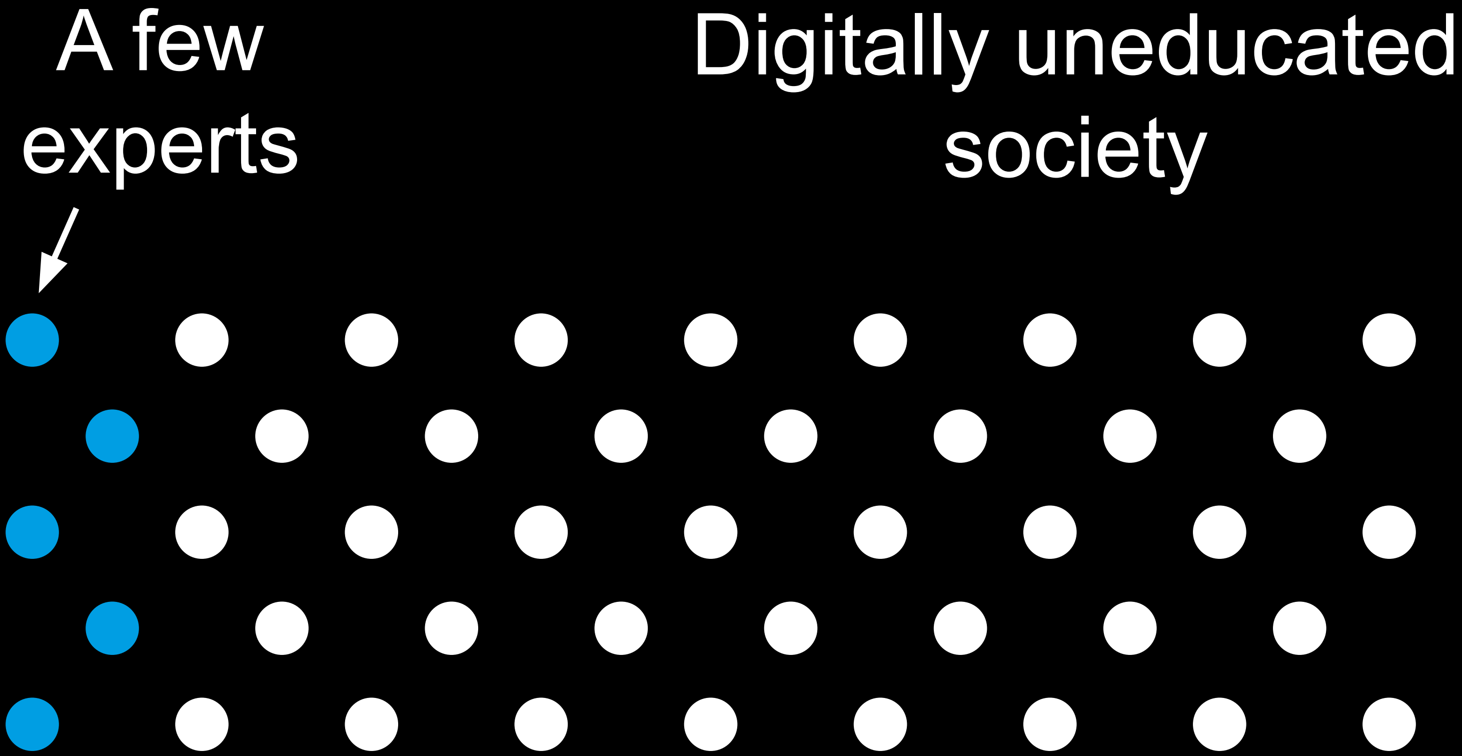
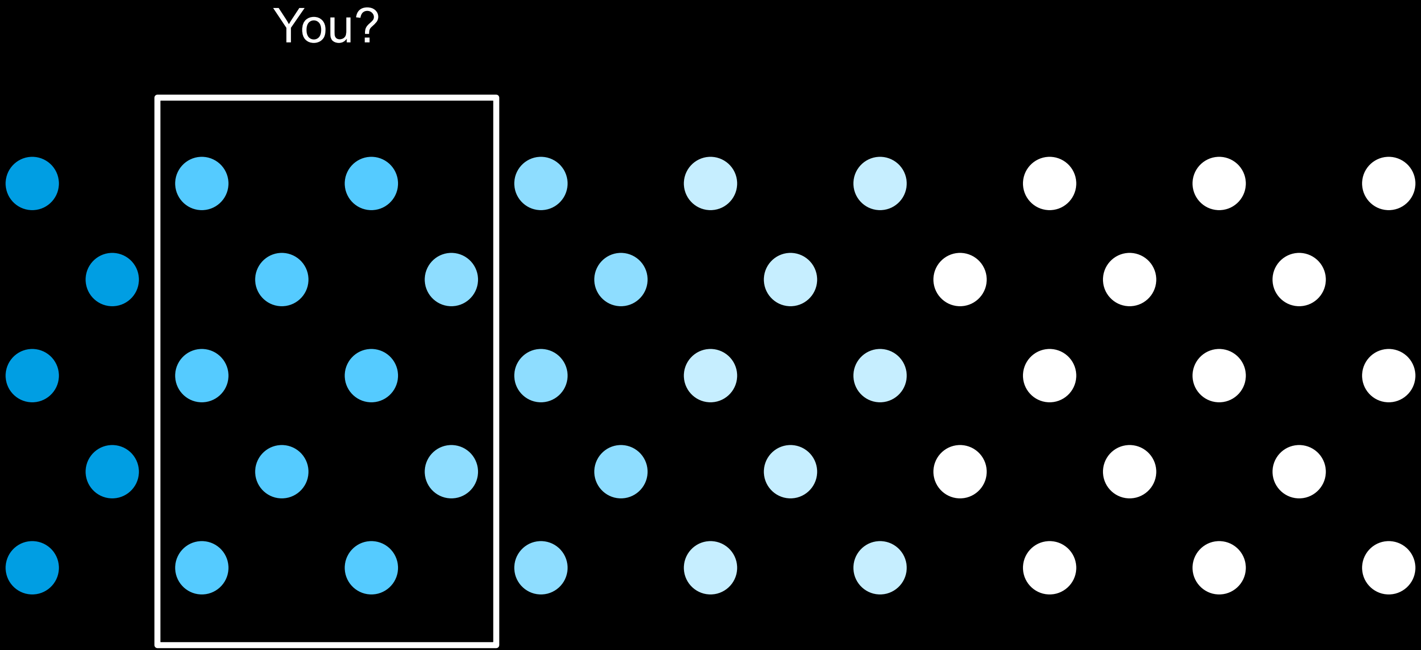
MOTIVATION



Digitally illiterate society with a few experts

Collective Understanding



Society with a distributed and high degree of digital education

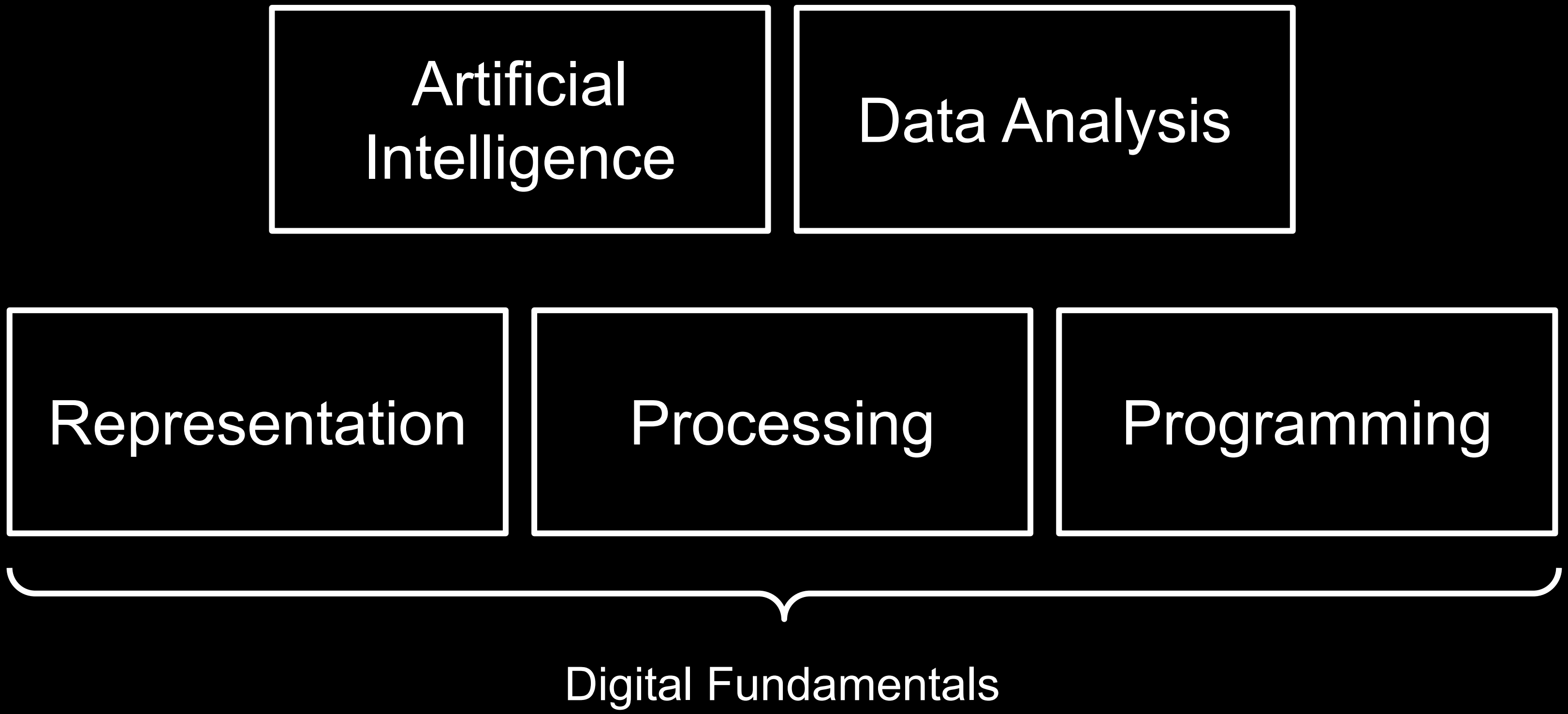
Artificial Intelligence

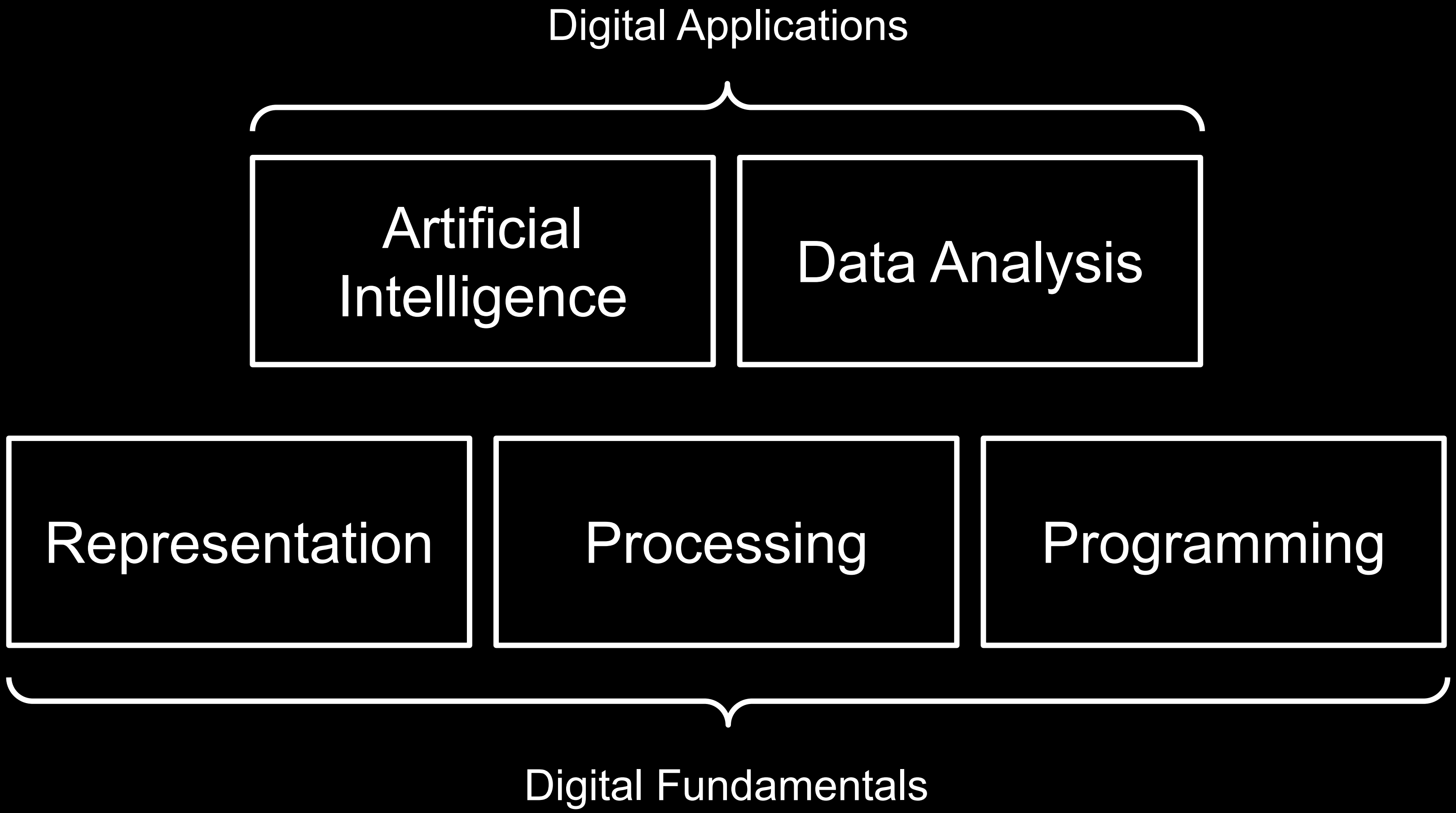
Data Analysis

Representation

Processing

Programming


Artificial Intelligence
Data Analysis
Representation
Processing
Programming
Digital Fundamentals


Digital Applications
Artificial Intelligence
Data Analysis
Representation
Processing
Programming
Digital Fundamentals

Digital Applications

Artificial Intelligence

Data Analysis

Representation

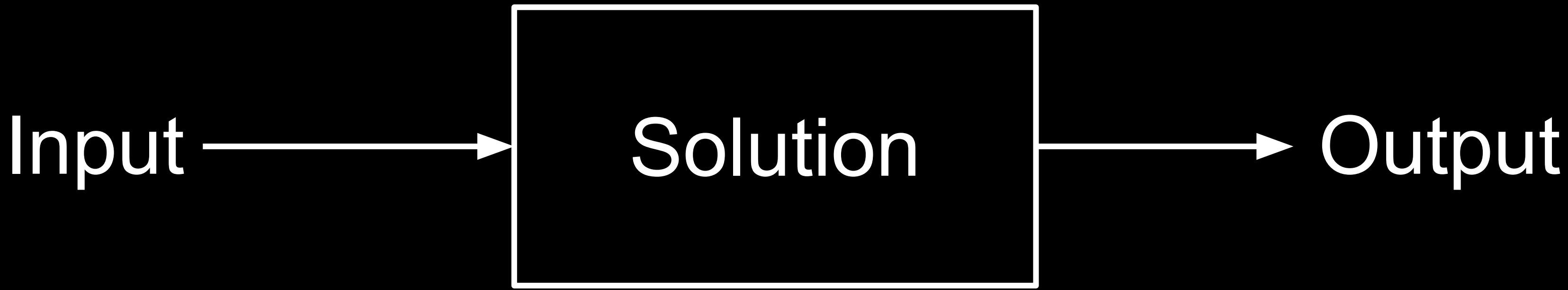
Processing

Programming

Digital Fundamentals

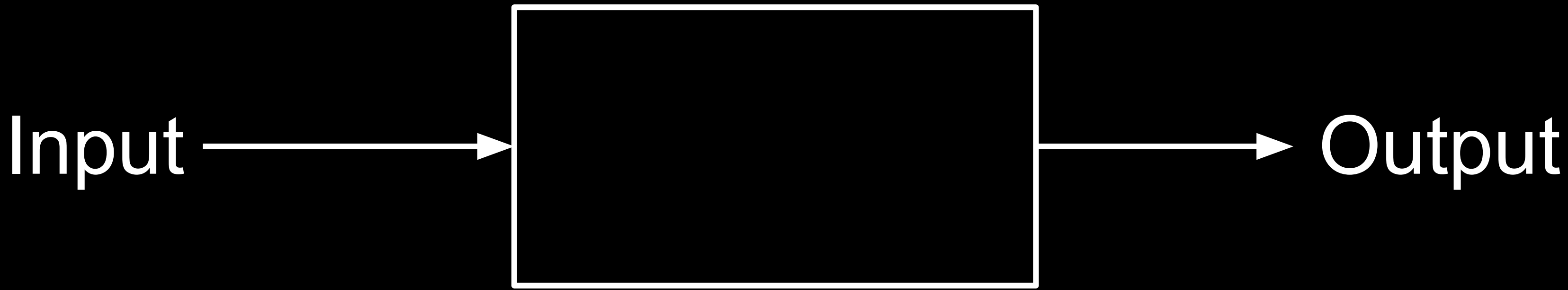
PROBLEM SOLVING

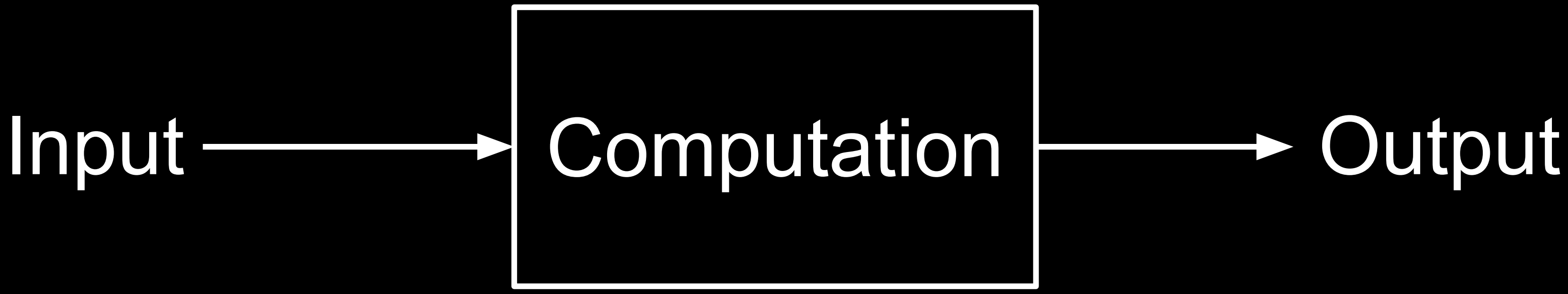
A Model for Solving Problems



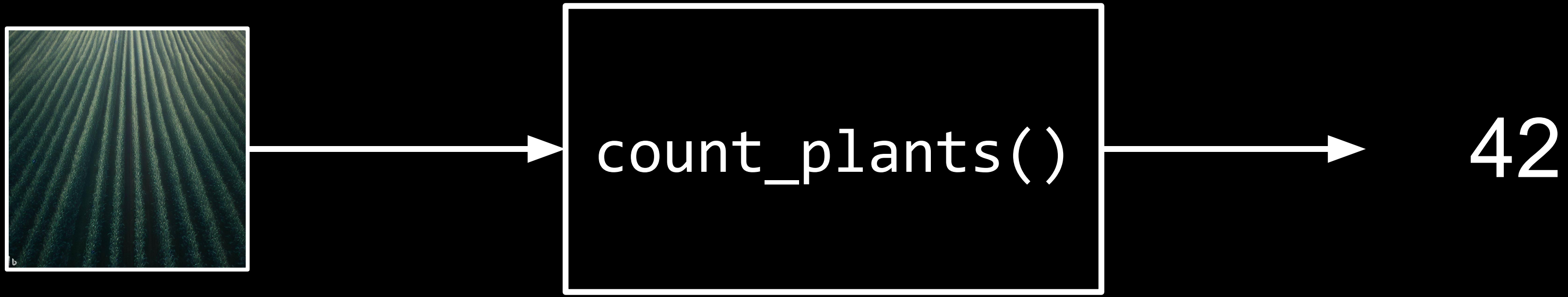
A Model for Solving Problems

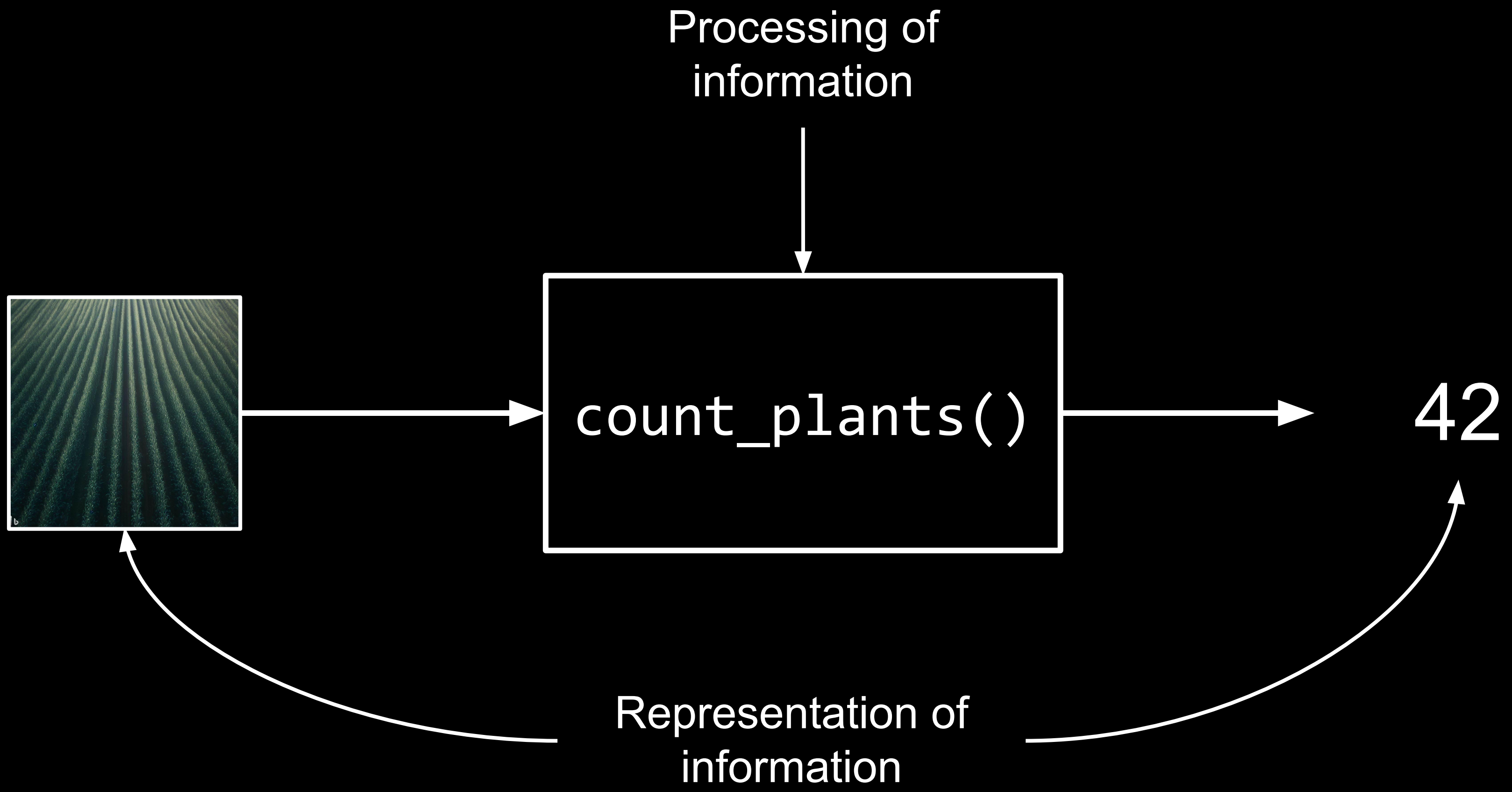
$$\begin{tikzcd}
\{a, b\} \arrow[r] & \boxed{a + b} \arrow[r] & s
\end{tikzcd}$$

