



# CS 50 - .

## I - Computational thinking

Essentially, computer programming is about taking some inputs and creating some outputs - thus solving a problem. What happens in between is what we would call a black box.



For example, we may need to take attendance for a class. We could use a system called unary to count, a finger at a time. Computers today use a system called binary to count. It's from the term binary digit that we get a familiar term called bit. A bit is a zero or a one.

Computers only speak in terms of zeros and ones. Zero represent off. Ones represent on. Computers are millions, and perhaps billions, of transistors that are being turned on and off.

They generally use eight bits to represent a number. For example, 00000101 is the number 5 in binary.

## II - Text

Just as numbers are binary patterns of ones and zeros, letters are too!

Since there is an overlap between the 1s and 0s that represent numbers and letters, the **ASCII** standard was created to map specific letters to specific numbers.

### American Standard Code for Information Interchange

For example, the letter A was decided to map to the number 65.

If you received a text message, the binary under that message might represent the numbers 72, 73, 33. Mapping these out to ASCII, your message would look like! HI!

### III - Emojis

As time has rolled on, there are more and more ways to communicate via text.

Since there were not enough digits in binary to represent all the various characters that could be represented by humans, the **Unicode** standard expanded the number of bits that can be transmitted and understood by computers.

More and more features are being added to the **unicode** standard to represent further characters and emojis

### IV - RGB

Red, Green and Blue (called **RGB**) is a combination of 3 numbers

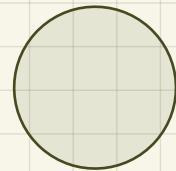
72	73	33
----	----	----

Red

Green

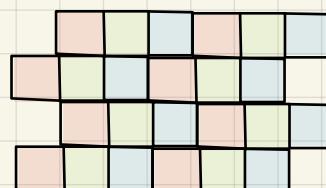
Blue

Taking our previous used 72, 73 and 33, which said HI! via text, would be interpreted by image readers as a light shade of yellow.



### V - Images, Video and Sound

Images are simply collections of RGB values.



Videos are sequences of many images that are stored together, just like a flipbook.

## VI. Algorithms

Problem solving is central to computer science and programming.

one approach could be to simply read from page to page one to the next to the next until reaching the last page.

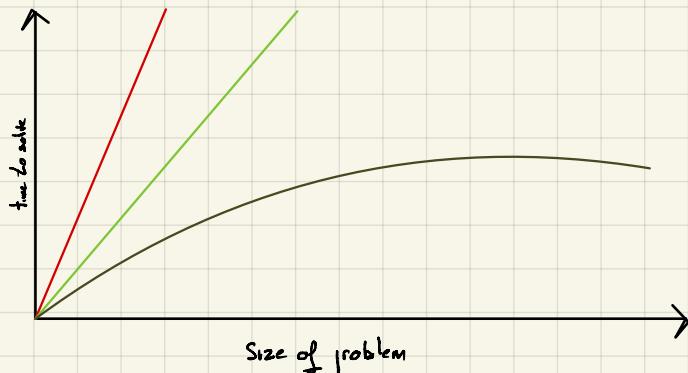


Imagine the basic problem of trying to locate a single name in a phone book.

Another approach could be to search two pages at a time.

A final and perhaps better approach could be to go to the middle of the book and ask, "Is the name I'm looking for to the left or to the right?" Then repeat the process, cutting the problem in half and half and half.

Each of the approaches could be called algorithms. The speed of each of these algorithms can be pictured as follows in what is called big-O notation.



## VII - Pseudocode and the Basic Building Blocks of Programming

Pseudocode is a human-readable version of your code. For example, considering the 3<sup>rd</sup> algorithm, we could compose pseudocode as follows -

The ability to create pseudocode is central to ones success.

Pick up the phone book  
Open to the middle of phone book  
Look at page  
If person is on page  
    Call person  
Else if person is earlier in book  
    Open to the middle of left half of book  
    Go back to line 3  
Else if person is later in book  
    Open to middle Of rig l ht half of book  
    Go back to line 3  
Else  
    Quit

Functions

Conditionals

boolean expressions

loops

★ When you pseudocode before you create formal code, It allows you to think through the logic of your problem in advance

★ You can later provide this information to others that are seeking to understand your coding decisions and how your code works