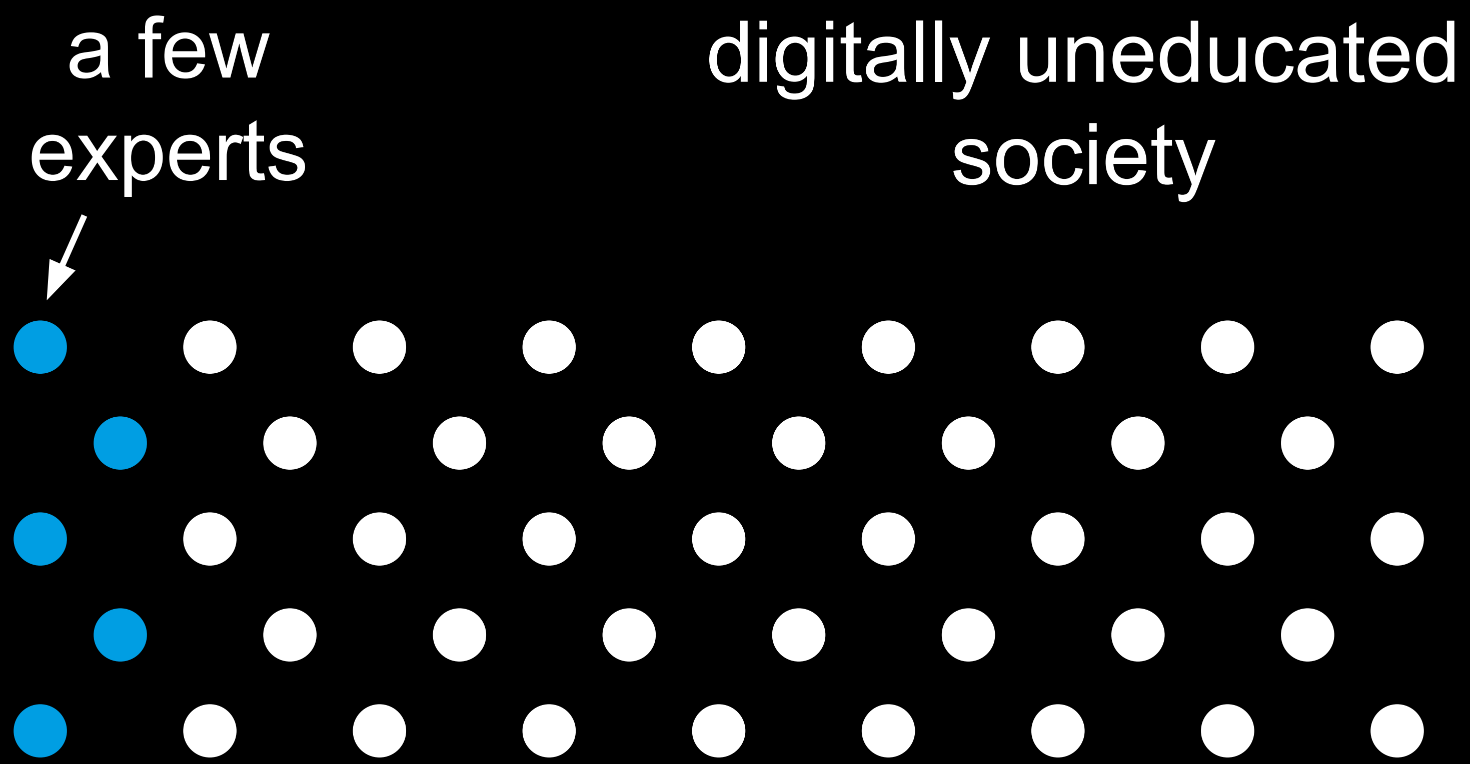
0. MOTIVATION
1. PROBLEM SOLVING
2. INFORMATION
3. COUNTING
4. BITS
5. CODES
6. ALGORITHMS
7. COMPUTERS
8. ARITHMETIC
9. MEMORY
10. ANALOG VS. DIGITAL

The slides are meant as visual support for the lecture.
They are neither a documentation nor a script.