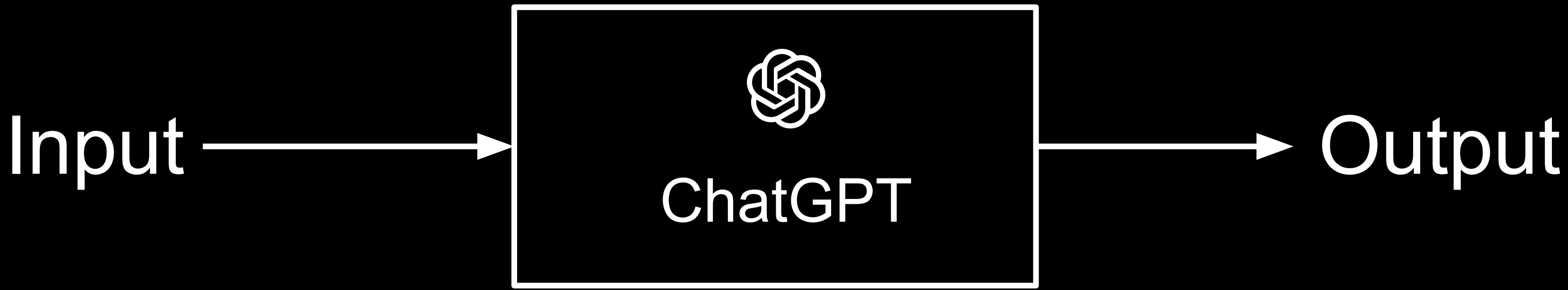

Input
Output

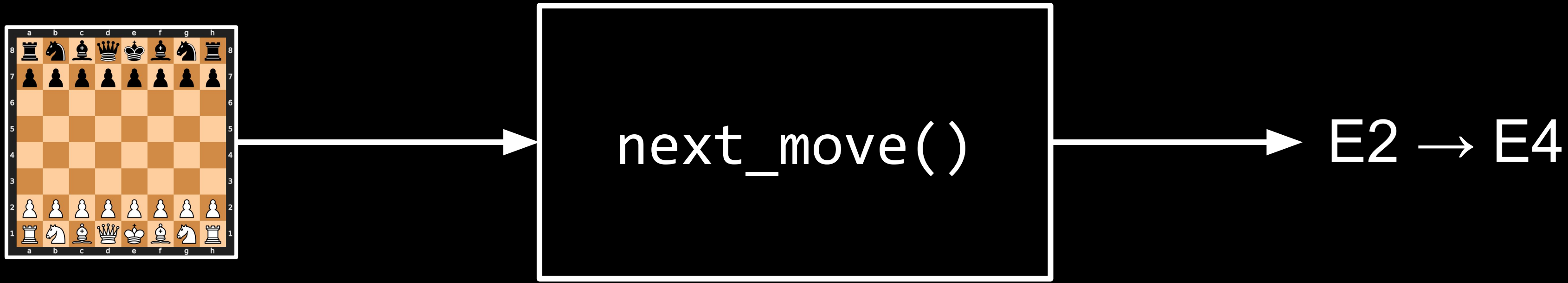

Input
Computation
Output

$$\begin{tikzcd}\text{Input} \arrow[r] & \boxed{\texttt{count\_plants()}} \arrow[r] & \text{Output}\end{tikzcd}$$


count_plants()
42


Processing of information
count_plants()
42
Representation of information

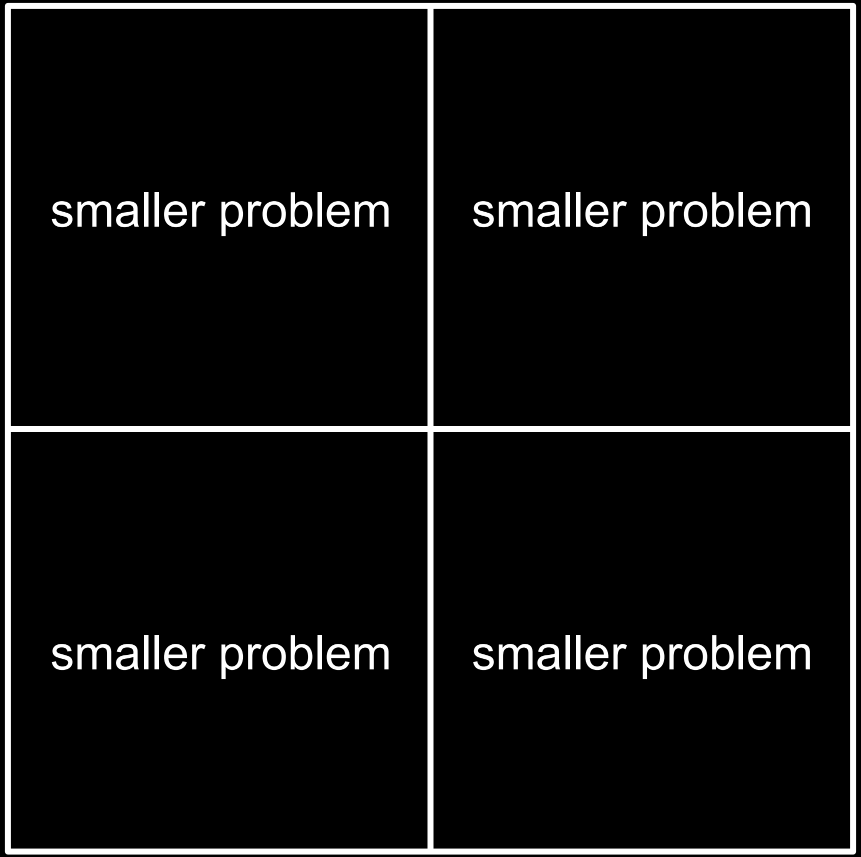

Input
ChatGPT
Output

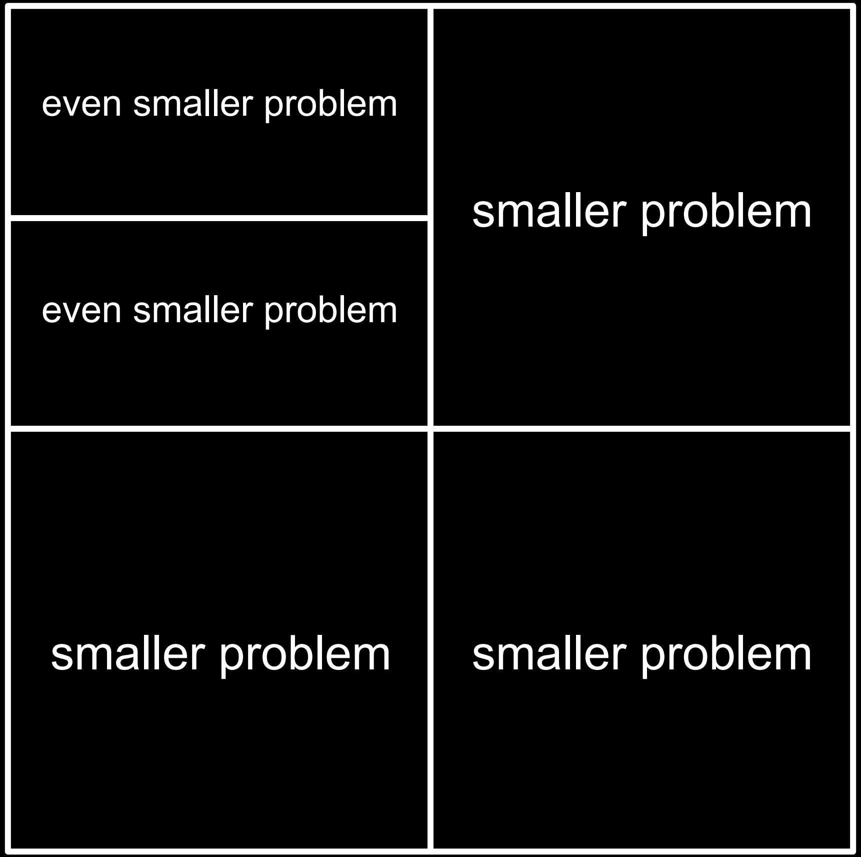

next_move()
E2 → E4

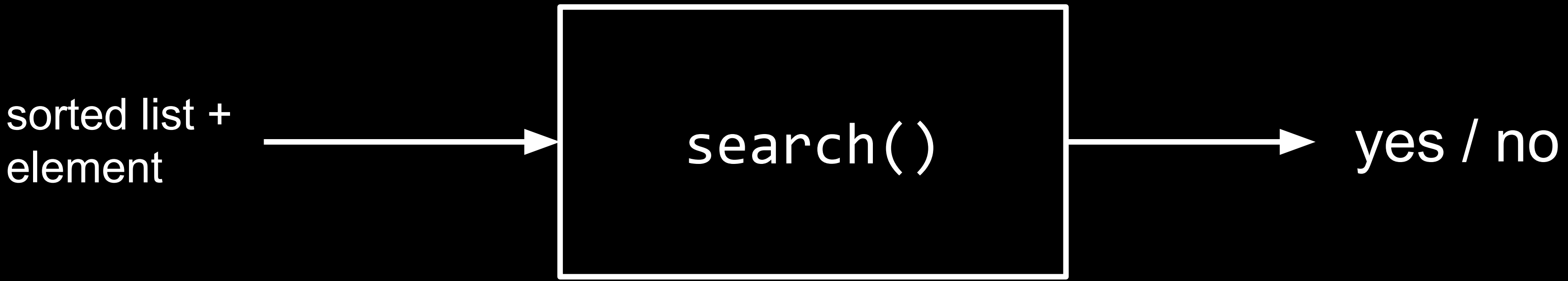
problem solving strategies

divide and conquer

large and complex problem


smaller problem
smaller problem
smaller problem
smaller problem


even smaller problem
even smaller problem
smaller problem
smaller problem
smaller problem


sorted list +
element
search()
yes / no

is 67 a prime number?

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97
```

linear search

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97
                    ⬆
```

19 steps… can we do better?

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97
```

binary search

67 != 41

$\downarrow$

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, ~~41~~,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97
```

