Lecture 0 - Scratch

Computational Thinking

input → → output

Unary vs. binary

→ computers use this

can take two possible values

0 = off

1 = on

For example, using 3 transistors, we can count 2^3 possible values, including 0.

0 → 000
1 → 001
2 → 010
3 → 011
4 → 100
5 → 101
6 → 110
7 → 111

Why?

Computers use base-2 to count

2^2 2^1 2^0

4 2 1

→ Computers usually use 8 bits to represent a number. This gives 2^8 possible values (256).

Text - ASCII

0	NUL	16	DLE	32	SP	48	0	64	@	80	P	96	`	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(	56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41	)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[	107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	|
13	CR	29	GS	45	-	61	=	77	M	93	]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

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

ASCII standard was created to map specific letters to specific numbers.

Mapping "Hi!" out to ASCII, the message would look as follows: 72 73 33

Emoji - Unicode

The Unicode standard was expanded to accomodate more characters beyond binary.

Emojis posed a challenge when assigning various skin tones to each one for personalization.

To solve this issue, creators and contributors of emojis decided to use the initial bits for the structure, and the following bits for the chosen skin tone.

RGB

72 73 33 →

Red, green and blue (RGB) is a combination of 3 numbers.

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

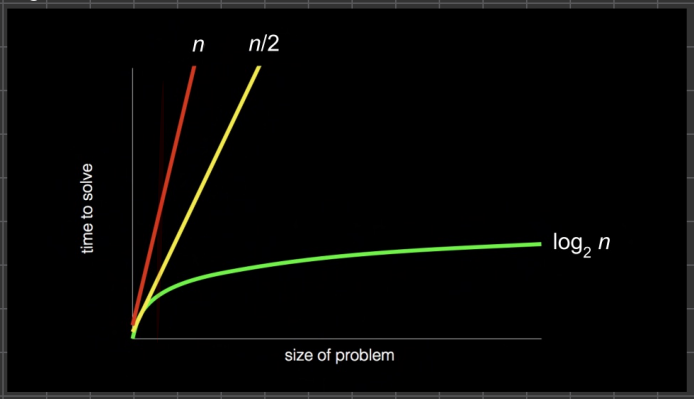
Images, Video and Sound

Images are simply collections of RGB values called pixels.

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

Music can be represented through MIDI data

Algorithms



Notice that the first algorithm, highlighted in red, has a big-O of n because if there are 100 names in the phone book, it could take up to 100 tries to find the correct name. The second algorithm, where two pages were searched at a time, has a big-O of 'n/2' because we searched twice as fast through the pages. The final algorithm has a big-O of log2n as doubling the problem would only result in one more step to solve the problem.

Problem solving is central to computer science and computer programming.

Imagine the basic problem of trying to locate a single name in a phone book. How might you go about this?

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

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 phone book and ask, "Is the name I am looking for to the left or to the right?" Then, repeat this process, cutting the problem in half and half and half.

Each of these approaches could be called algorithms.

speed

Pseudocode and the Basic Building Blocks of Programming

```
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
```

The ability to create pseudocode is central to one's success in computer programming

human-readable version of your code. For example

Pseudocode allows you to → think through the logic of your problem in advance

→ later provide this info to others that are seeking to understand your code.