Please do not print the slides.

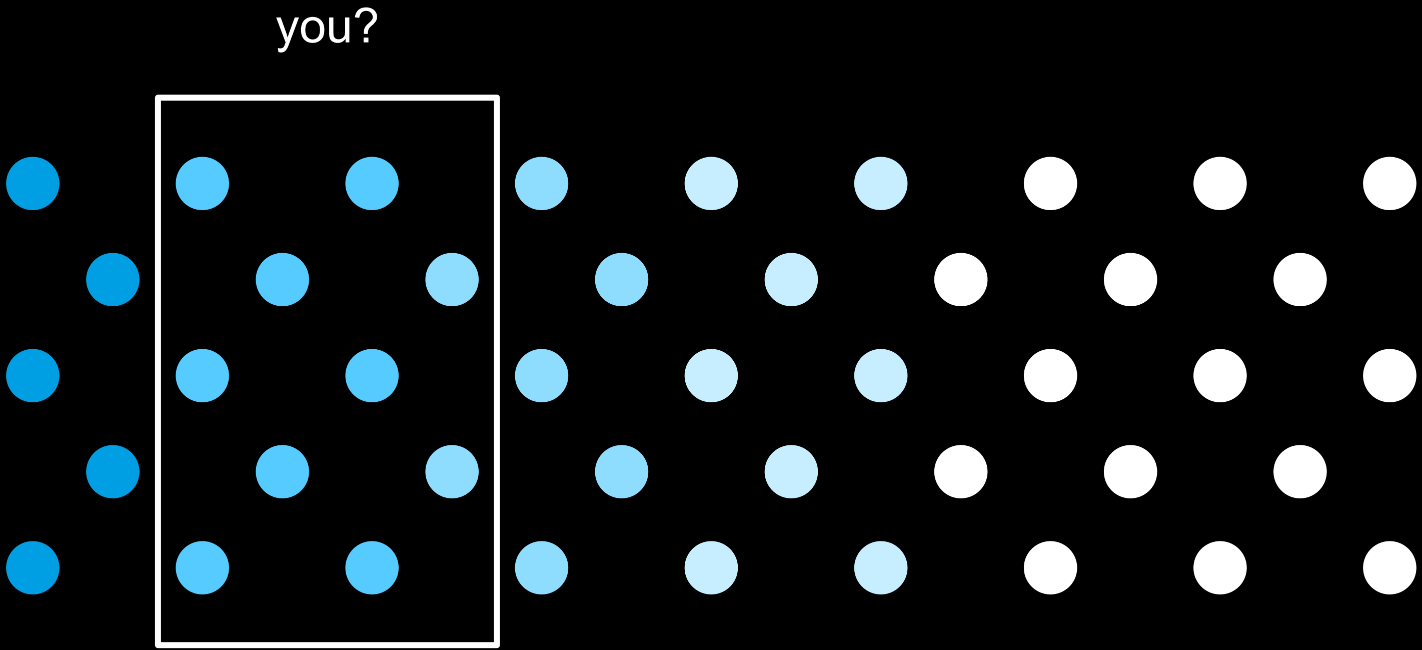
Comments and feedback at n.meseth@hs-osnabrueck.de