binary search

67 > 41

⬇

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, ~~41~~,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97
```

binary search

67 > 41

⬇

~~2~~, ~~3~~, ~~5~~, ~~7~~, ~~11~~, ~~13~~, ~~17~~, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

binary search

~~2~~, ~~3~~, ~~5~~, ~~7~~, ~~11~~, ~~13~~, ~~17~~, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

$\uparrow$

67 != 71

binary search

~~2~~, ~~3~~, ~~5~~, ~~7~~, ~~11~~, ~~13~~, ~~17~~, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
43, 47, 53, 59, 61, 67, ~~71~~, 73, 79, 83, 89, 97

⬆

67 != 71

binary search

~~2~~, ~~3~~, ~~5~~, ~~7~~, ~~11~~, ~~13~~, ~~17~~, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
43, 47, 53, 59, 61, 67, ~~71~~, ~~73~~, ~~79~~, ~~83~~, ~~89~~, ~~97~~

⬆

67 < 71

binary search

~~2~~, ~~3~~, ~~5~~, ~~7~~, ~~11~~, ~~13~~, ~~17~~, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
43, 47, 53, ~~59~~, 61, 67, ~~71~~, ~~73~~, ~~79~~, ~~83~~, ~~89~~, ~~97~~

⬆

67 != 59

binary search

~~2~~, ~~3~~, ~~5~~, ~~7~~, ~~11~~, ~~13~~, ~~17~~, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
~~43~~, ~~47~~, ~~53~~, ~~59~~, 61, 67, ~~71~~, ~~73~~, ~~79~~, ~~83~~, ~~89~~, ~~97~~

⬆

67 > 59

binary search

~~2~~, ~~3~~, ~~5~~, ~~7~~, ~~11~~, ~~13~~, ~~17~~, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
~~43~~, ~~47~~, ~~53~~, ~~59~~, 61, 67, ~~71~~, ~~73~~, ~~79~~, ~~83~~, ~~89~~, ~~97~~

⬆

67 = 67

binary search

~~2~~, ~~3~~, ~~5~~, ~~7~~, ~~11~~, ~~13~~, ~~17~~, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
~~43~~, ~~47~~, ~~53~~, ~~59~~, ~~61,~~ 67, ~~71~~, ~~73~~, ~~79~~, ~~83~~, ~~89~~, ~~97~~

⬆

67 = 67

3 splits → much better

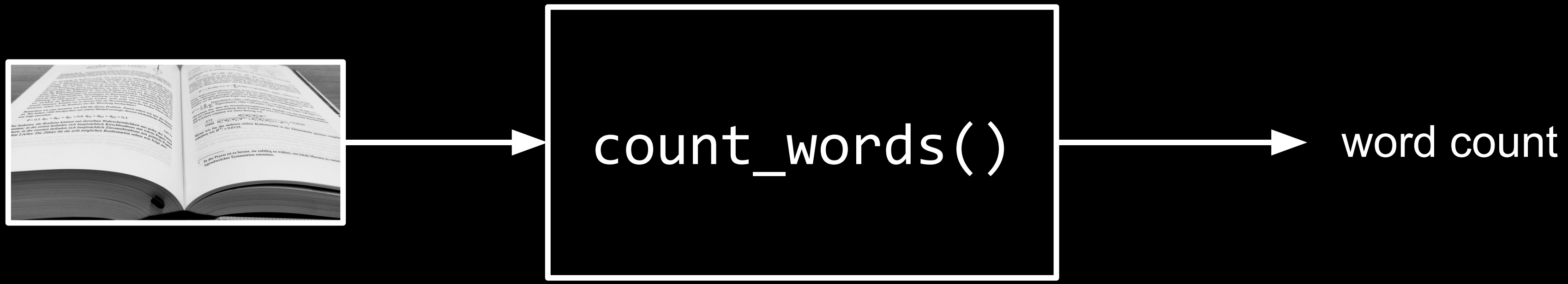
~~2~~, ~~3~~, ~~5~~, ~~7~~, ~~11~~, ~~13~~, ~~17~~, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
~~43~~, ~~47~~, ~~53~~, ~~59~~, ~~61,~~ 67, ~~71~~, ~~73~~, ~~79~~, ~~83~~, ~~89~~, ~~97~~

⬆

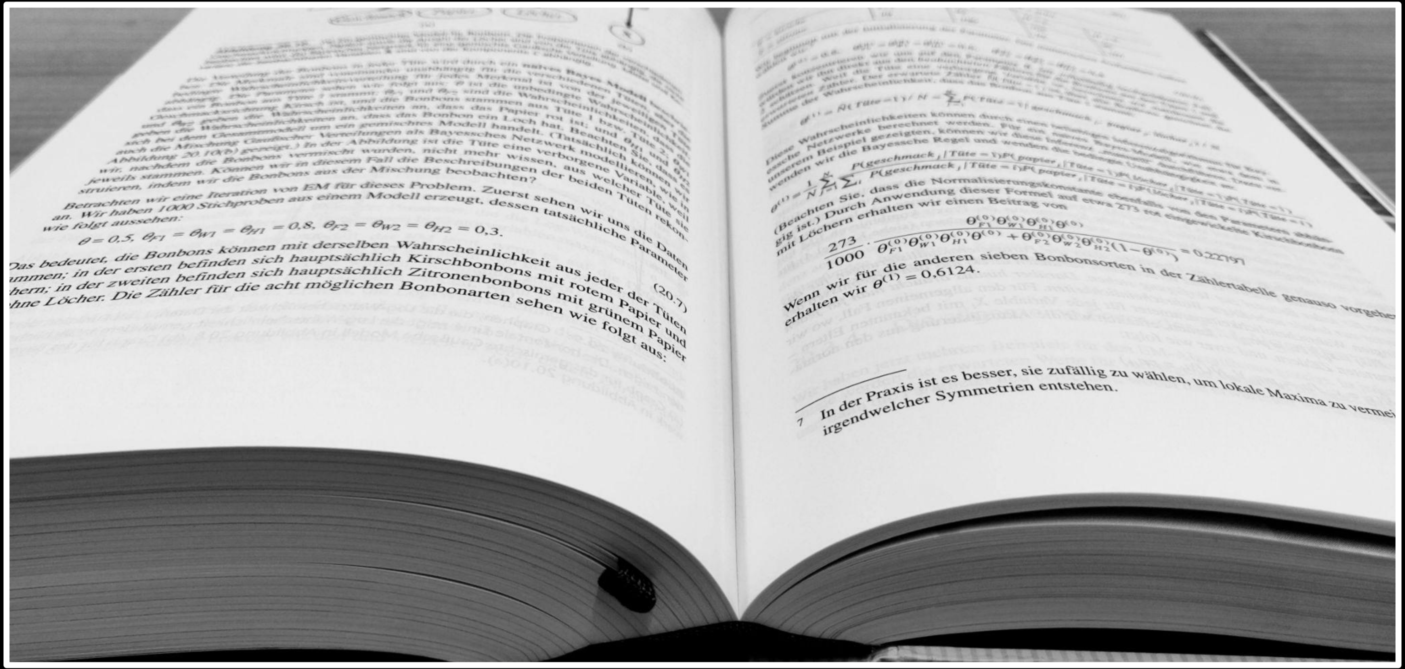
67 = 67

🤔

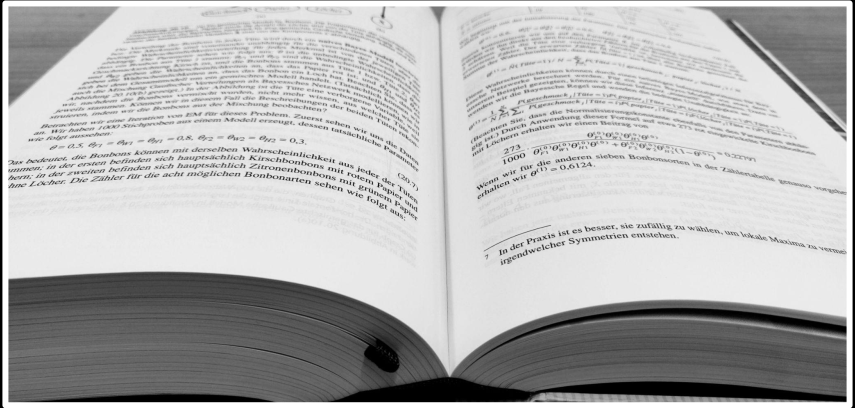
how efficient are linear and binary search in general?


count_words()
word count

how many words are in the book?



strategies, anyone?



[illegible] jeweils stammen. Können wir in diesem Fall die Beschreibungen der beiden Tüten rekon-
struieren, indem wir die Bonbons aus der Mischung beobachten?

Betrachten wir eine Iteration von EM für dieses Problem. Zuerst sehen wir uns die Daten an. Wir haben 1000 Stichproben aus einem Modell erzeugt, dessen tatsächliche Parameter wie folgt aussehen:

