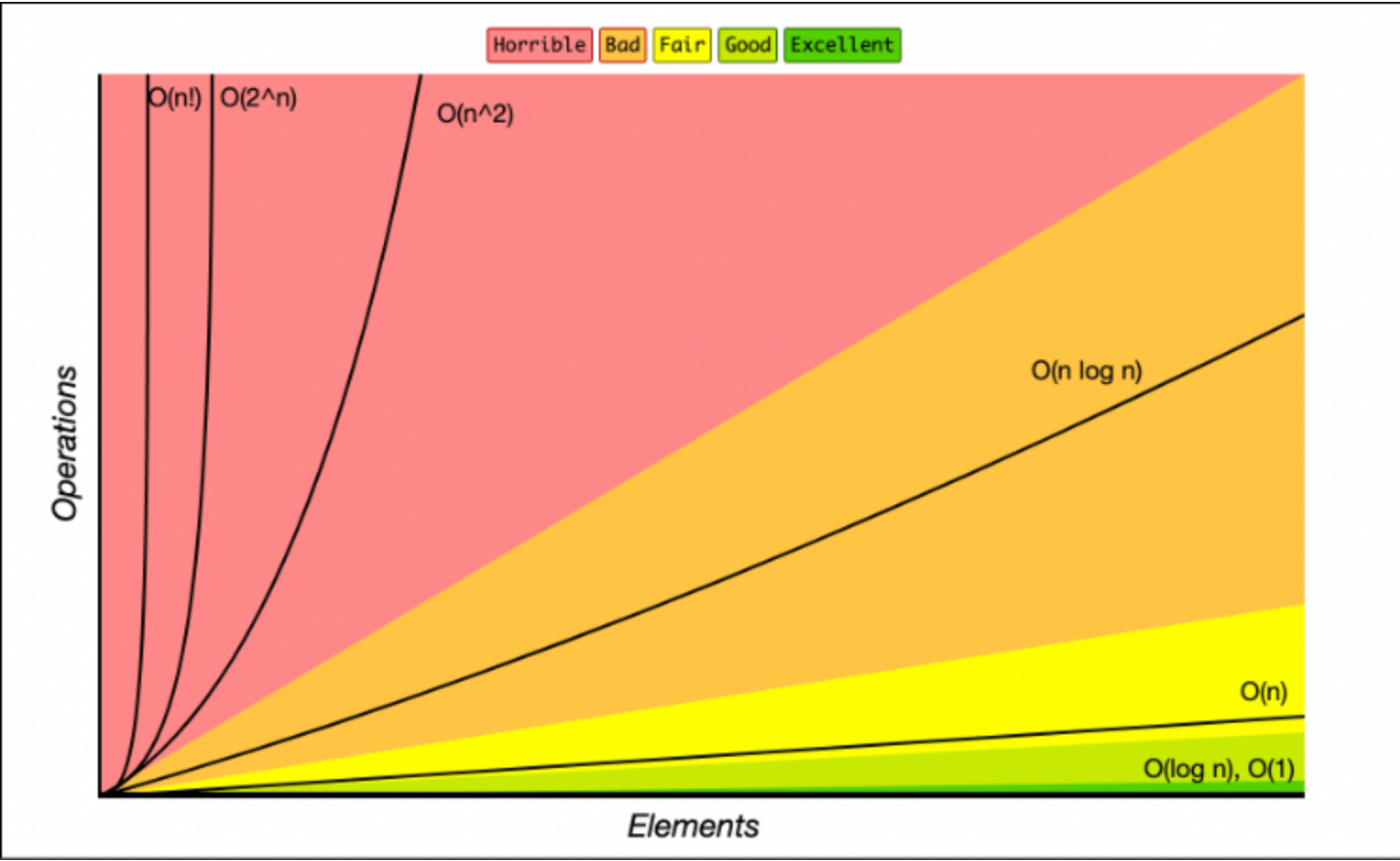
A video is just a compilation of images.

Algorithms

Abstraction means the simplification of something so that you don't focus on the lower level implementation, but you focus on the high level goals. An algorithm is a step by step instruction on how to solve a problem.

If you have for example, a list of names, one algorithm would be to split the list in half, see what the specific name you are looking for is in the direction relative to the half split, and then keep doing that. This is a “binary search” algorithm.

The time complexity of just searching through a phone book page by page is $O(n)$ meaning that it is linear. However, binary search is $O(\log_2(n))$, meaning it is much more effective.



Pseudocode

It is not a language, but however, it is in English like syntax. It is how many programmers would begin to solve a problem.

```
1  Pick up phone book
2  Open to middle of phone book
3  Look at page
4  If person is on page
5      Call person
6  Else if person is earlier in book
7      Open to middle of left half of book
8      Go back to line 3
9  Else if person is later in book
10     Open to middle of right half of book
11     Go back to line 3
12 Else
13     Quit
```

In our pseudocode, the verbs are known as functions, which will be used when programming.

A Boolean expression is just a question with a true or false response.

Indentation is also important. For example, line 5 will only occur if line 5 is true.

Scratch

Scratch is a very simple way to graphically visualise the code without the curly braces and all of that stuff.