MOTIVATION



digitally illiterate society with a few experts

collective understanding



society with a distributed and high degree of digital education

representation

representation

processing

representation | processing | programming

representation

processing

programming

digital fundamentals

data analysis

representation

processing

programming

digital fundamentals

data analysis

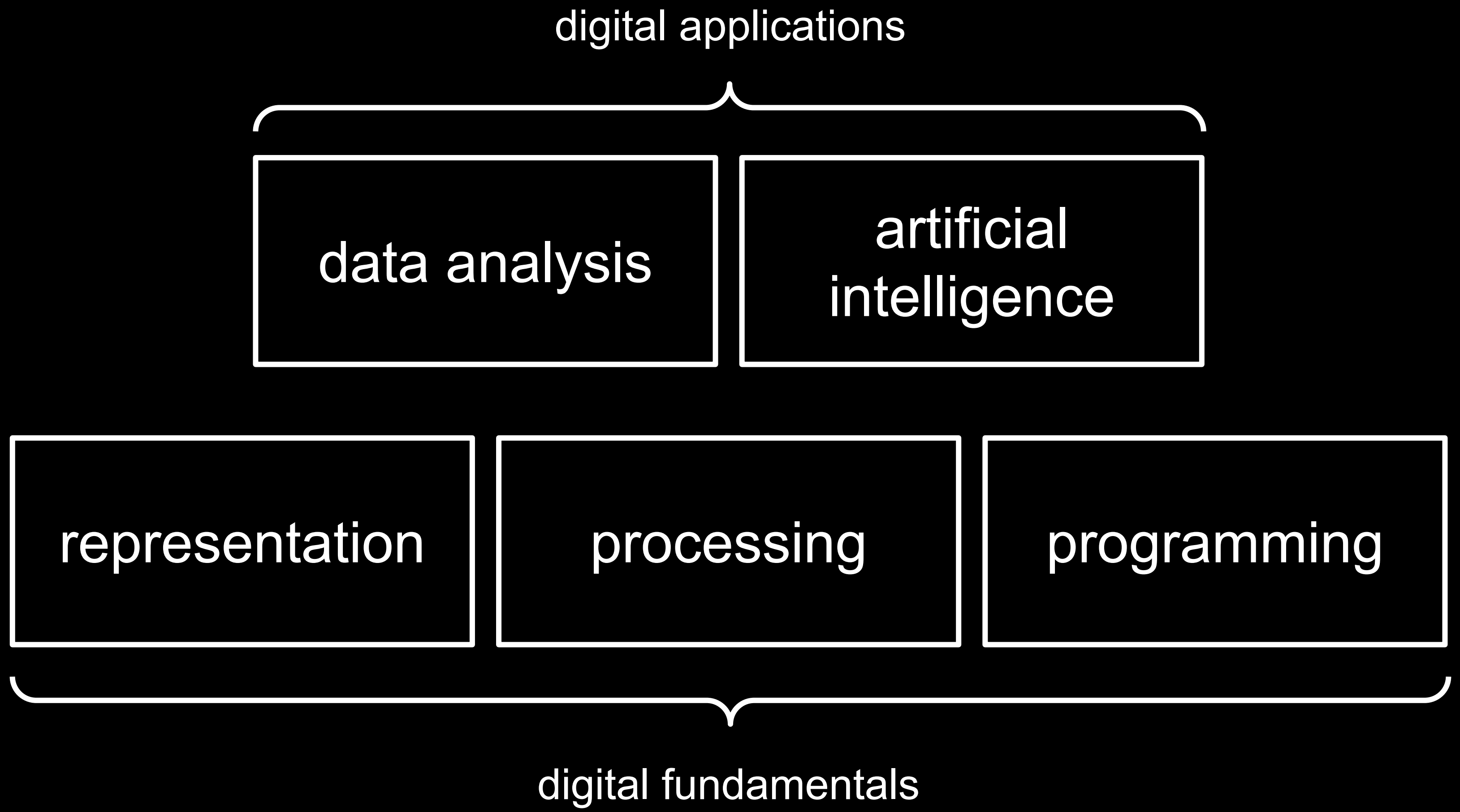
artificial intelligence

representation

processing

programming

digital fundamentals


digital applications
data analysis
artificial intelligence
representation
processing
programming
digital fundamentals