$$\theta = 0{,}5,\ \theta_{F1} = \theta_{W1} = \theta_{H1} = 0{,}8,\ \theta_{F2} = \theta_{W2} = \theta_{H2} = 0{,}3. \quad (20.7)$$

Das bedeutet, die Bonbons können mit derselben Wahrscheinlichkeit aus jeder der Tüten [illegible]mmen; in der ersten befinden sich hauptsächlich Kirschbonbons mit rotem Papier und [illegible]hern; in der zweiten befinden sich hauptsächlich Zitronenbonbons mit grünem Papier [illegible]hne Löcher. Die Zähler für die acht möglichen Bonbonarten sehen wie folgt aus:

[illegible]

Diese Wahrscheinlichkeiten können durch einen [illegible] [illegible]esche Netzwerke berechnet werden. [illegible] unserem Beispiel gezeigten, können wir diese [illegible] wenden wir die Bayessche Regel und wenden [illegible]

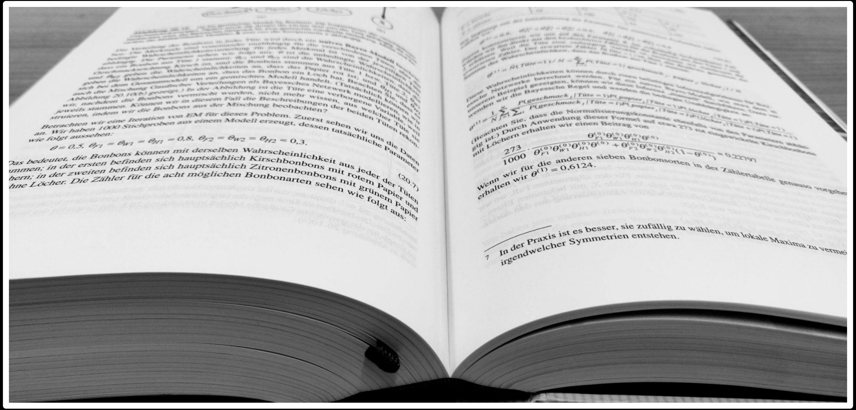
$$\theta^{(1)} = \frac{1}{N}\sum_{j=1}^{N}\frac{P(geschmack_j \mid Tüte = 1)P(papier_j \mid Tüte = 1) [illegible]}{\sum_i P(geschmack_j \mid Tüte = i)P(papier_j \mid Tüte = i) [illegible]}$$

(Beachten Sie, dass die Normalisierungskonstante ebenfalls von den Parametern abhän-gig ist.) Durch Anwendung dieser Formel auf etwa 273 rot eingewickelte Kirschbonbons mit Löchern erhalten wir einen Beitrag von

$$\frac{273}{1000}\cdot\frac{\theta^{(0)}_{F1}\theta^{(0)}_{W1}\theta^{(0)}_{H1}\theta^{(0)}}{\theta^{(0)}_{F1}\theta^{(0)}_{W1}\theta^{(0)}_{H1}\theta^{(0)} + \theta^{(0)}_{F2}\theta^{(0)}_{W2}\theta^{(0)}_{H2}(1-\theta^{(0)})} = 0{,}22797$$

Wenn wir für die anderen sieben Bonbonsorten in der Zählertabelle genauso vorgehe[illegible] erhalten wir $\theta^{(1)} = 0{,}6124$.

7 In der Praxis ist es besser, sie zufällig zu wählen, um lokale Maxima zu vermei[illegible] irgendwelcher Symmetrien entstehen.



[illegible]

Die Verteilung der Bonbons in jeder Tüte wird durch ein naives Bayes-Modell [illegible] ... sind die Wahrscheinlichkeiten, dass das Papier rot ist; und θ_{H1} und θ_{H2} geben die Wahrscheinlichkeiten an, dass das Bonbon ein Loch hat. Beachten Sie, dass es sich bei dem Gesamtmodell um ein gemischtes Modell handelt. (Tatsächlich können wir auch die Mischung Gaußscher Verteilungen als Bayessches Netzwerk modellieren, wie in Abbildung 20.10(b) gezeigt.) In der Abbildung ist die Tüte eine verborgene Variable, weil wir, nachdem die Bonbons vermischt wurden, nicht mehr wissen, aus welcher Tüte sie jeweils stammen. Können wir in diesem Fall die Beschreibungen der beiden Tüten rekonstruieren, indem wir die Bonbons aus der Mischung beobachten?

