

# ① Computer science CS50 - Harvard

Computer science is not programming and inversely.  
Programming is just a tool. This is applicable to many fields. A computer gets input and generates output

## WEEK 0 lecture

The computer uses and understands only 0 and 1 (binary)  
This is a sufficient alphabet to represent any piece of data (number, video, graphics...).

bit = binary digit

To represent 50 with bits:

1	2	4	8	16	32	64	128
0	1	0	0	1	1	0	0

1 bit 0: 255

ASCII: standard that matches numbers to letters of the alphabet. Here, a part of it with upper-case letters.

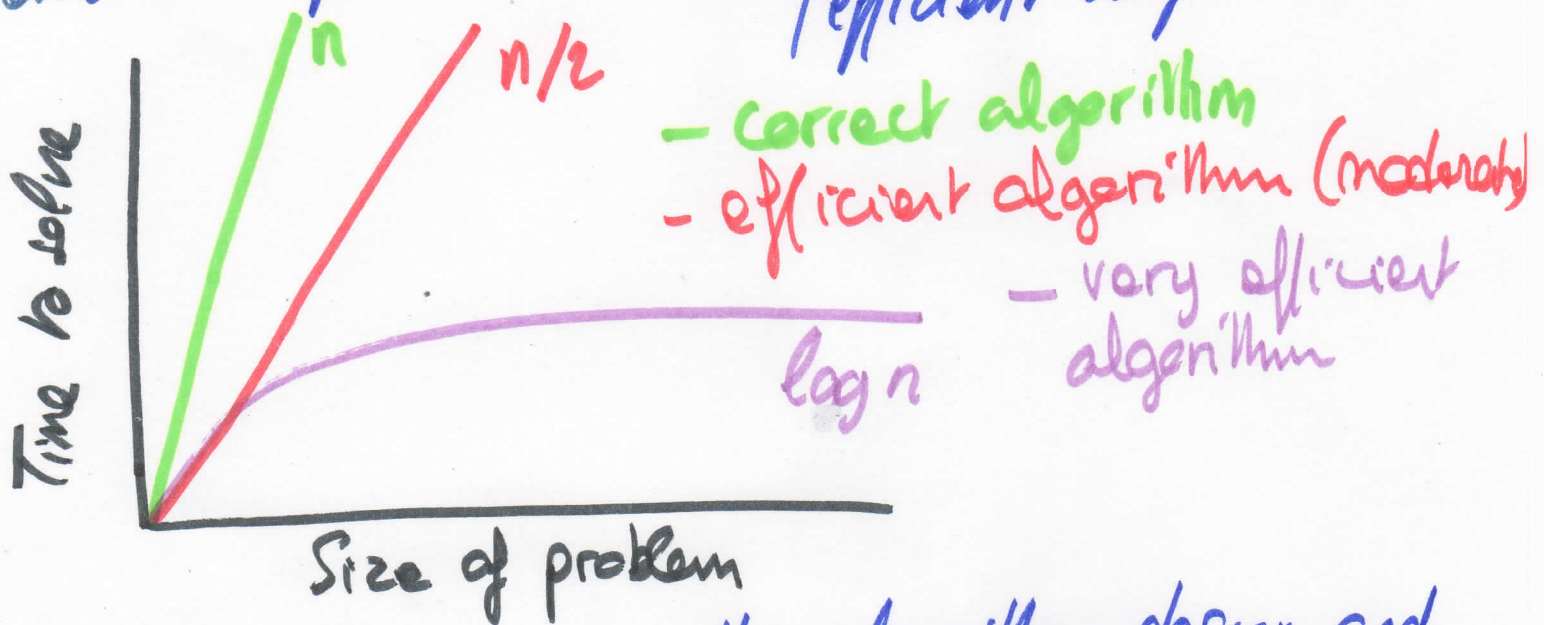
A	B	C	D	E	F	G	H	I	J	K	L	M	N
65	66	67	68	69	70	71	72	73	74	75	76	77	78
O	P	Q	R	S	T	U	V	W	X	Y	Z		
79	80	81	82	83	84	85	86	87	88	89	90		

E.g. H I  
(72) (73)

Therefore, patterns of numbers can easily be interpreted as letters.