digital applications

data analysis

artificial intelligence

representation

processing

programming

digital fundamentals

digital applications

data analysis

artificial intelligence

representation

processing

programming

digital fundamentals

digital applications

data analysis

artificial intelligence

representation

processing

programming

digital fundamentals

PROBLEM SOLVING

BACK

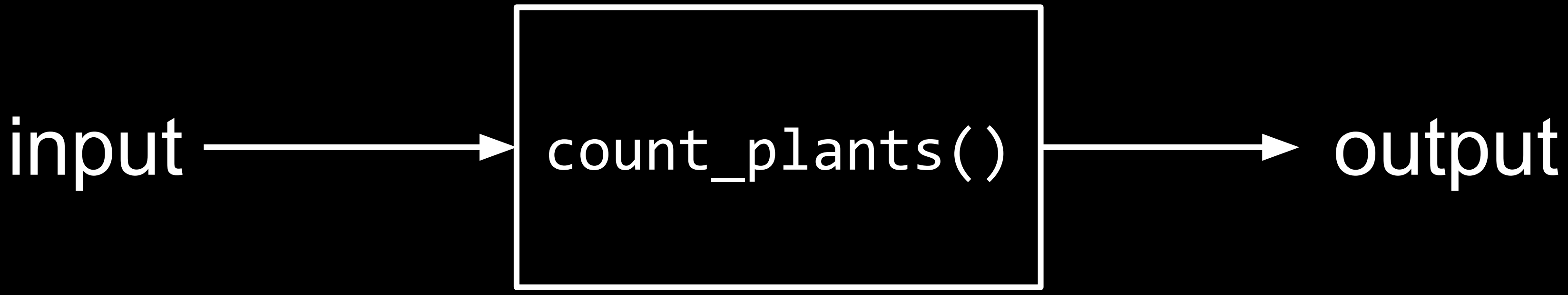
a model for solving problems

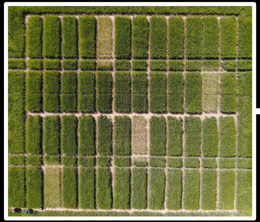
a model for solving problems

a model for solving problems

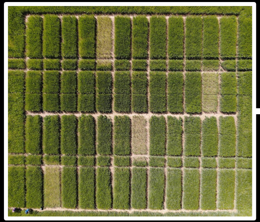