Betrachten wir eine Iteration von EM für dieses Problem. Zuerst sehen wir uns die Daten an. Wir haben 1000 Stichproben aus einem Modell erzeugt, dessen tatsächliche Parameter wie folgt aussehen:

$$\theta = 0{,}5,\ \theta_{F1} = \theta_{W1} = \theta_{H1} = 0{,}8,\ \theta_{F2} = \theta_{W2} = \theta_{H2} = 0{,}3. \qquad (20.7)$$

Das bedeutet, die Bonbons können mit derselben Wahrscheinlichkeit aus jeder der Tüten stammen; in der ersten befinden sich hauptsächlich Kirschbonbons mit rotem Papier und Löchern; in der zweiten befinden sich hauptsächlich Zitronenbonbons mit grünem Papier ohne Löcher. Die Zähler für die acht möglichen Bonbonarten sehen wie folgt aus:

[illegible]

Diese Wahrscheinlichkeiten können durch einen beliebigen Inferenzalgorithmus für Bayessche Netzwerke berechnet werden. Für ein naives Bayes-Modell, wie etwa dem in unserem Beispiel gezeigten, können wir diese Inferenz ... selbst ... Dabei verwenden wir die Bayessche Regel und wenden die bedingte Unabhängigkeit an:

$$\theta^{(1)} = \frac{1}{N}\sum_{j=1}^{N} \frac{P(geschmack_j \mid Tüte=1)P(papier_j \mid Tüte=1) \ldots}{\sum_i P(geschmack_j \mid Tüte=i)P(papier_j \mid Tüte=i)P(löcher_j \mid Tüte=i)P(Tüte=i)}$$

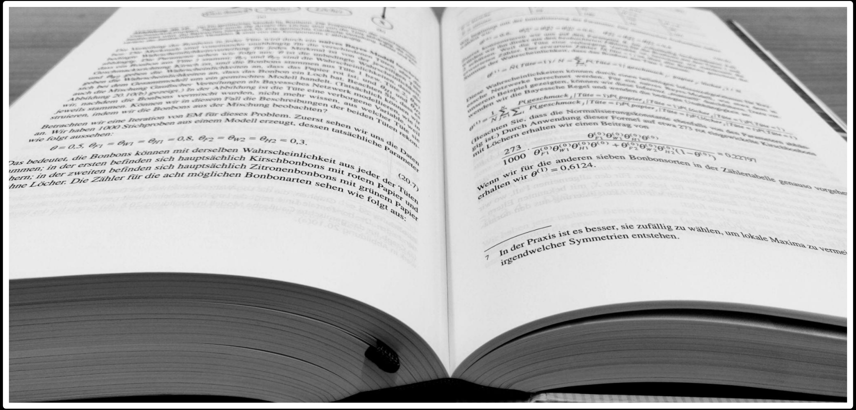
(Beachten Sie, dass die Normalisierungskonstante ebenfalls von den Parametern abhängig ist.) Durch Anwendung dieser Formel auf etwa 273 rot eingewickelte Kirschbonbons mit Löchern erhalten wir einen Beitrag von

$$\frac{273}{1000}\cdot\frac{\theta_{F1}^{(0)}\theta_{W1}^{(0)}\theta_{H1}^{(0)}\theta^{(0)}}{\theta_{F1}^{(0)}\theta_{W1}^{(0)}\theta_{H1}^{(0)}\theta^{(0)} + \theta_{F2}^{(0)}\theta_{W2}^{(0)}\theta_{H2}^{(0)}(1-\theta^{(0)})} = 0{,}22797$$

Wenn wir für die anderen sieben Bonbonsorten in der Zählertabelle genauso vorgehe erhalten wir $\theta^{(1)} = 0{,}6124$.

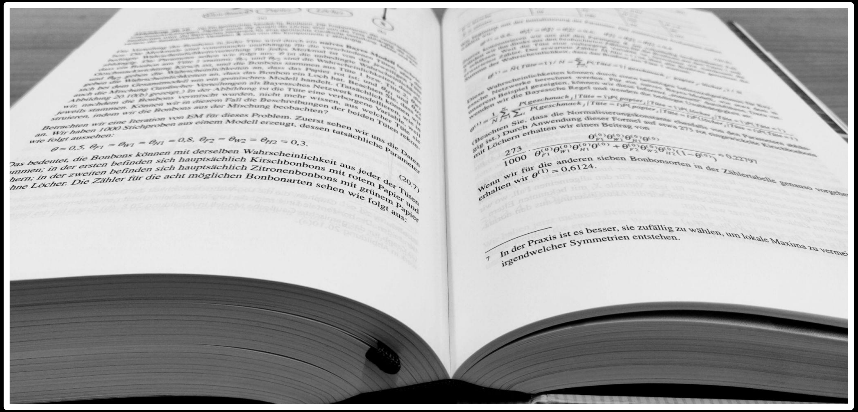
7 In der Praxis ist es besser, sie zufällig zu wählen, um lokale Maxima zu vermei irgendwelcher Symmetrien entstehen.

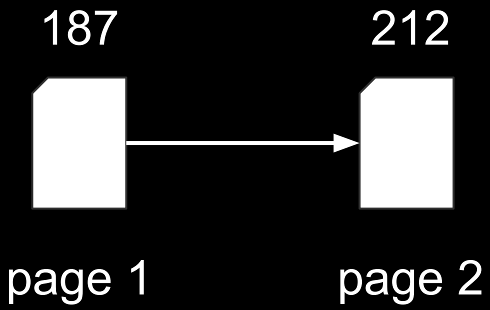
page 1

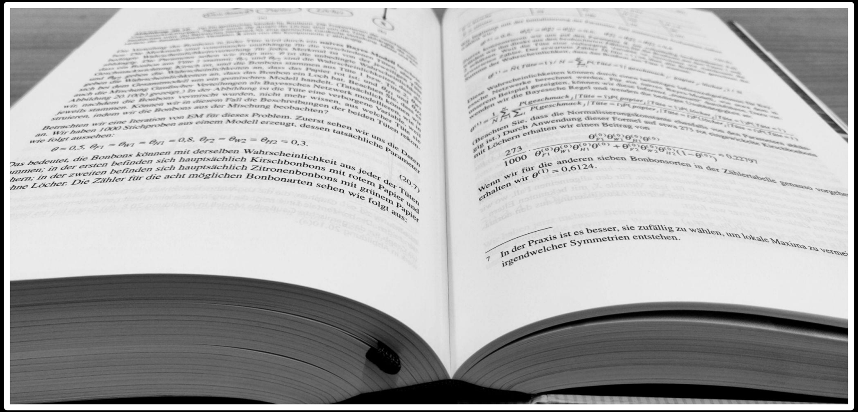


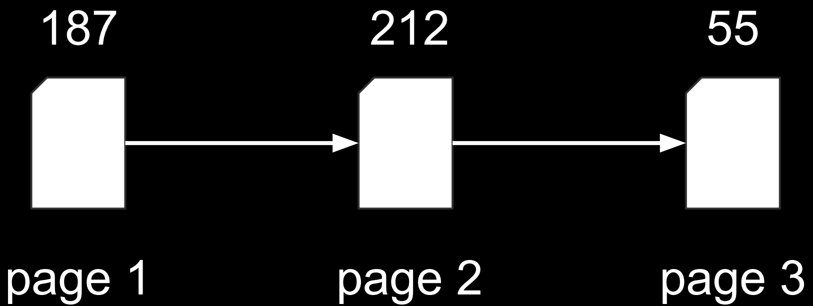
187

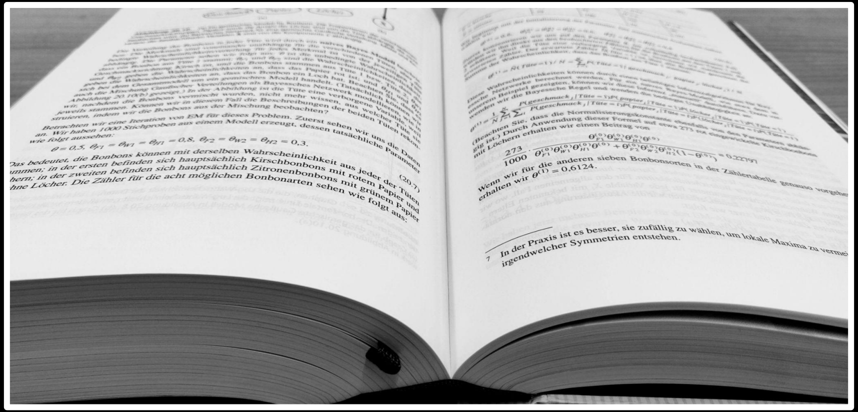
page 1

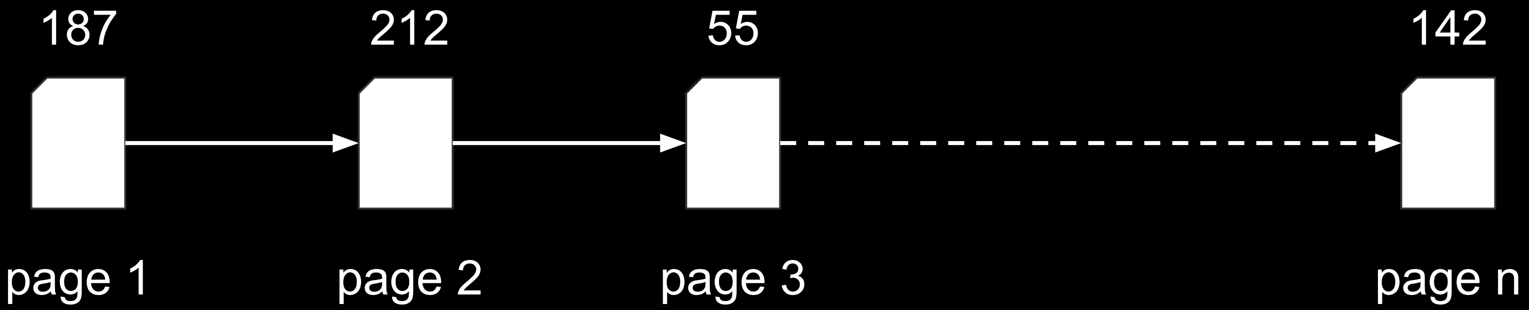


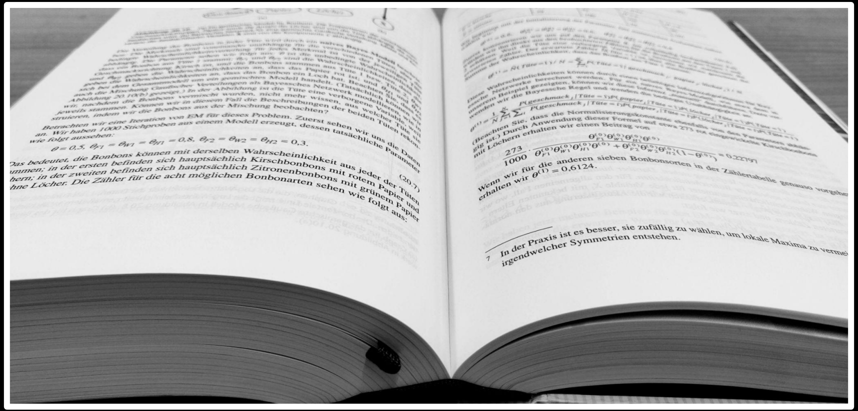

187
212
page 1
page 2




187
212
55
page 1
page 2
page 3




187
212
55
142
page 1
page 2
page 3
page n



😅

n = 1327 pages

Ø 2:23 minutes per page

~ 52.34 hours

divide and conquer

+

?

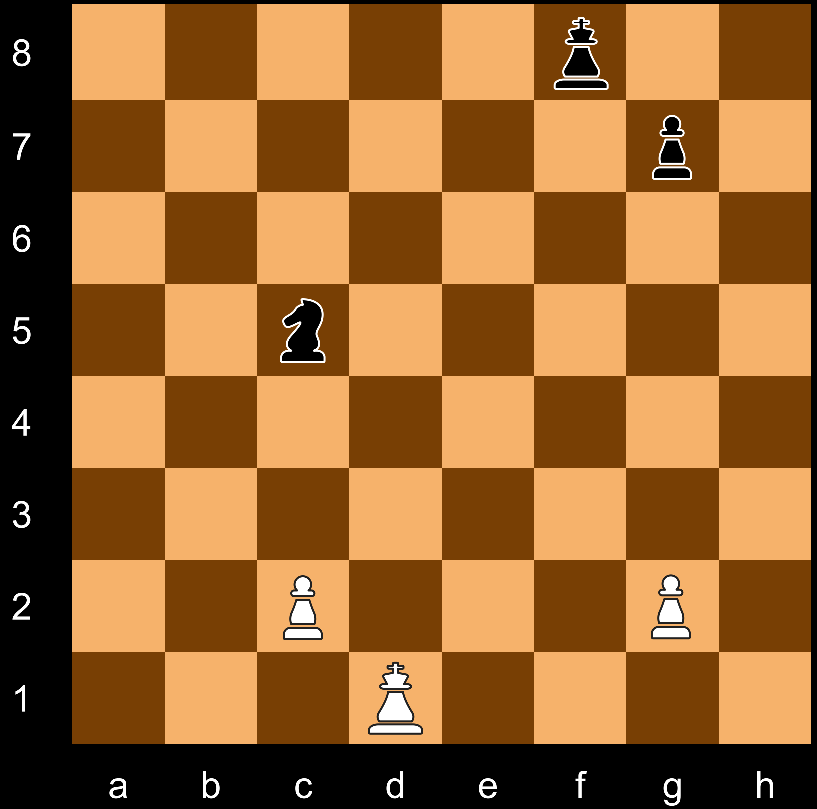
divide and conquer

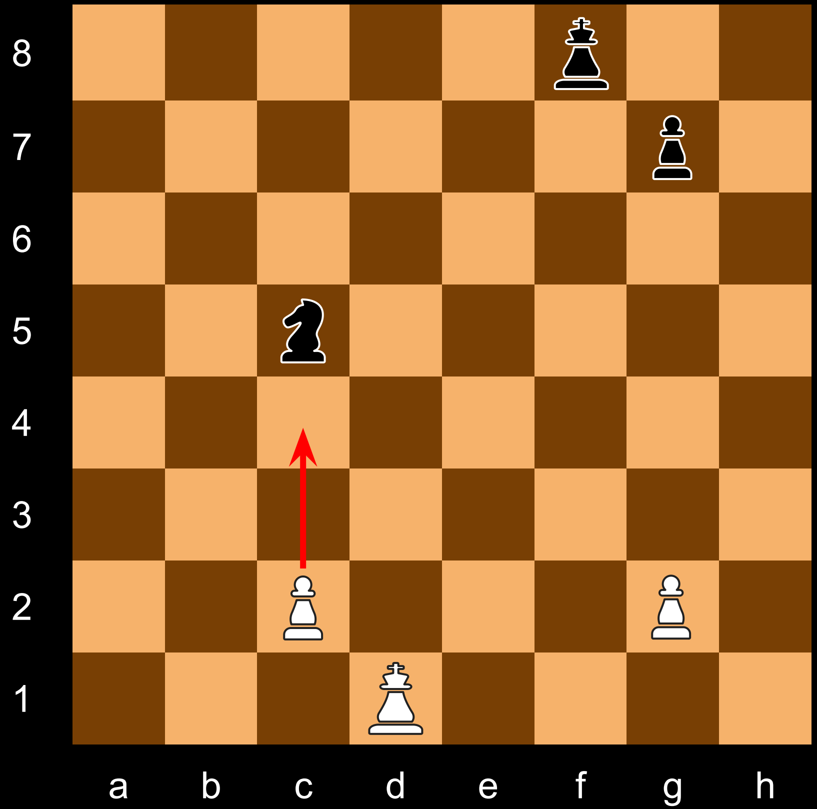
+

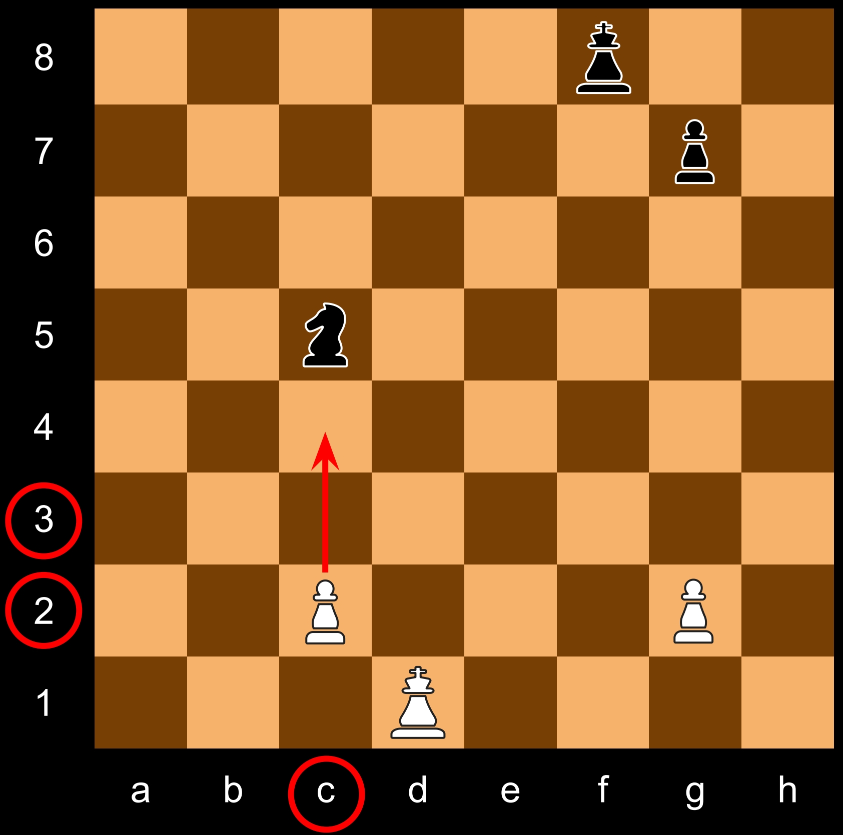