## ② Computer science CS50 - Harvard

**Algorithm:** set of instructions that solves a problem.  
An algorithm takes an input and produces an output.  
It is one thing to implement something correctly and  
another to implement something efficiently. ⚠  
efficient algorithm



• Efficiency = careful with the algorithm design and input

Pseudocode:

- 1 pick up phone book
- 2 open to middle of phone book
- 3 look at names
- 4 if "Smith" is among names
- 5     call Mike
- 6 else if "Smith" is earlier in book
- 7     open to middle of left half of book

Help to anticipate  
the possibilities



### ③ Computer science CS50 - Harvard

#### Short week 0

**Algorithm**: a set of step by step instructions a computer program follows to do something.

**1960's**: different standards for representing letters on the computer → problem = encoding between ≠ machines difficult

**ANSI**: American National Standard Institute  
They formed a committee to develop a common scheme:

**ASCII (1963)**: American standard Code for Information Interchange

128 characters = each with 7 bits

**Unicode**: standard that has been developed for all the characters left behind by ASCII (ñ, chinese char...)

<u>BIN</u>	<u>DEC</u>	<u>CHAR</u>	<u>BIN</u>	<u>DEC</u>	<u>CHAR</u>
1000001	65	A	1100001	97	a
1000010	66	B	1100010	98	b
1000011	67	C	1100011	99	c
1000100	68	D	1100100	100	d
1000101	69	E	1100101	101	e
1000110	70	F	1100110	102	f
1000111	71	G	1100111	103	g

# ④ Computer Science CS50 - Harvard

Decimal notation: (decem = 10) using 10 digits

0 through 9:  $437 = 4 \times 100 + 3 \times 10 + 7 \times 1$   
 $= 4 \times 10^2 + 3 \times 10^1 + 7 \times 10^0$

10 is the special number for decimal notation since 10 is the base (exponential)

Binary notation: (bi = 2) 2 is the special number. We can represent as many numbers as with the decimal notation.

0: 0   1: 1   2: 10   3: 11   4: 100   5: 101   6: 110  
7: 111   8: 1000   ...

Find the number from binary notation

1	0	1	1	1	0	0	1	1
$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
256	128	64	32	16	8	4	2	1

$$1 \times 256 + 1 \times 64 + 1 \times 32 + 1 \times 16 + 1 \times 2 + 1 \times 1 = 371$$

Translate into binary notation

**237**: Find the largest power of 2 that is less than 237 which is 128. Then subtract 128 ( $237 - 128 = 109$ ) and repeat the process until 0. Put a 1 on every power of 2 chosen.

→

0	1	0	1	1	0	1	1	0	1
$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
256	128	64	32	16	8	4	2	1	



## ⑤ Computer science CS50 - Harvard

### Adding binary numbers

$$\begin{array}{r} 1101101 \\ + 1010010 \\ \hline 11000011 \end{array}$$

**Thread** = unit of processing / set of instructions that allow to run program concurrently (at the same time)  
We need a coordination between threads. We need to send message to one thread to another for coordination.

### Week 1 lecture

**Boolean expression** = question with an answer = Yes or No

**Compiler** : program that produces an object code.  
(machine computer code)

**Source code** → **compiler** → **object code**

C, C++... patterns of 0's and 1's  $\left( \begin{array}{l} 0010001 \\ 0100100 \dots \end{array} \right)$

**\n** : new line

**cd** : change directory e.g. cd Dropbox

**ls** : list (of files in the directory)

**hello-1.c : 7 : 21** → when error in C  
file line 7 character 21

# (6) Computer Science CS50 - Harvard

Main C library = `stdio.h` `#include <stdio.h>`

CS50 library: `cs50.h`

`./: .` = current directory `/` = separate directories/folders  
prompt (bash)

`rm`: remove file e.g. `rm hello`

`make filename` (c): convert the source code into code  
file name

`clang`: compiler e.g. `clang hello.c`  
it creates an output file (object code) `.out` (by default "a")

`mv`: move (filename) `mv a.out hello`

Everything that starts with a backslash "`\`" does something special. E.g. `\n`, `\'`, `\r`, `\\`

Placeholders: `%d` for decimals, `%s` for strings  
`%i` for decimal numbers (integer), `%c` for single character

Type of variables: `float`, `int`...

Condition

`if (condition)` can be here as well  
`{`  
`// do this` (`//` for comments)  
`}`  
`else if (condition & & condition)` and  
`{` `// or`  
`// do that`  
`}`  
`else { }`



## ⑦ Computer Science CS 50 - Harvard

**Loops** - **for** (initialization; condition; updater)

```
{  
    // do this again and again  
}
```

- **while** (condition)

```
{  
    // do this again and again  
}
```

- **do**

```
{  
    // do this again and again  
}  
while (condition);
```

**Variable**: you have to tell the computer what type of variable you are going to define

E.g. **int** counter; → here integer  
counter = 0; or better **int** counter = 0;