distribution and parallelization

INFORMATION

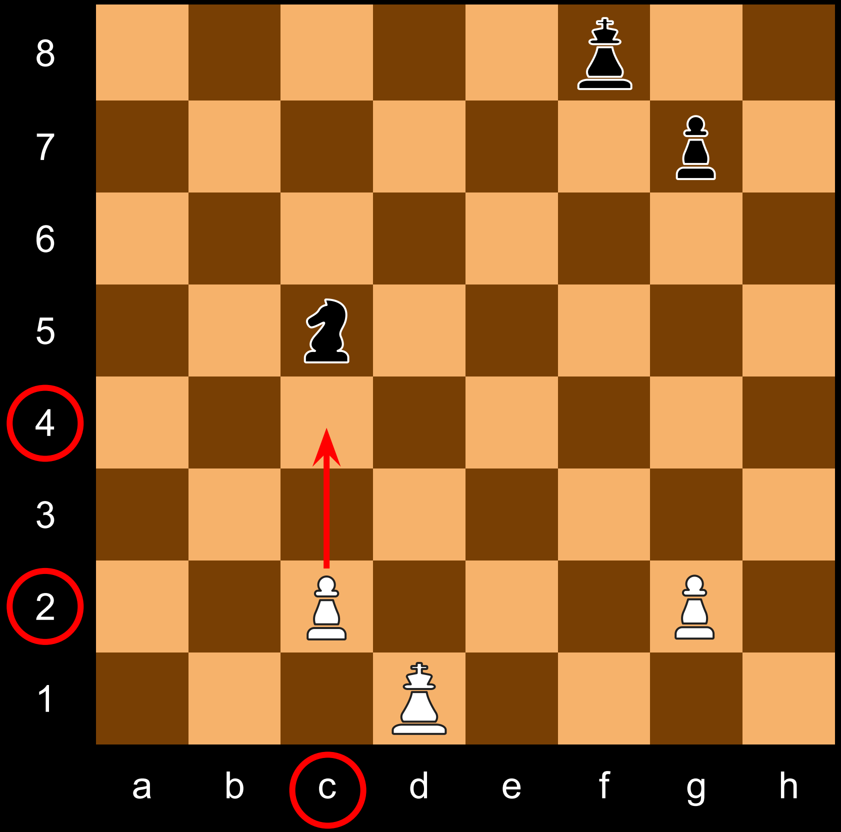
Image created with DALLE-3


8
7
6
5
4
3
2
1
a
b
c
d
e
f
g
h

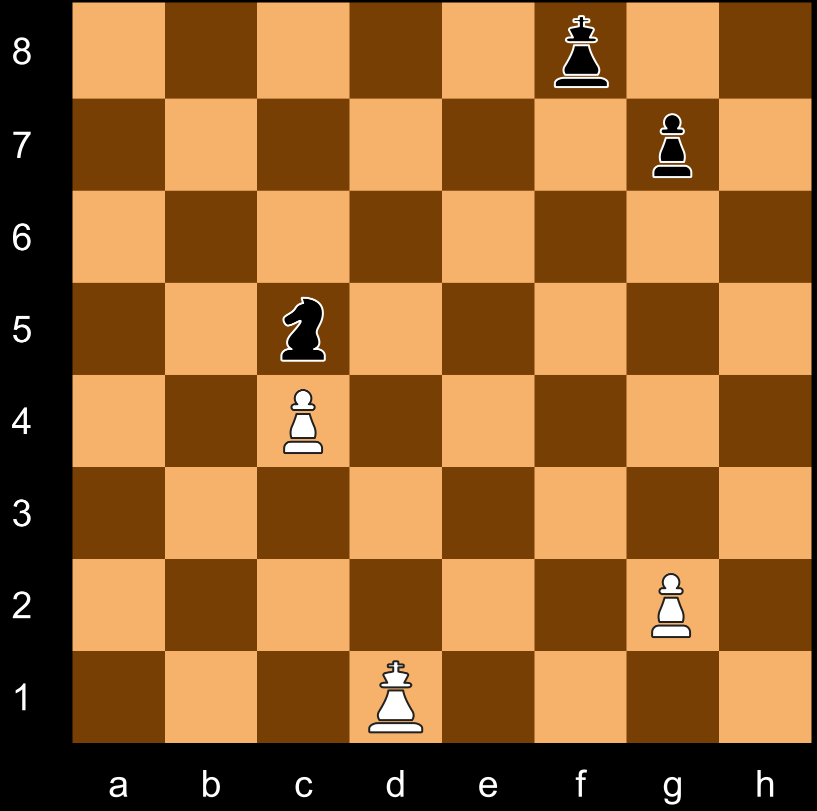

8
7
6
5
4
3
2
1
a
b
c
d
e
f
g
h


8
7
6
5
4
3
2
1
a
b
c
d
e
f
g
h

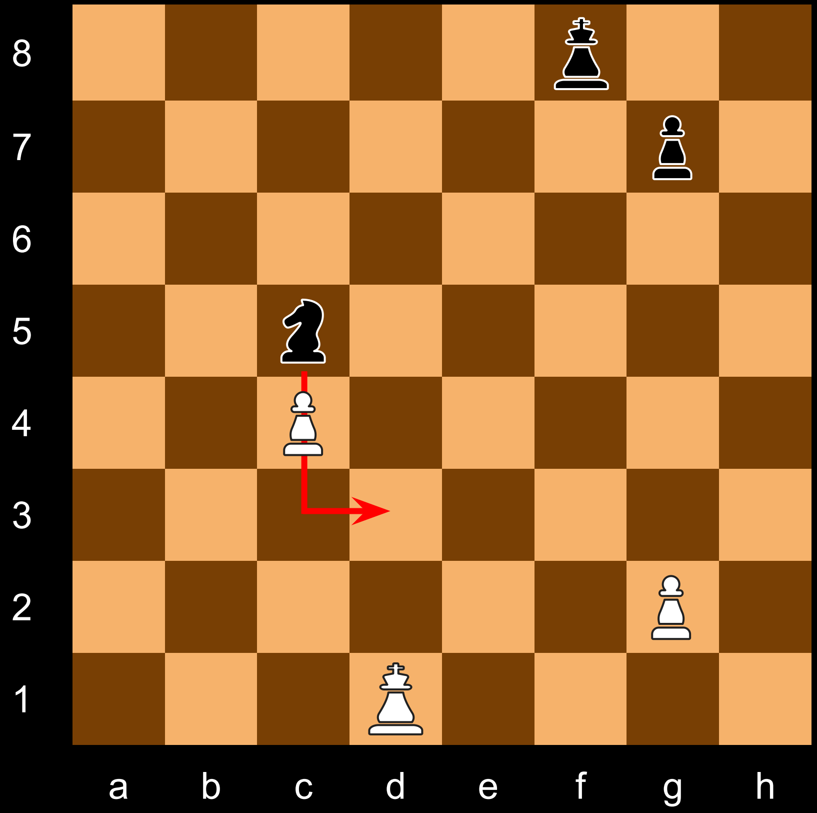
$c2 \rightarrow c4$



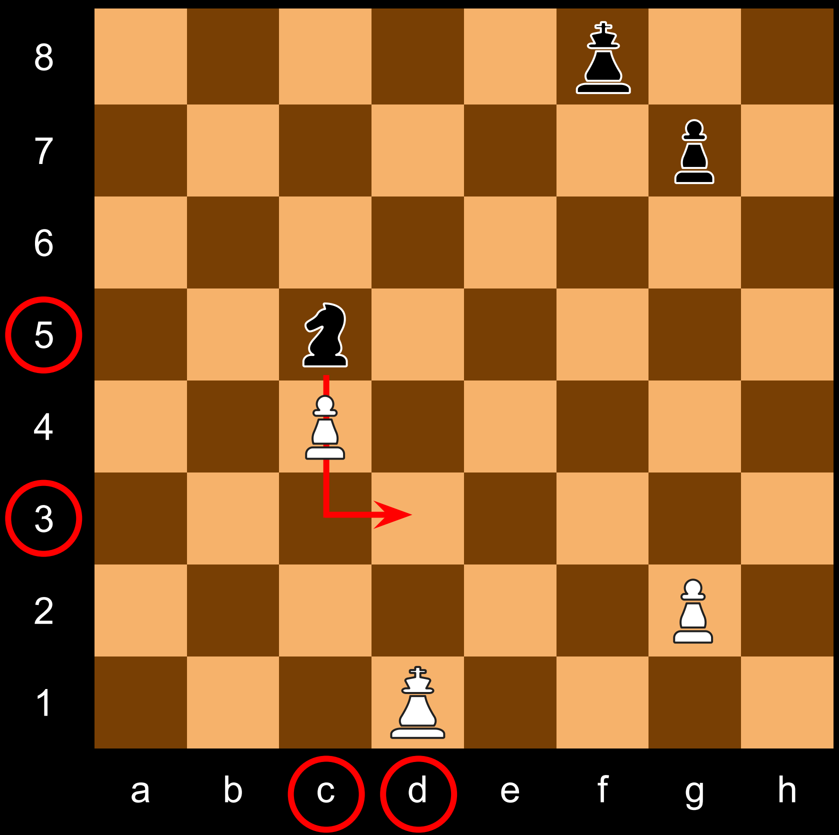
c2 → c4



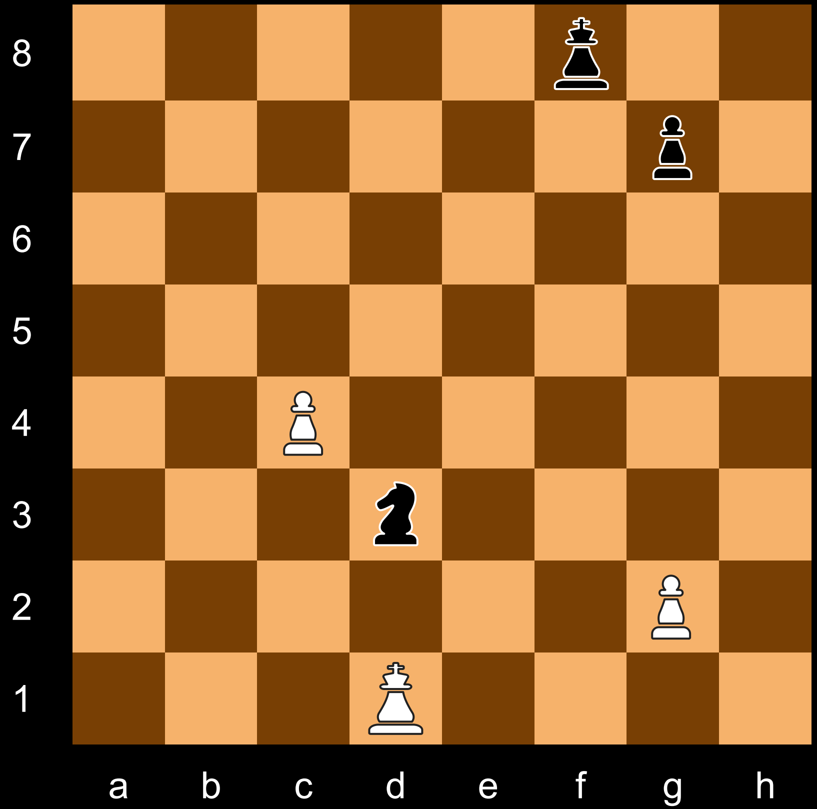
c2 → c4



$c2 \rightarrow c4$
$c5 \rightarrow d3$



c2 → c4
c5 → d3
…



{A}

_ _

{A}

A A

{A, B}

__ __

{A, B}

A A

{A, B}

A B

{A, B}

B A

{A, B}

B B

{A, B, C}

_ _ _

{A, B, C}

_ _

{A, B, C}

AA, AB, BA, BB,
AC, BC, CA, CB, CC

{A, B, C, D}

__ __

{A, B, C, D}

AA, AB, BA, BB, AC, BC, CA, CB,
CC, AD, DA, BD, DB, CD, DC, DD

{A, B, C, D, E}

_ _

{A, B, C, D, E}

AA, AB, BA, BB, AC, BC, CA, CB, CC, AD, DA, BD, DB, CD, DC, DD, AE, EA, BE, EB, CE, EC, DE, ED, EE

with length n = 2

# symbols	# messages
1	1
2	4
3	9
4	16
5	25

with length n = 2

# symbols		# messages
1		1
2		4
3	$\xrightarrow{f(x)}$	9
4		16
5		25

COUNTING

1 2 3

1	2	3
10^2	10^1	10^0

$$\begin{array}{ccc} 1 & 2 & 3 \\ \hline 10^2 & 10^1 & 10^0 \end{array}$$

$$= 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

$$= 1 \times 100 + 2 \times 10 + 3 \times 1$$

$$= 123$$

4	1	2	3
?	10^2	10^1	10^0

4	1	2	3
?	10^2	10^1	10^0

$$= 4 \times 10^3 + 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

4	1	2	3
?	10^2	10^1	10^0

$$= 4 \times 10^3 + 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

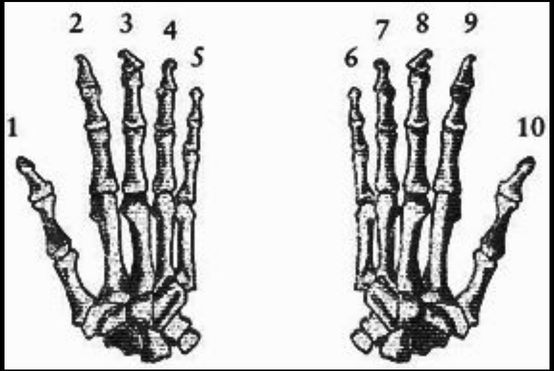
$$= 4 \times 1000 + 1 \times 100 + 2 \times 10 + 3 \times 1$$

4	1	2	3
?	10^2	10^1	10^0

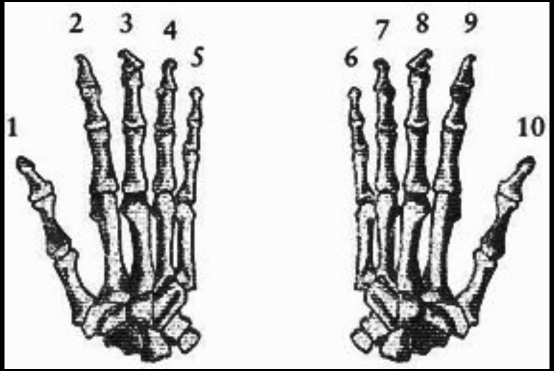
$$= 4 \times 10^3 + 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

$$= 4 \times 1000 + 1 \times 100 + 2 \times 10 + 3 \times 1$$

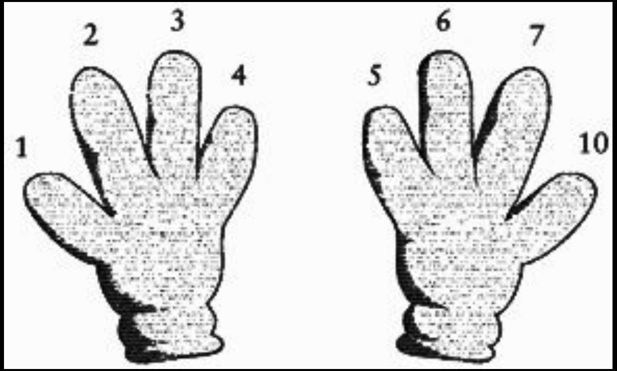
$$= 4123$$



Human Hand



Human Hand



Cartoon Character’s Hand

1 **2** **3** (octal)

1	**2**	**3**	(octal)
8^2	8^1	8^0	

$$\begin{array}{ccc} 1 & 2 & 3 \\ \hline 8^2 & 8^1 & 8^0 \end{array}$$ (octal)

$$= 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$$

$$\begin{array}{ccc} 1 & 2 & 3 \\ \hline 8^2 & 8^1 & 8^0 \end{array}$$

(octal)

$$= 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$$

$$= 1 \times 64 + 2 \times 8 + 3 \times 1$$

$$\begin{array}{ccc} 1 & 2 & 3 \\ \hline 8^2 & 8^1 & 8^0 \end{array}$$ (octal)

$= 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$

$= 1 \times 64 + 2 \times 8 + 3 \times 1$

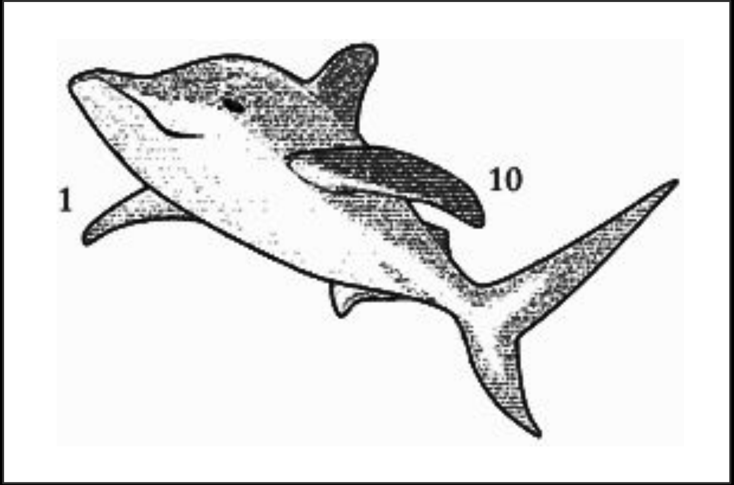
$= 83$ (decimal)

$$\begin{tikzcd}
\text{decimal} & \text{octal} \\
8 \arrow[r] & ?
\end{tikzcd}$$

$$\begin{tikzcd} \text{decimal} & \text{octal} \\ ? & 7 \arrow[l] \end{tikzcd}$$

$$\begin{tikzcd}
\text{decimal} & \text{octal} \\
16 \arrow[r] & ?
\end{tikzcd}$$

$$\begin{tikzcd} \text{decimal} & & \text{octal} \\ ? & & 100 \arrow[ll] \end{tikzcd}$$



What now?

0, 1, …

0, 1, 10, …

0, 1, 10, 11, …

0, 1, 10, 11, 100, …

0, 1, 10, 11, 100, 101, …

0, 1, 10, 11, 100, 101, 110

1 1 0 (binary)

1	1	0	(binary)
2^2	2^1	2^0	

$$\begin{array}{ccc} 1 & 1 & 0 \\ \hline 2^2 & 2^1 & 2^0 \end{array}$$ (binary)

$$= 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

1 1 0 (binary)

2^2 2^1 2^0

$$= 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 1 \times 4 + 1 \times 2 + 0 \times 1$$

$$1 \quad 1 \quad 0$$ (binary)

$$2^2 \quad 2^1 \quad 2^0$$

$$= 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 1 \times 4 + 1 \times 2 + 0 \times 1$$

$= 6$ (decimal)

$$\begin{array}{ccccccc} & & 2 & 3 & 4 & 5 & 6 \\ 0, & 1, & 10, & 11, & 100, & 101, & 110 \end{array}$$

Place Value Systems

$$N = d_n * R^{n-1} + \ldots + d_1 * R^1 + d_0 * R^0$$

$d \in \{ 0, 1, \ldots R\text{-}1 \}$

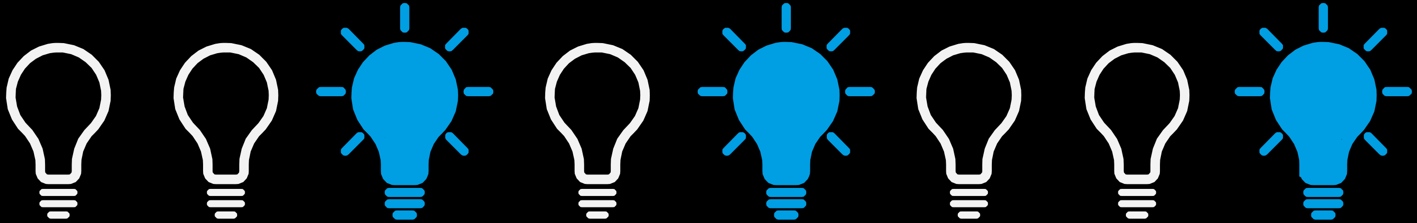
n = Number of digits

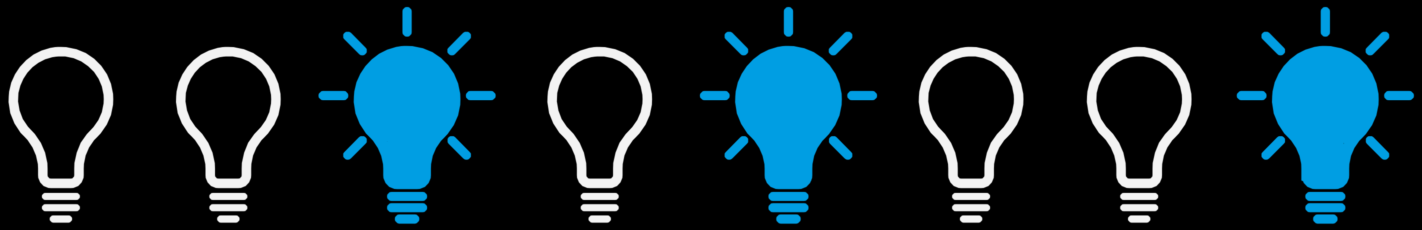
Place Value Systems

$$R \geq 2$$

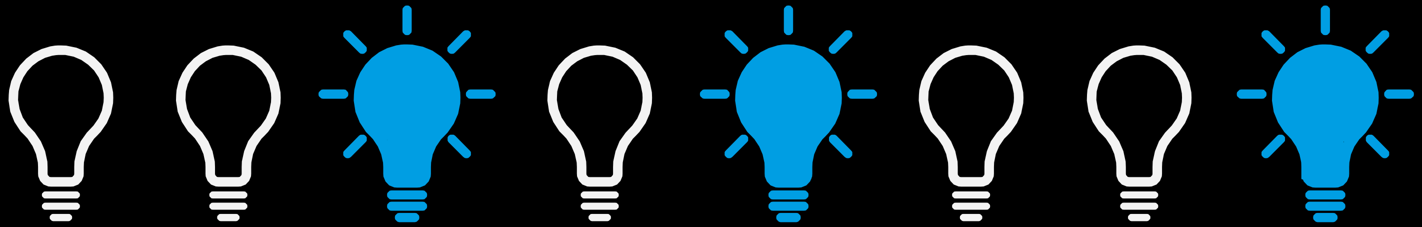
BITS

Why do computers think binary?





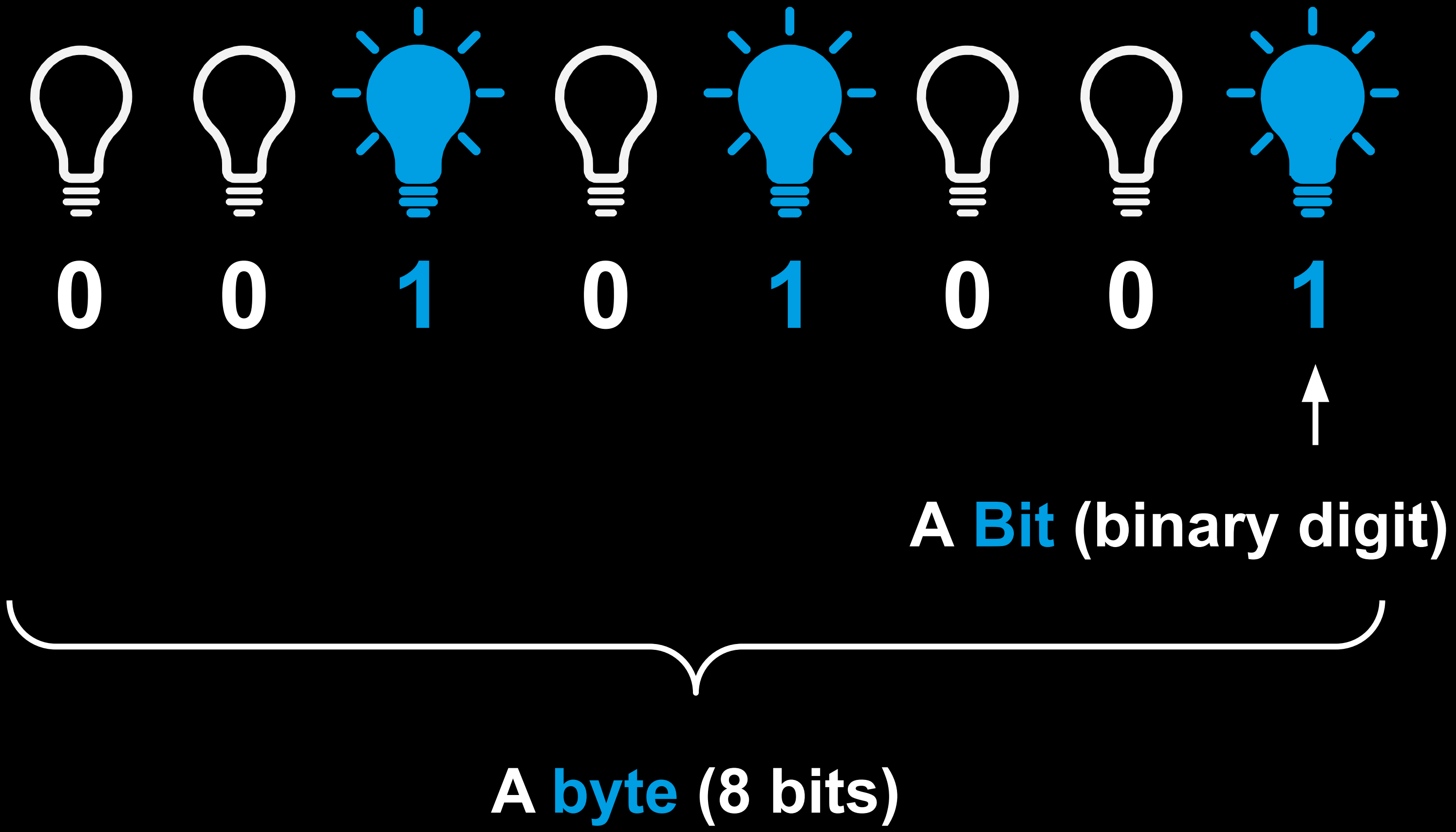
0 0 1 0 1 0 0 1

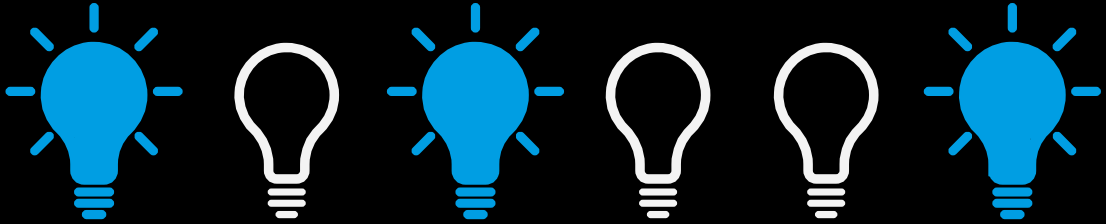


0 0 1 0 1 0 0 1

↑

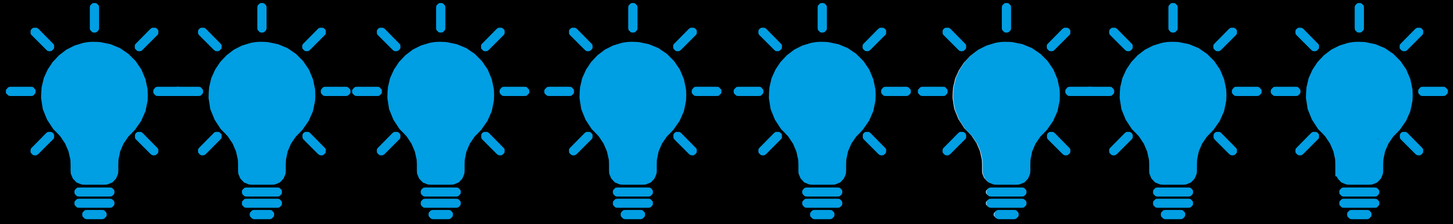
A **Bit** (binary digit)


0 0 1 0 1 0 0 1
A Bit (binary digit)
A byte (8 bits)



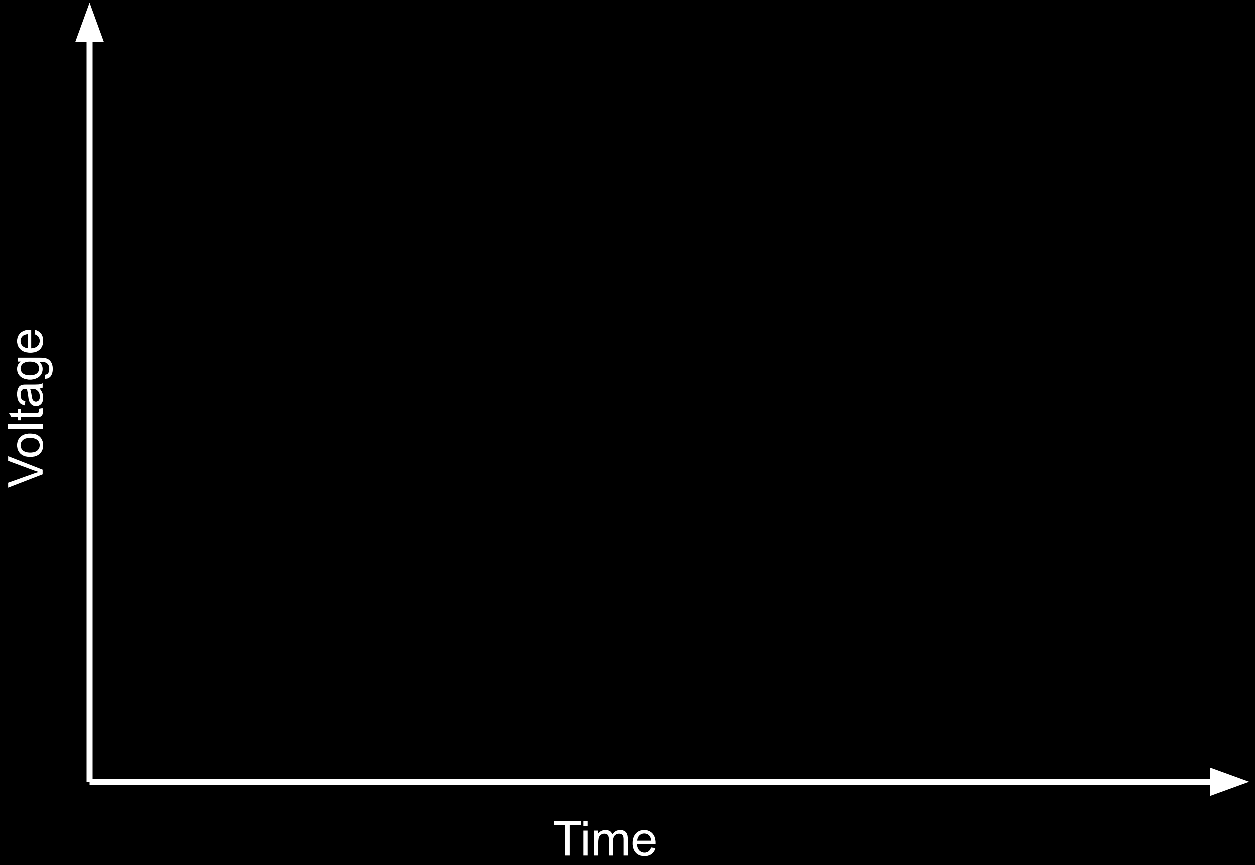
0	0	1	0	1	0	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

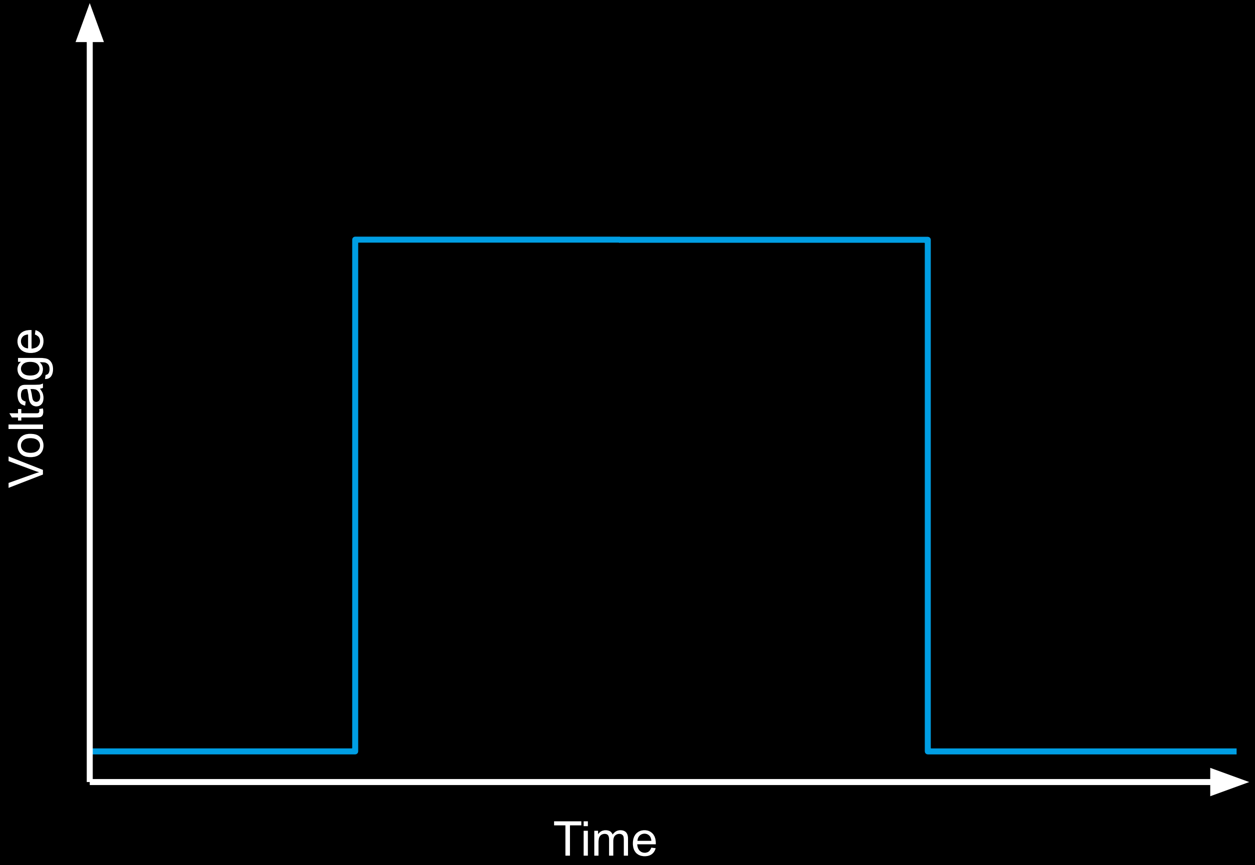
0	0	1	0	1	0	0	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1



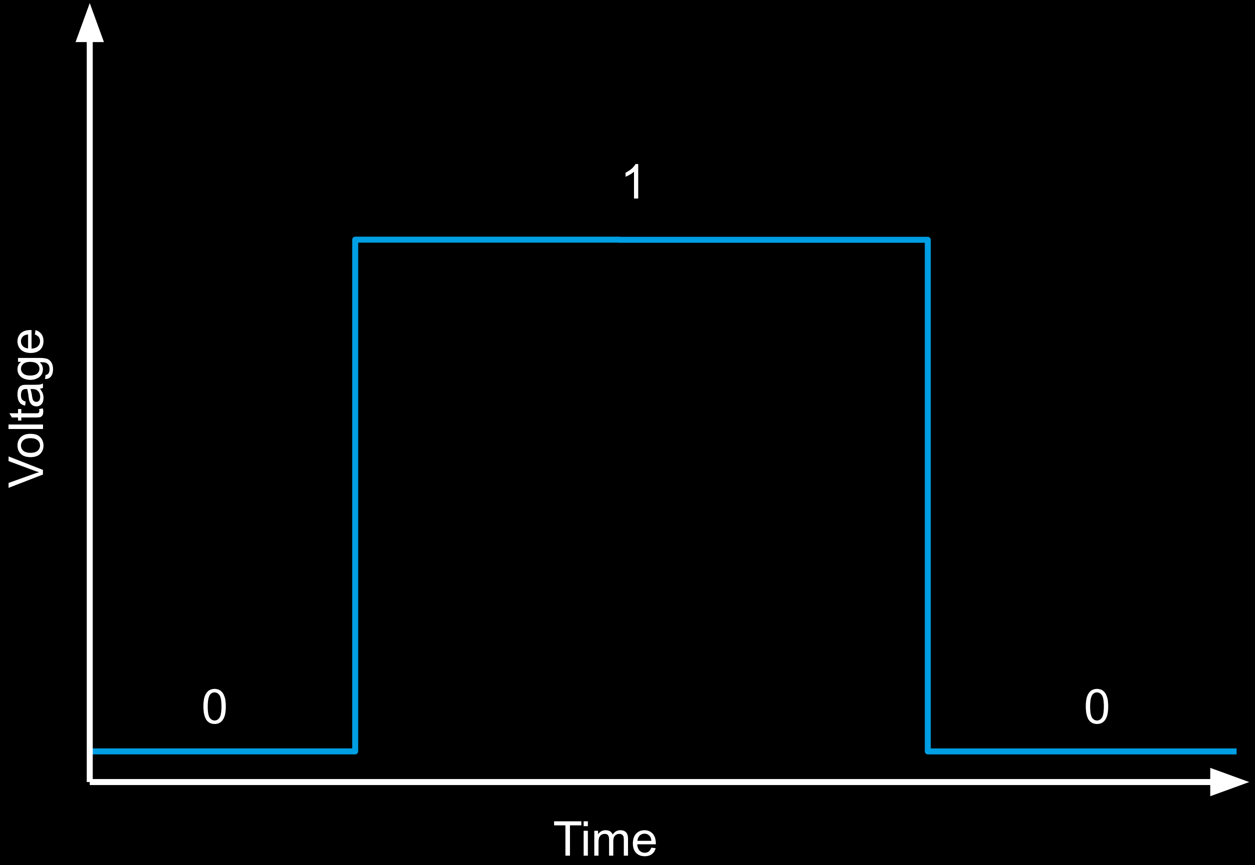
What can we store in one byte?

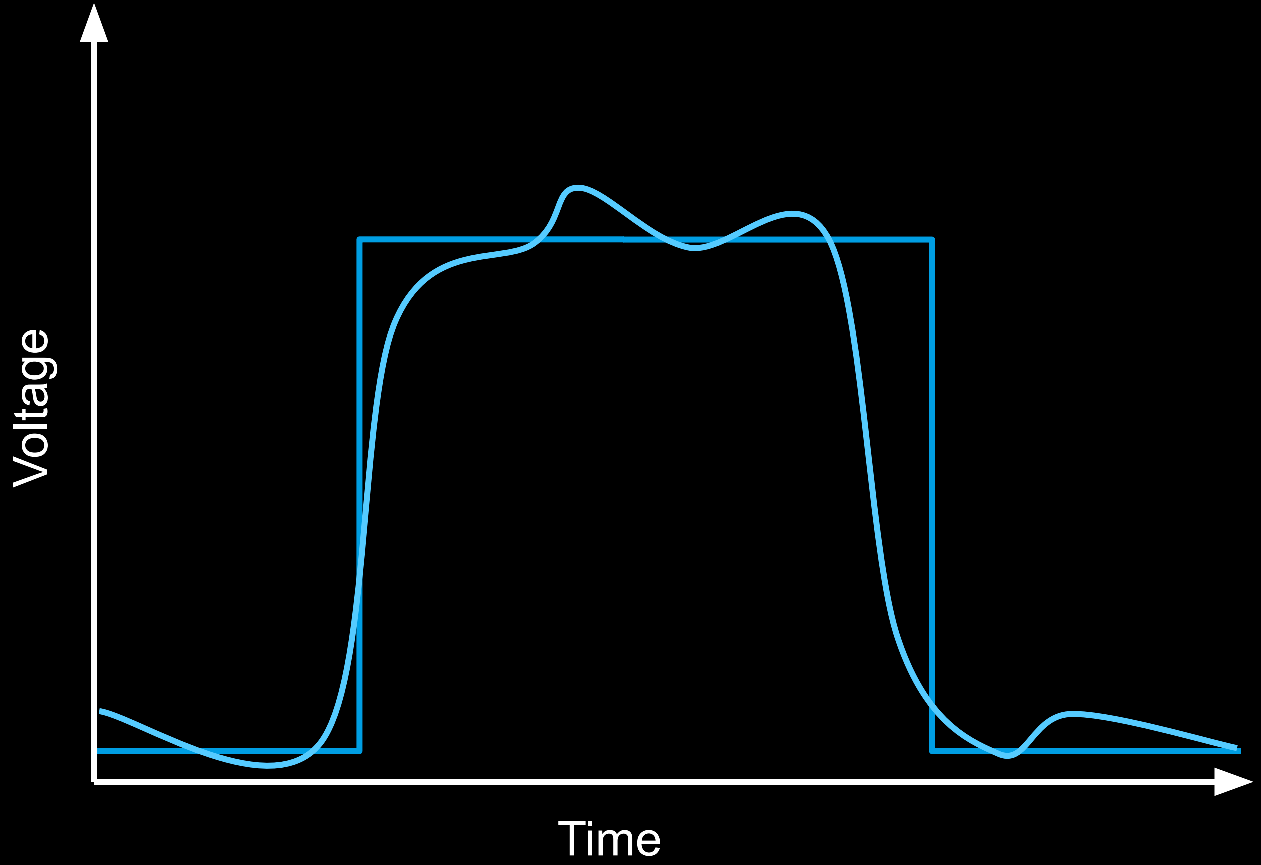
Are we stuck with binary?

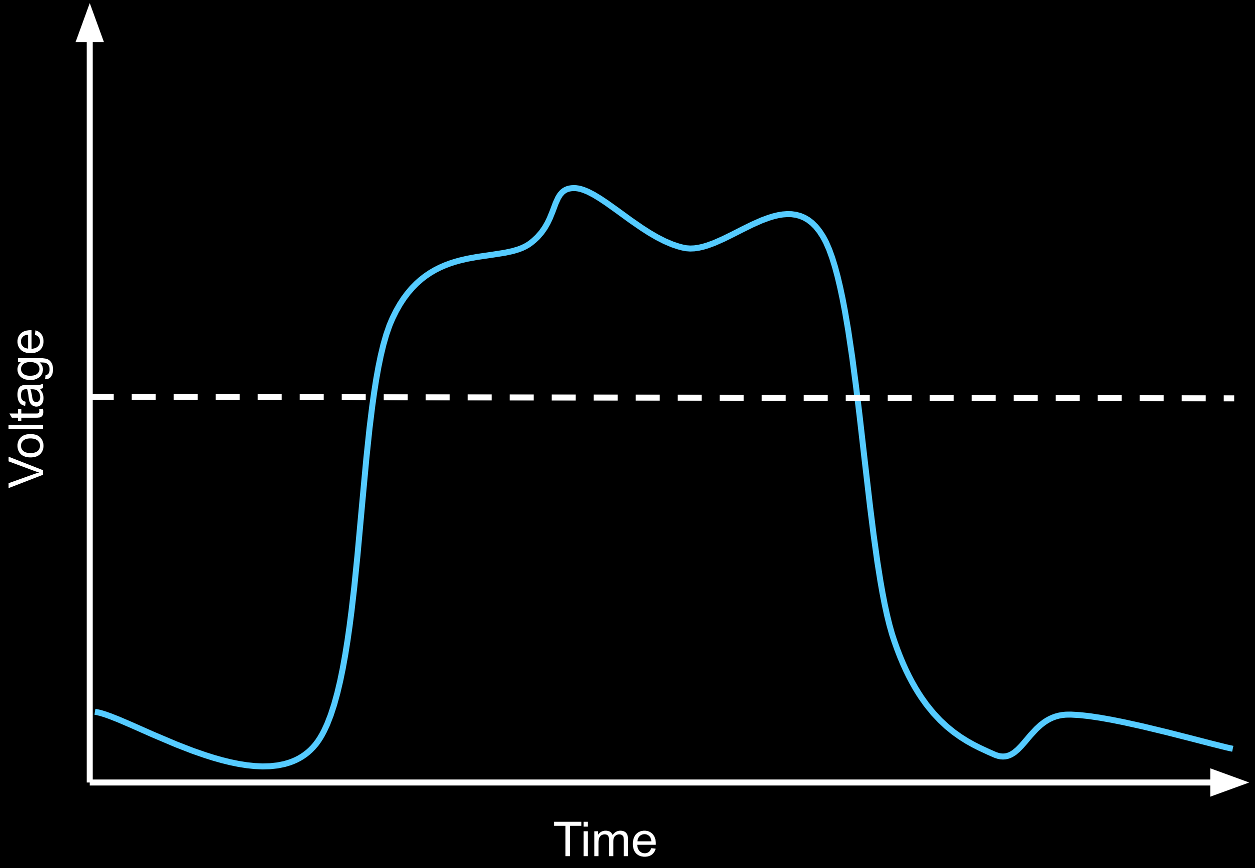

Voltage
Time

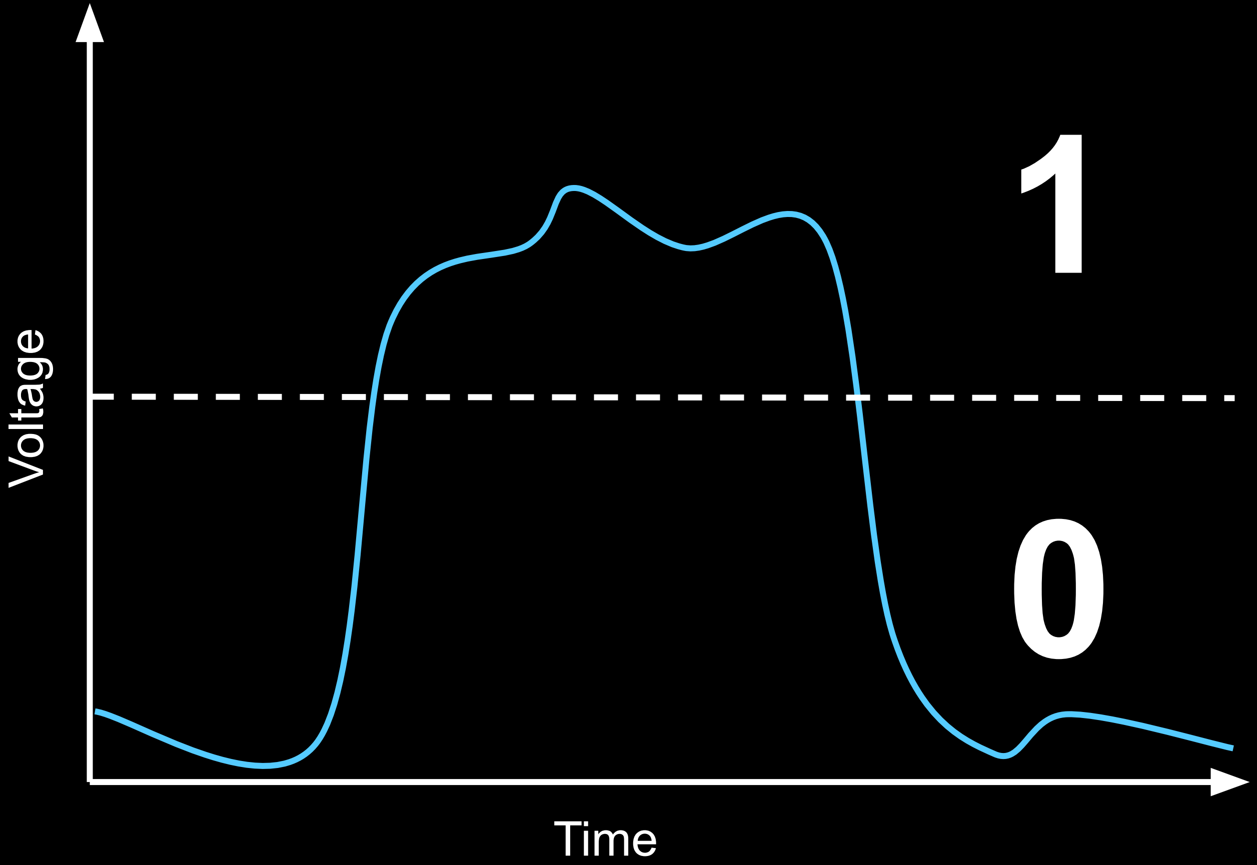


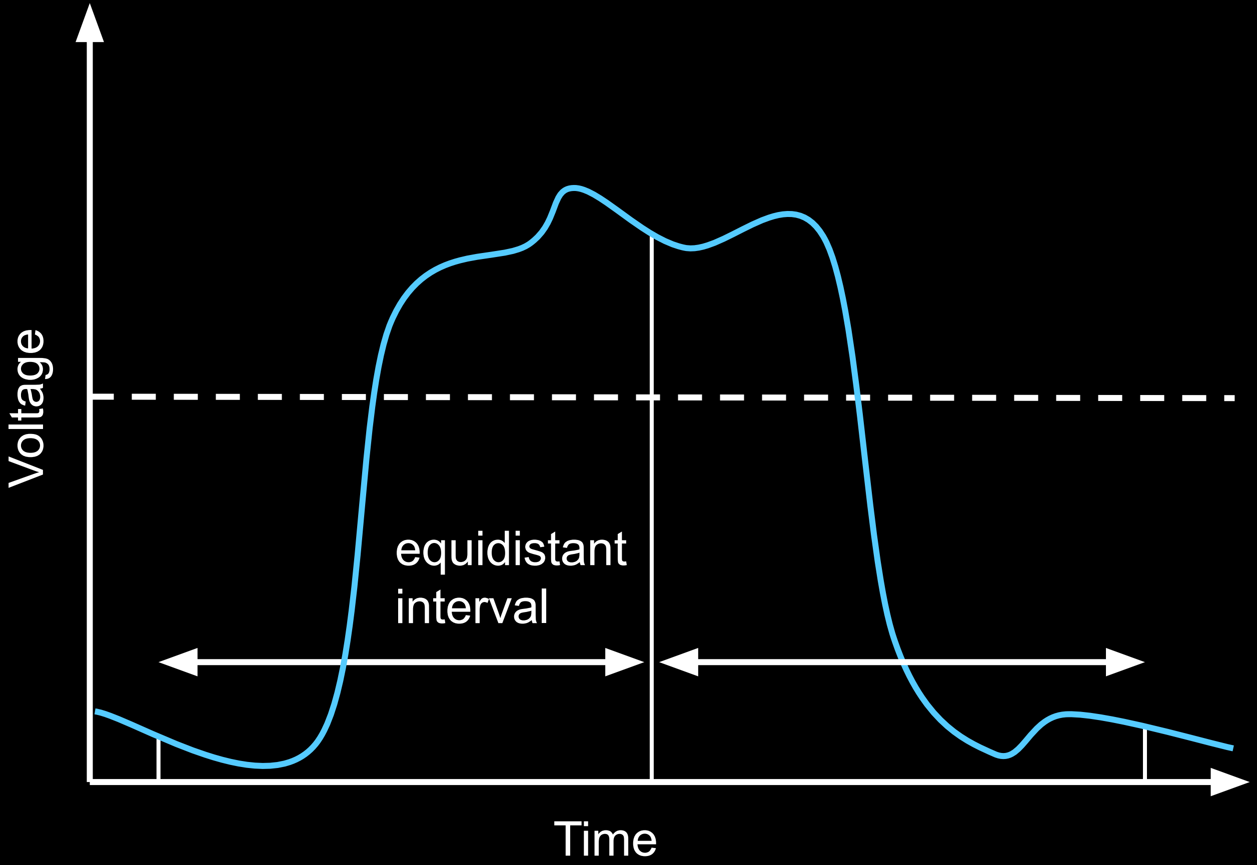
Voltage
Time

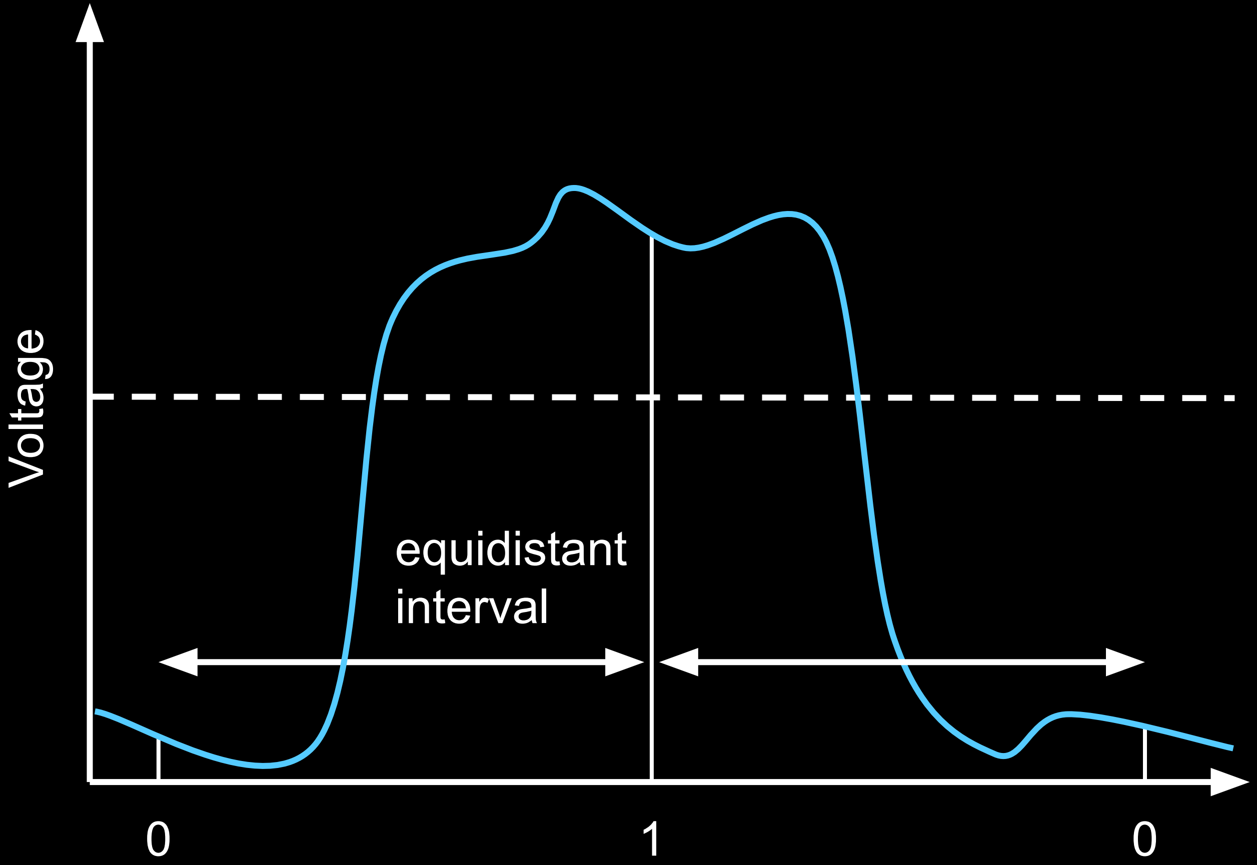

Voltage
1
0
0
Time


Voltage
Time

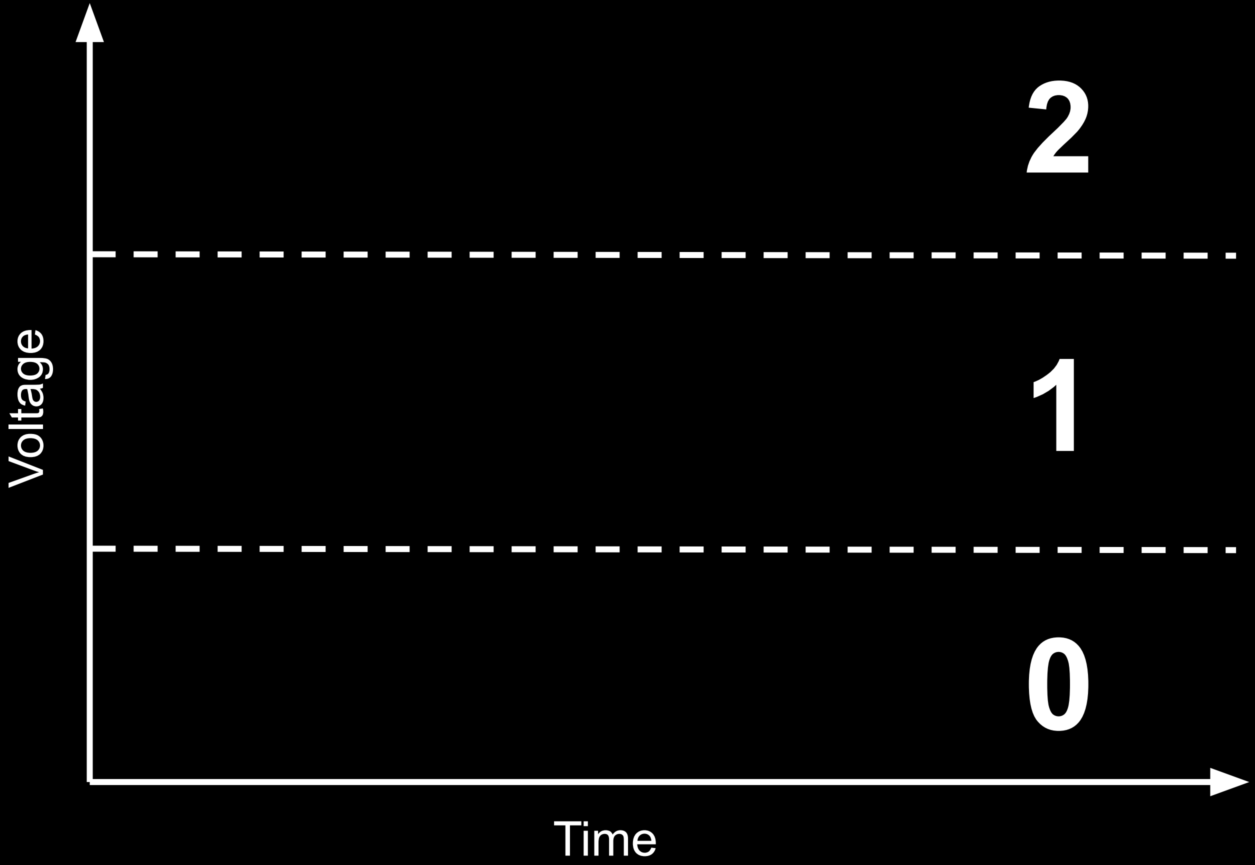

Voltage
Time

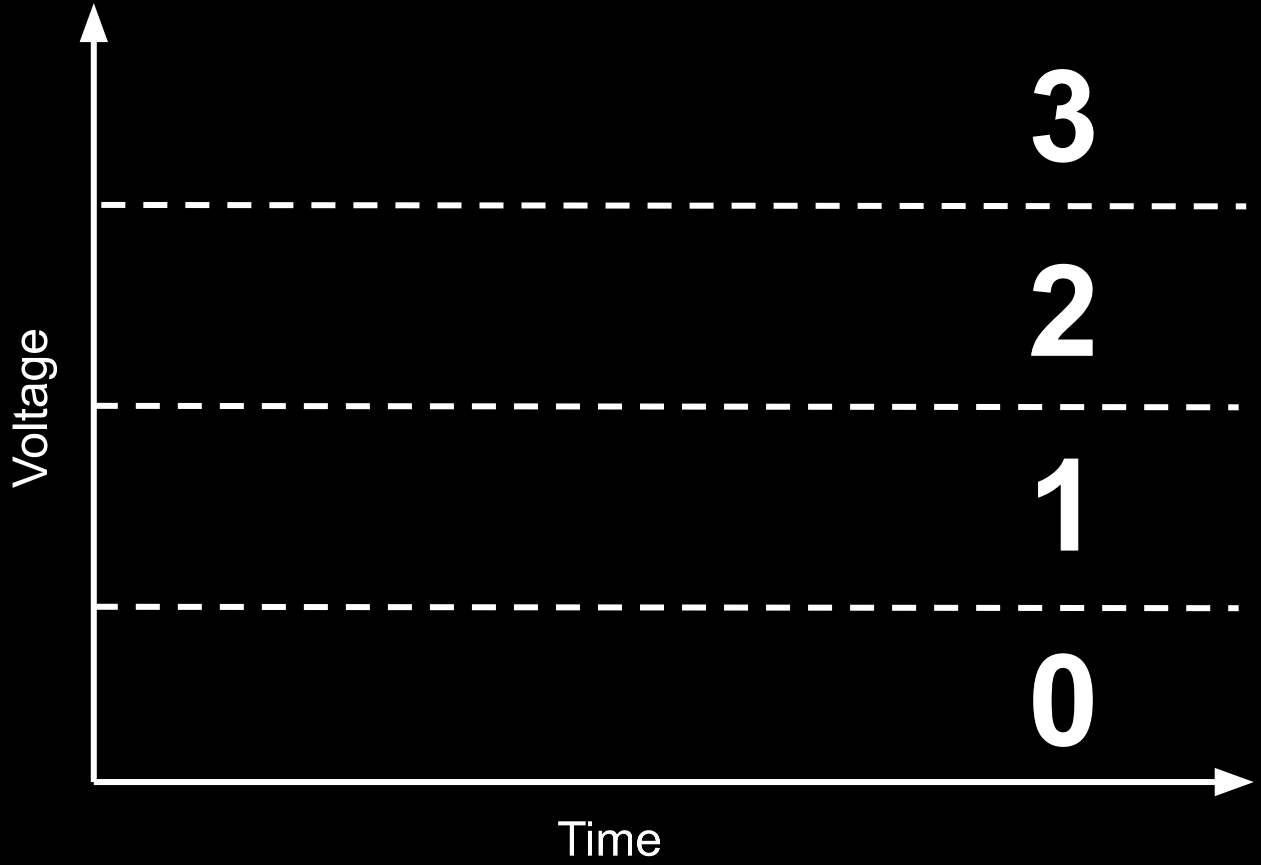

Voltage
1
0
Time

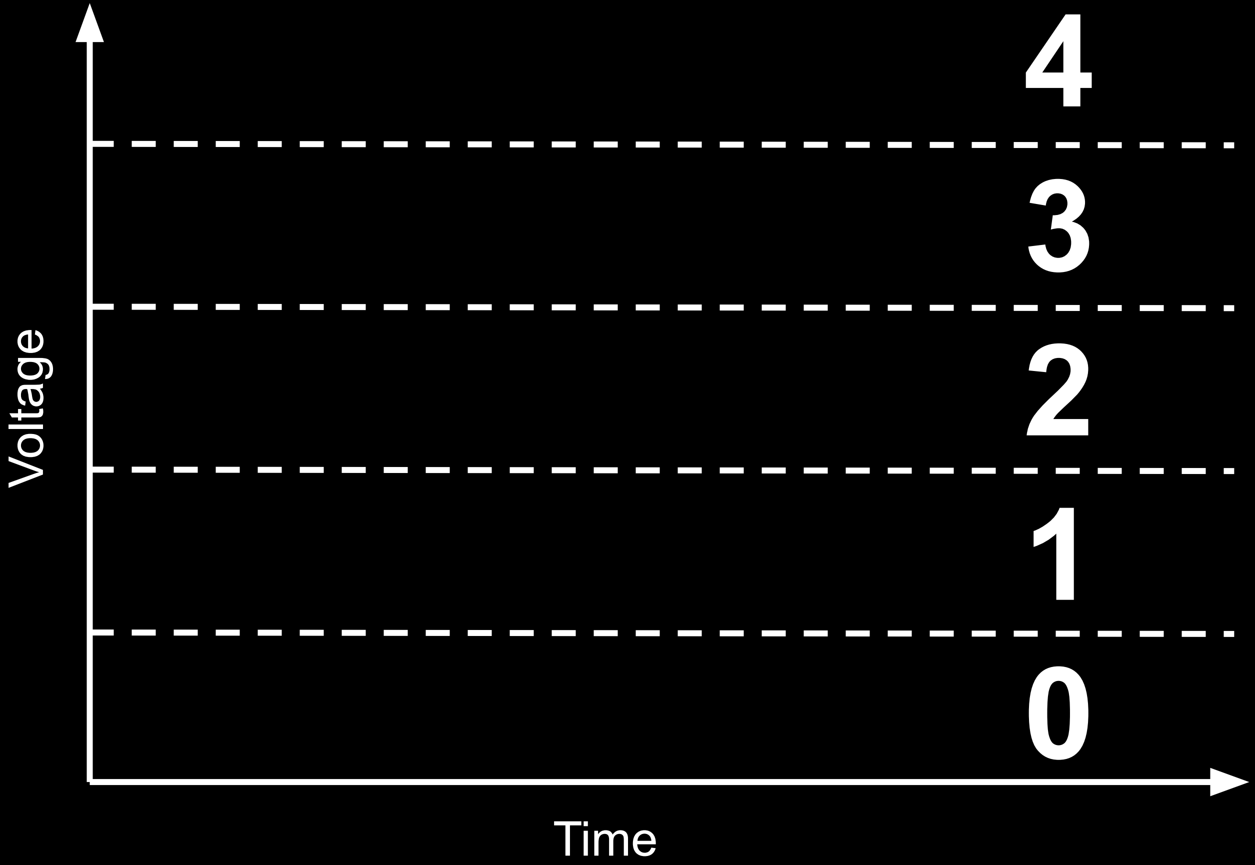

Voltage
equidistant
interval
Time

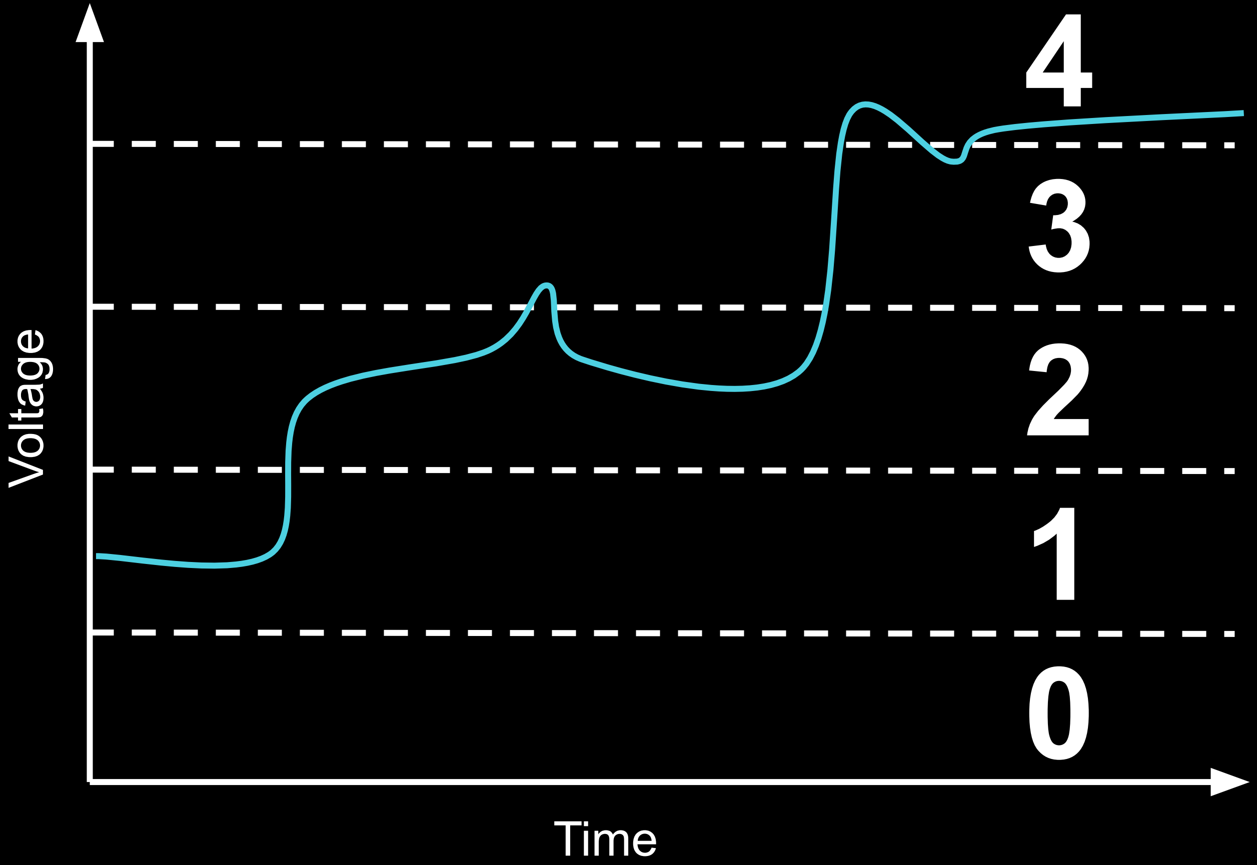

Voltage
equidistant interval
0
1
0

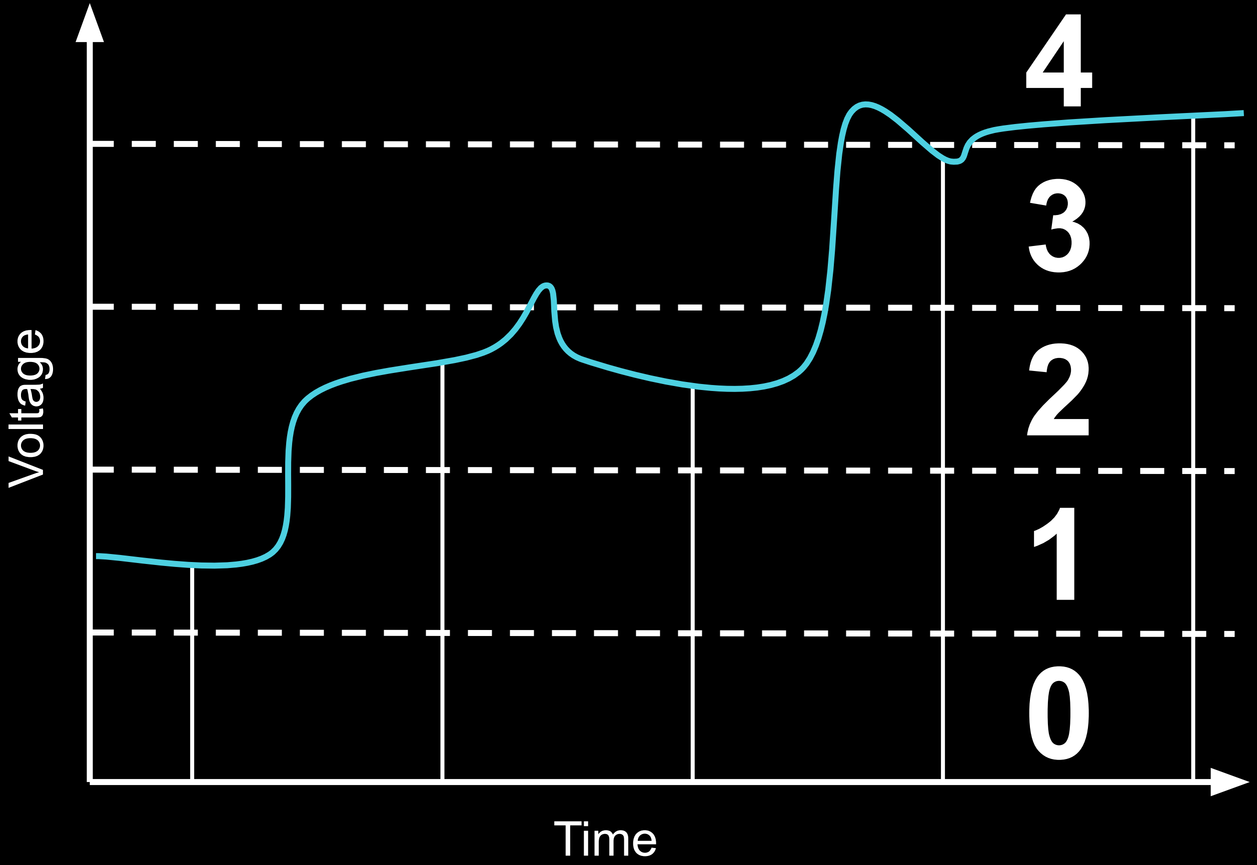
What about ternary?

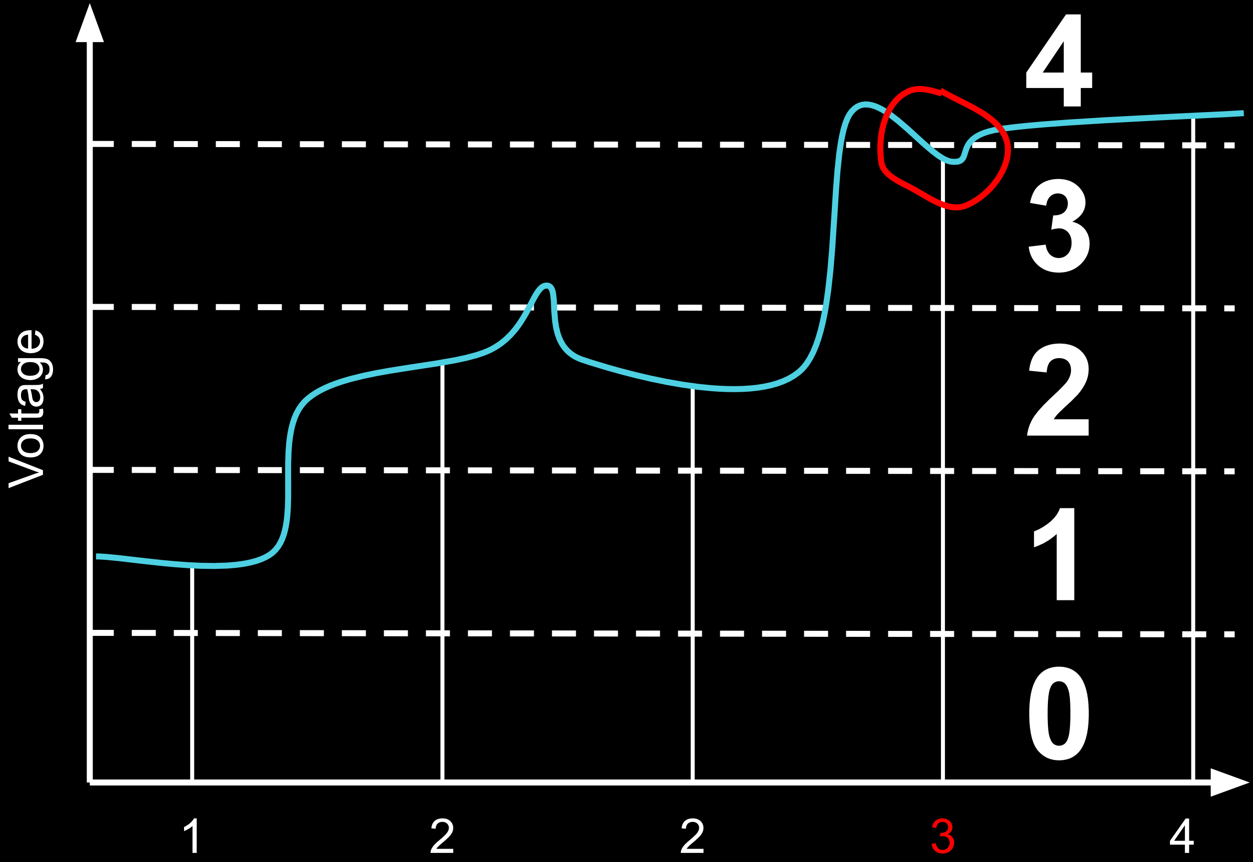

2
1
0
Voltage
Time


3
2
1
0
Voltage
Time


4
3
2
1
0
Voltage
Time

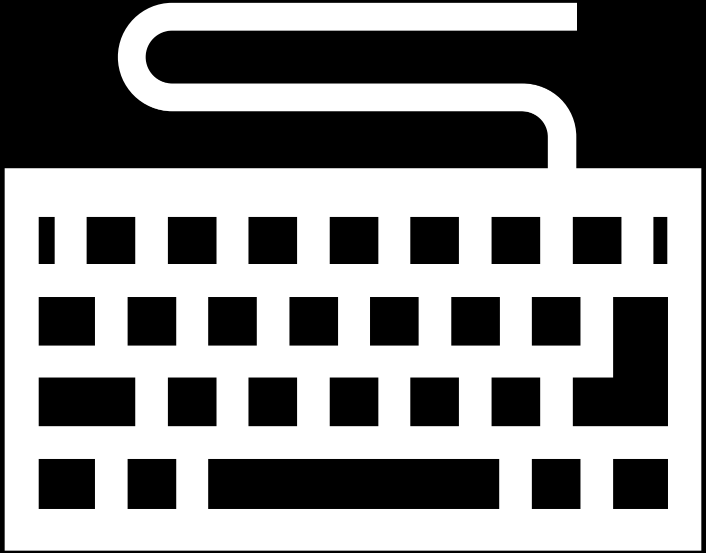

4
3
2
1
0
Voltage
Time


Voltage
Time
4
3
2
1
0


Voltage
4
3
2
1
0
1
2
2
3
4

CODES

● ● ● ▬ ▬ ▬ ● ● ●



A	B	C	D	...	a	b	c	d
65	66	67	68		97	98	99	100

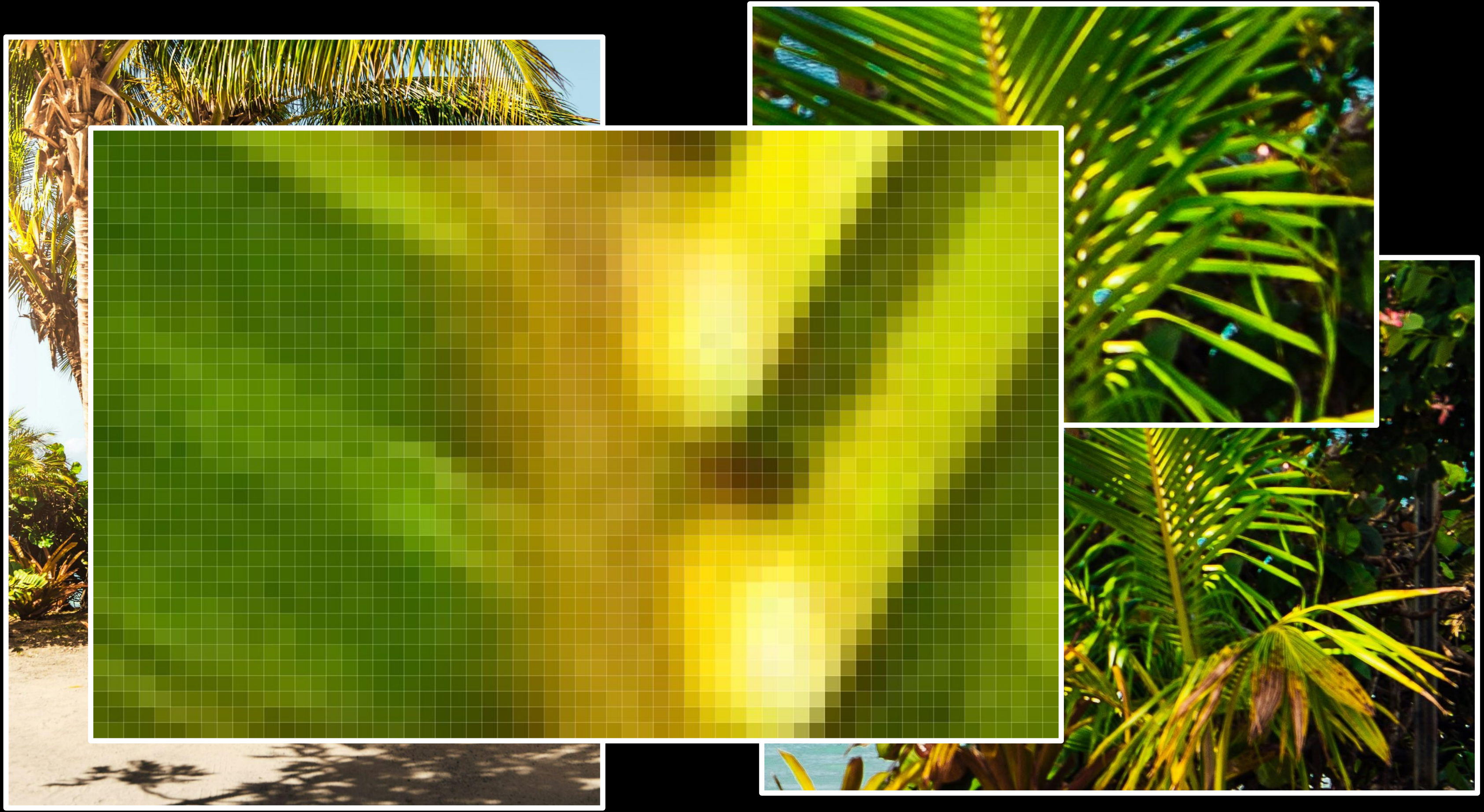
ASCII Code

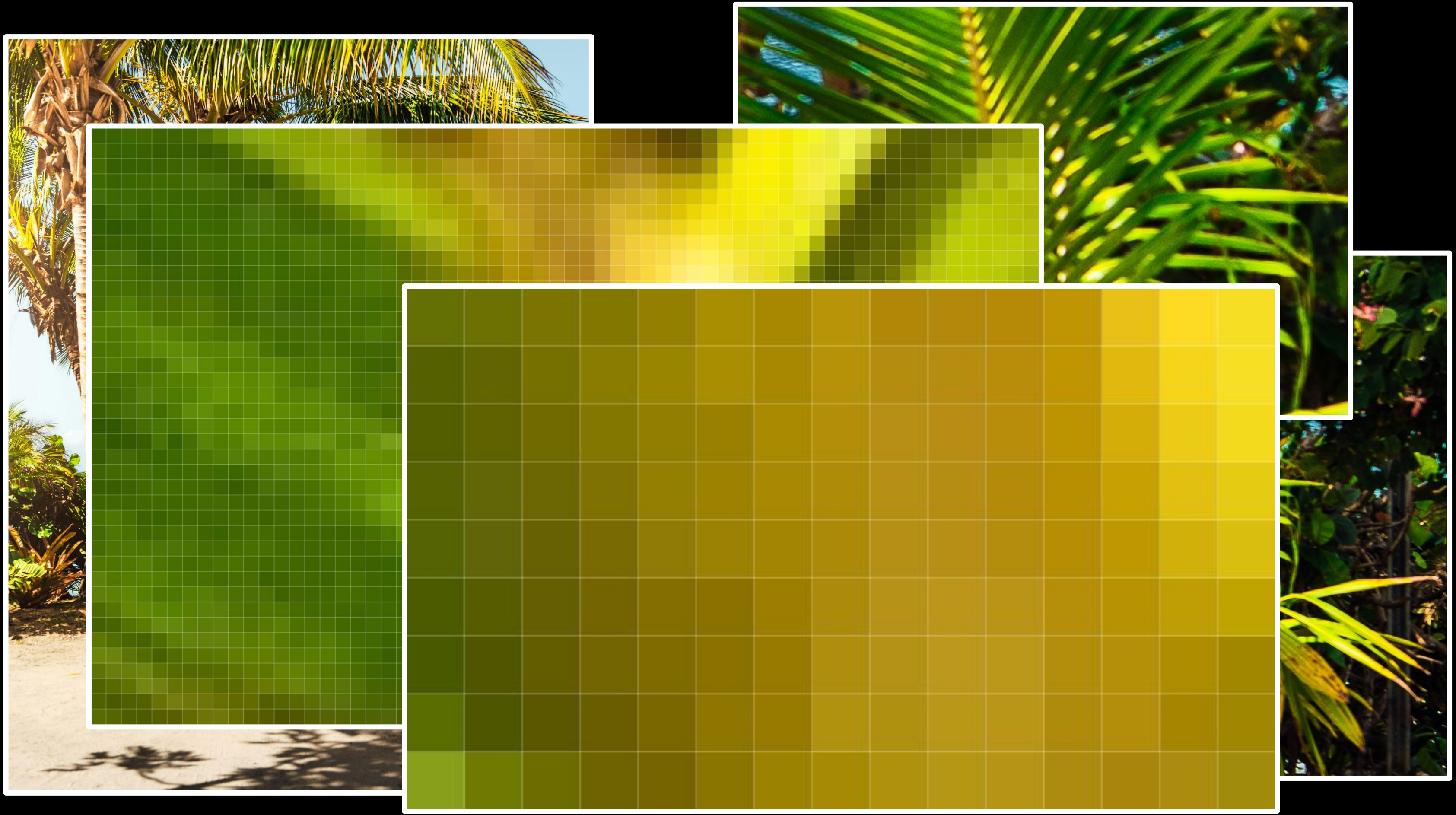
A	B	C	D	...	a	b	c	d
65	66	67	68		97	98	99	100

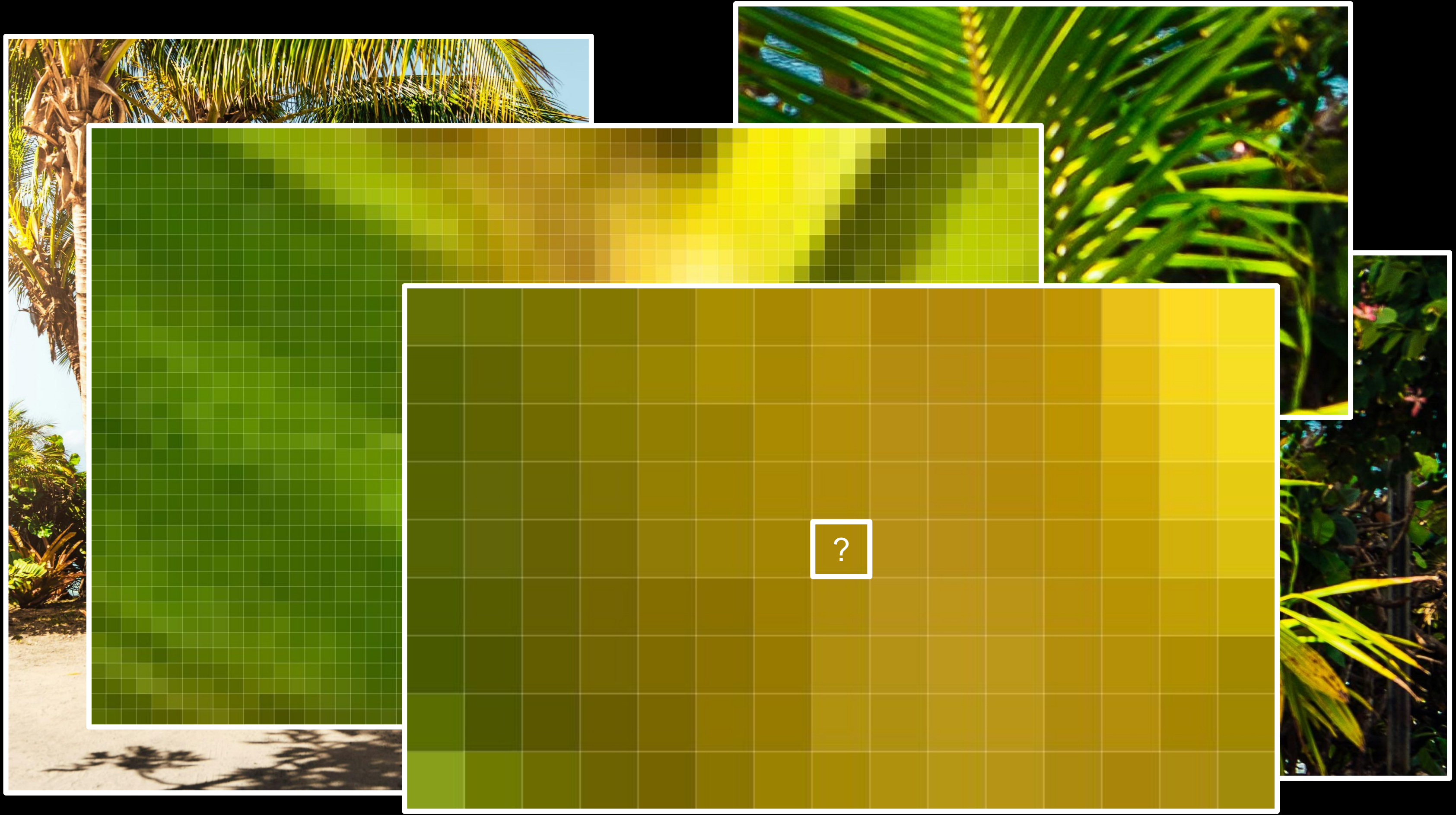
😀	😁	😂	😃	...	🙈	🙉	🙊	🙋
1F600	1F601	1F602	1F603		1F648	1F649	1F64A	1F64B

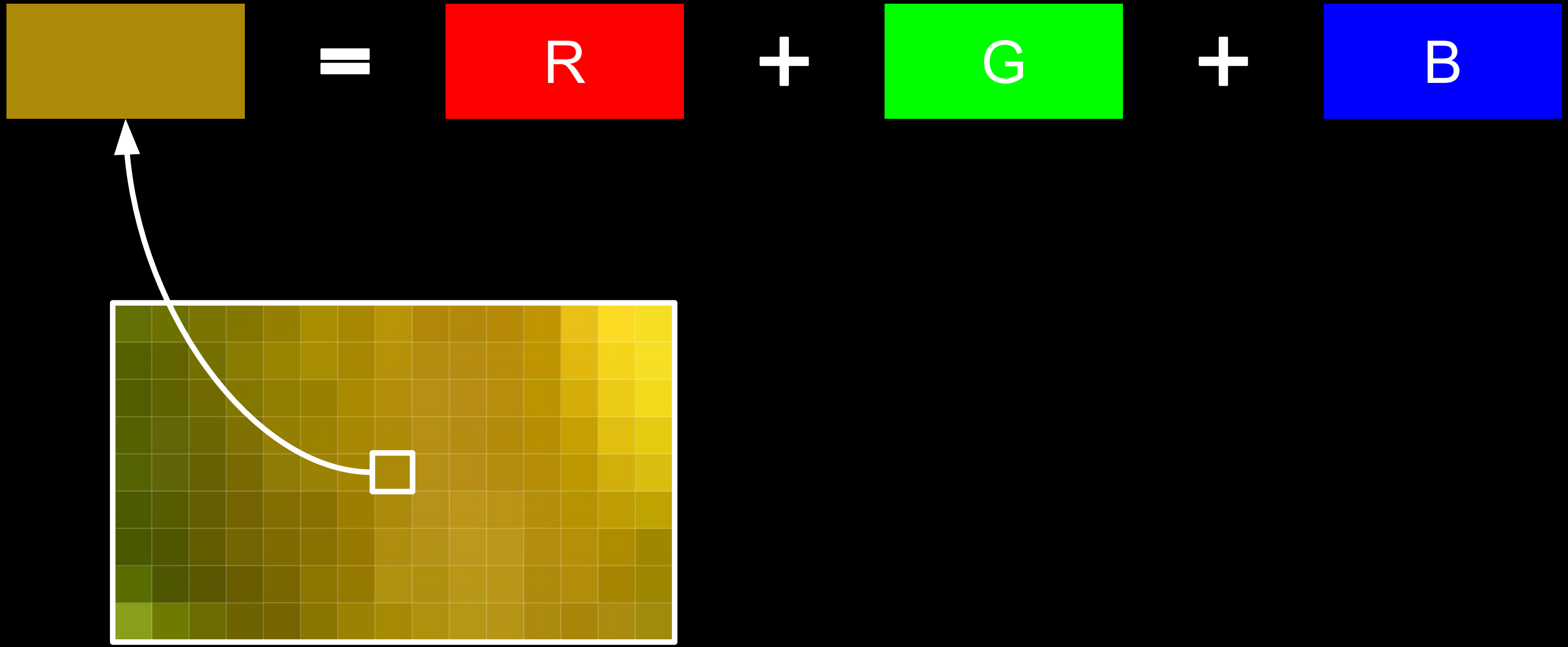
Unicode

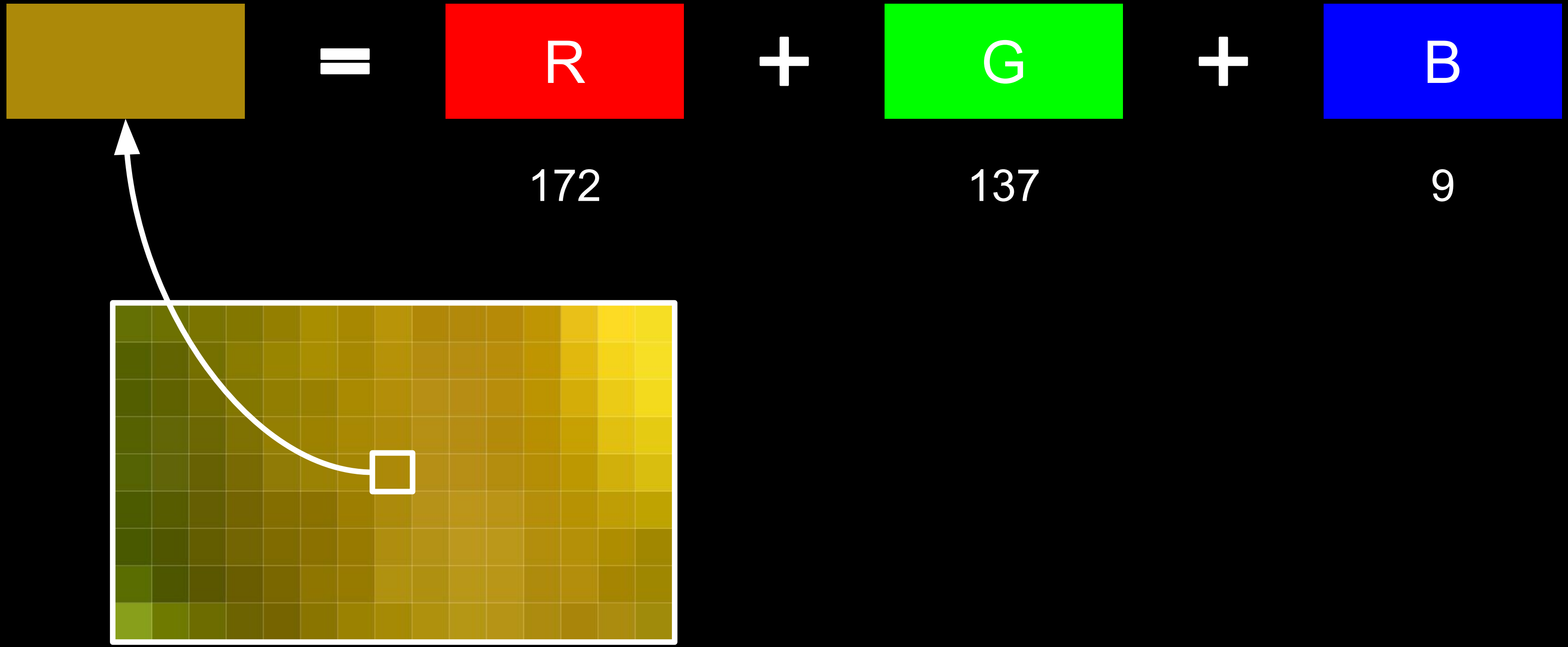
😀	😁	😂	😃	...	🙈	🙉	🙊	🙋‍♀️
1F600	1F601	1F602	1F603		1F648	1F649	1F64A	1F64B

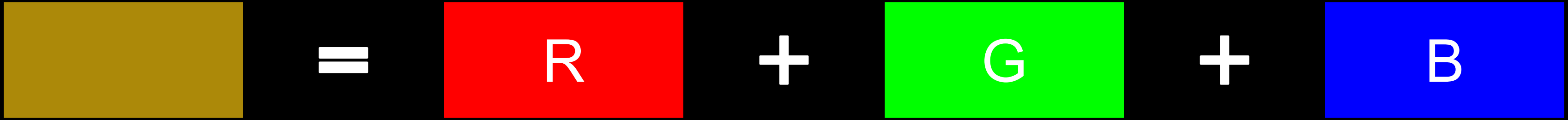






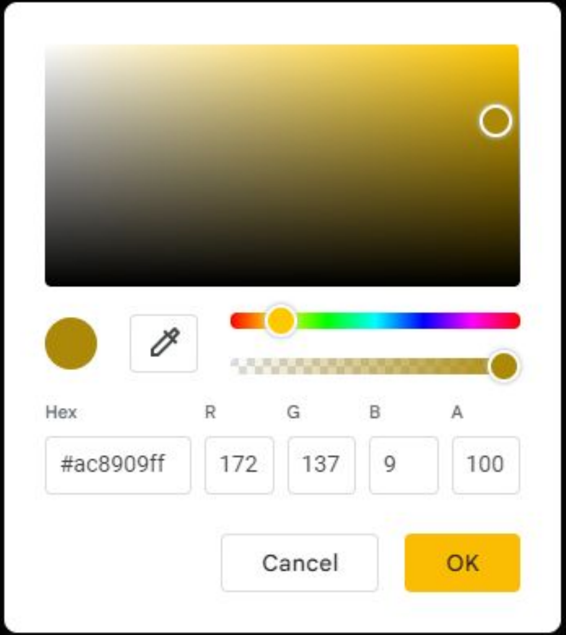

=
R
+
G
+
B


=
R
+
G
+
B
172
137
9



172 137 9

#AC8909



Hex: #ac8909ff
R: 172
G: 137
B: 9
A: 100

🟫	=	🟥 R	+	🟩 G	+	🟦 B
		172		137		9
#AC8909		AC		89		09

🟫	=	🟥 R	+	🟩 G	+	🟦 B
		172		137		9
#AC8909		AC		89		09
		10101100		01011001		00001001

# possible colors?

R

$2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

R	G	B
2^{23} 2^{22} 2^{21} 2^{20} 2^{19} 2^{18} 2^{17} 2^{16}	2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^{9} 2^{8}	2^{7} 2^{6} 2^{5} 2^{4} 2^{3} 2^{2} 2^{1} 2^{0}

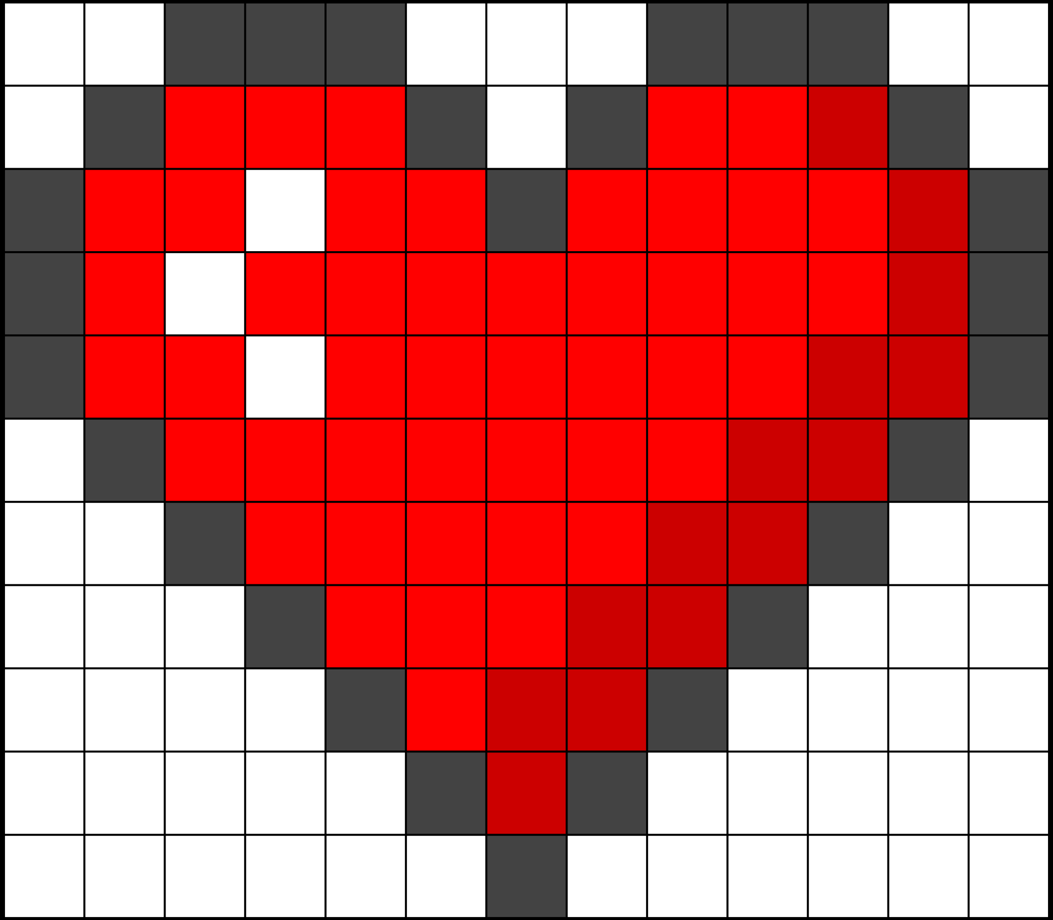
R G B

$$2^{23}\ \underbrace{2^{22}\ 2^{21}\ 2^{20}\ 2^{19}\ 2^{18}\ 2^{17}\ 2^{16}\quad 2^{15}\ 2^{14}\ 2^{13}\ 2^{12}\ 2^{11}\ 2^{10}\ 2^{9}\ 2^{8}\quad 2^{7}\ 2^{6}\ 2^{5}\ 2^{4}\ 2^{3}\ 2^{2}\ 2^{1}\ 2^{0}}$$

$$8.388.608 + 8.388.607 = 16.777.215$$

R G B

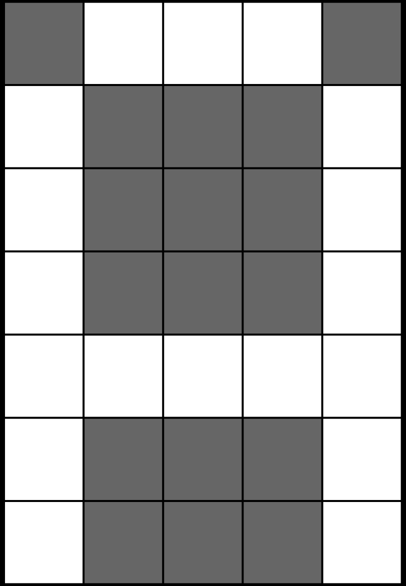
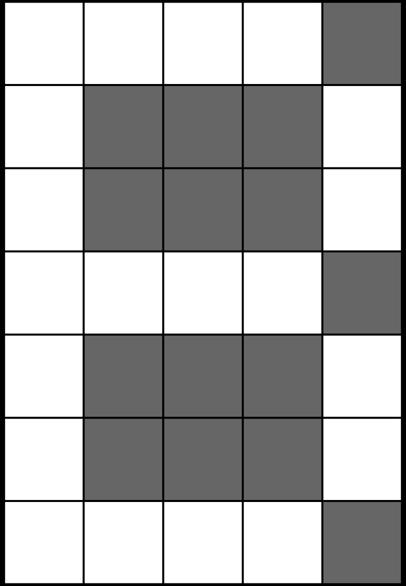
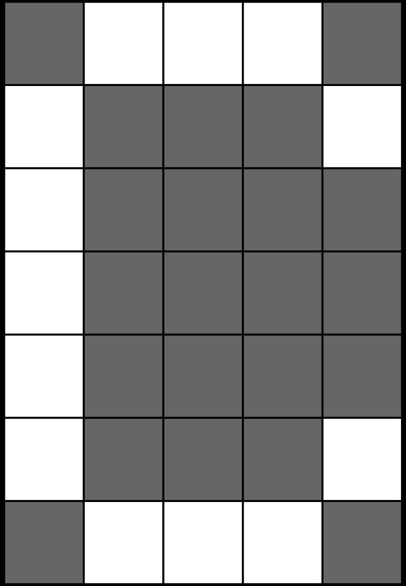
$$\underbrace{2^{23}\ 2^{22}\ 2^{21}\ 2^{20}\ 2^{19}\ 2^{18}\ 2^{17}\ 2^{16}}_{256} \times \underbrace{2^{15}\ 2^{14}\ 2^{13}\ 2^{12}\ 2^{11}\ 2^{10}\ 2^{9}\ 2^{8}}_{256} \times \underbrace{2^{7}\ 2^{6}\ 2^{5}\ 2^{4}\ 2^{3}\ 2^{2}\ 2^{1}\ 2^{0}}_{256}$$

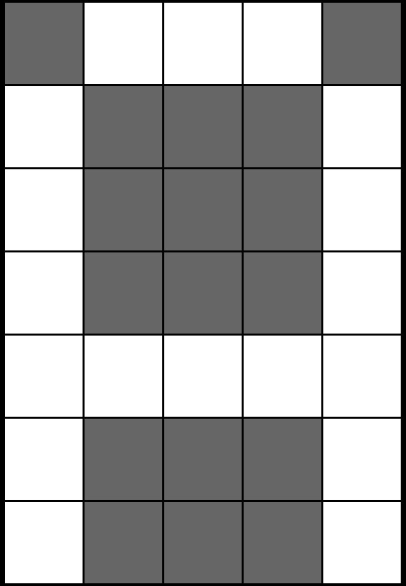


CS50 Pixel Art

🤔

compression?



0	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1

0	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1

0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0
0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1

5

7

0	1	1	1	0
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1

0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 0
0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 1