input → computation → output

$$\{a, b\} \longrightarrow \boxed{a + b} \longrightarrow s$$


input
count_plants()
output

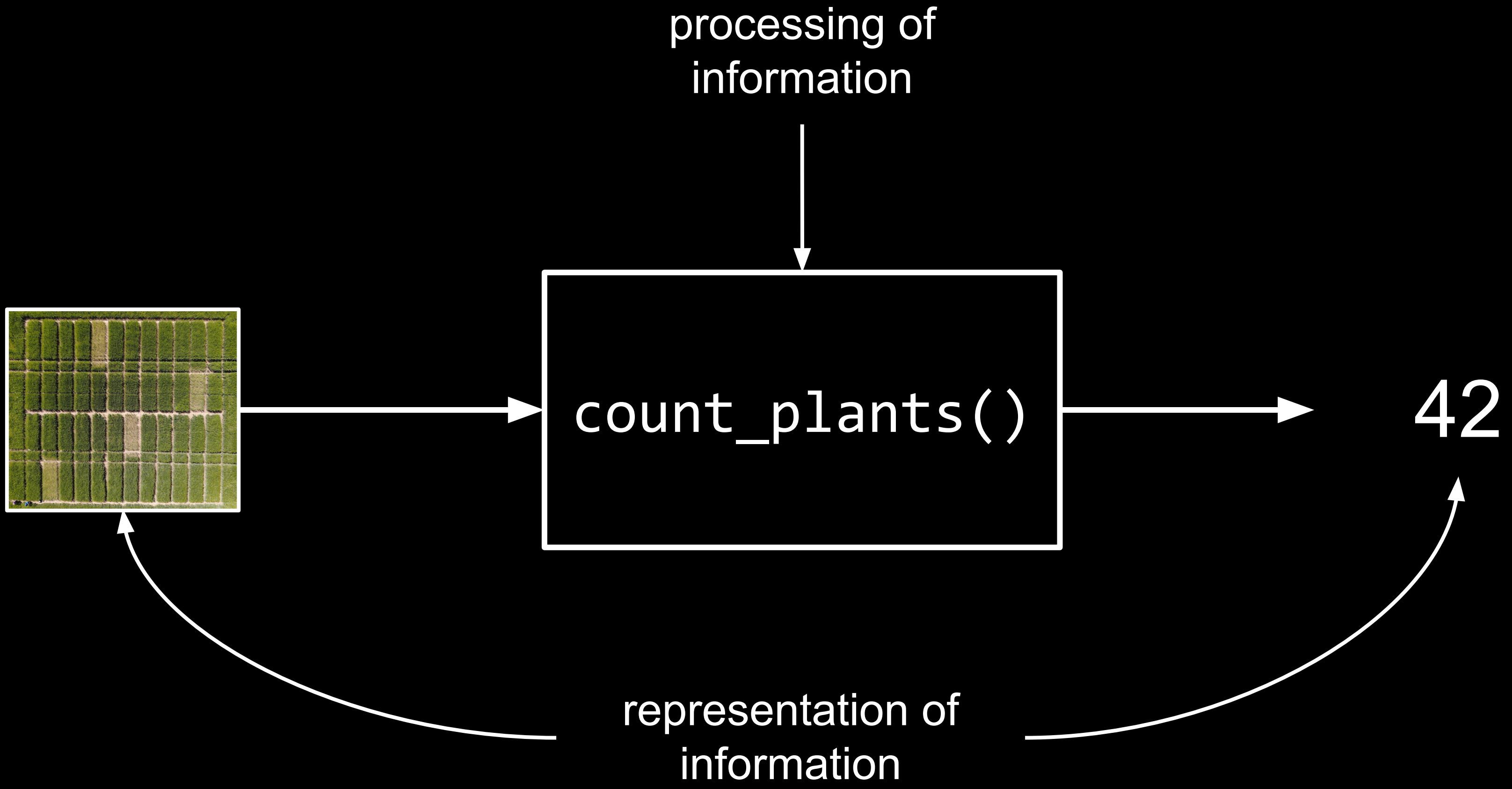


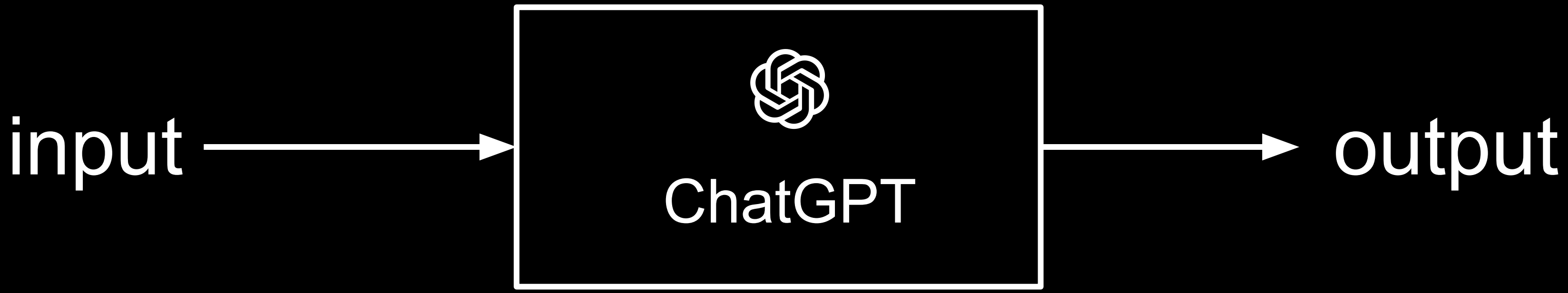
`count_plants()` → output

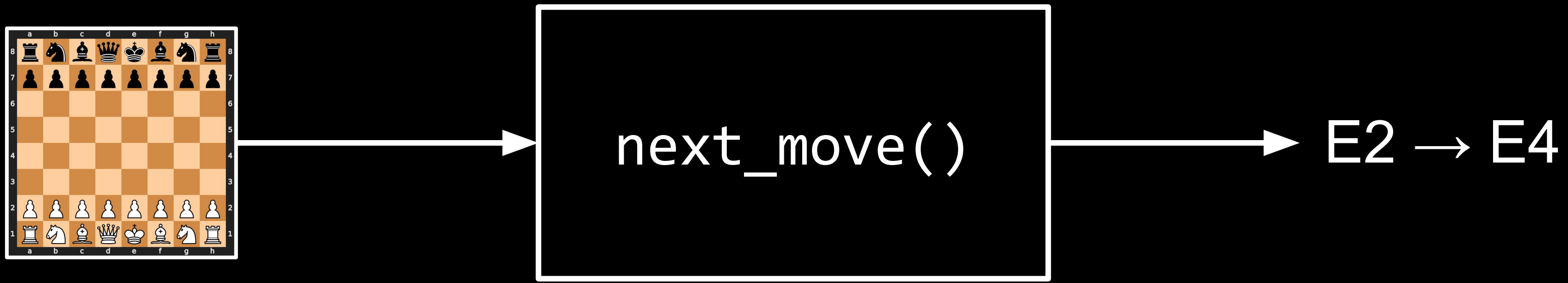


count_plants()

42


processing of information
count_plants()
42
representation of information


input
ChatGPT
output

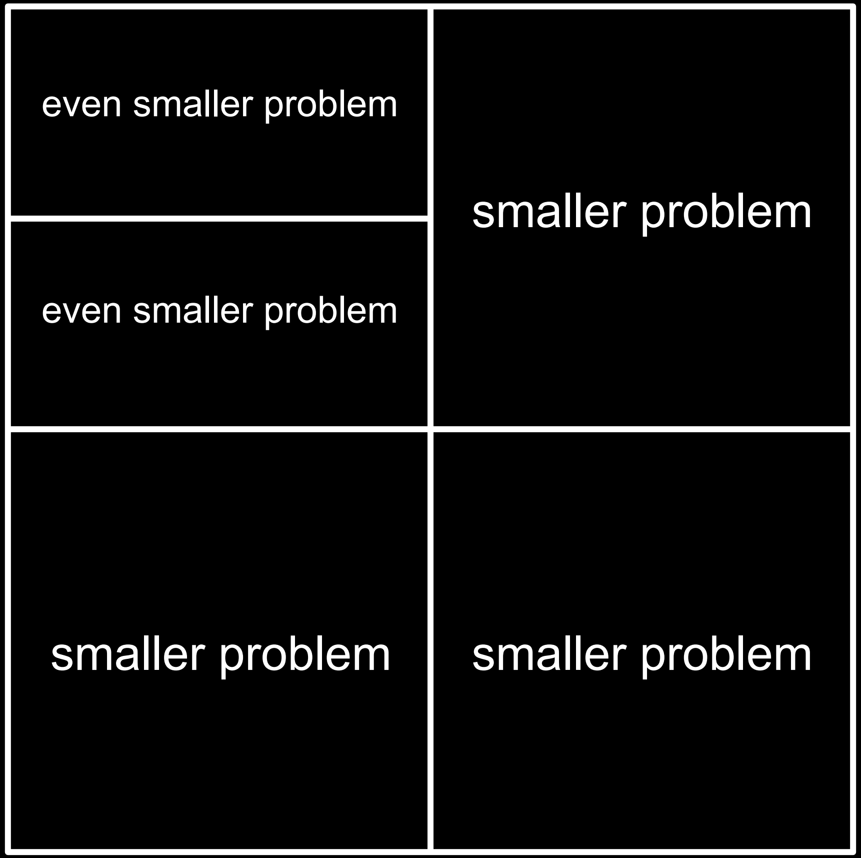

next_move()
E2 → E4

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

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

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

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

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’t 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

large and complex problem

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

large and complex problem

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

smaller problem

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41

smaller problem

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

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

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

⬆

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

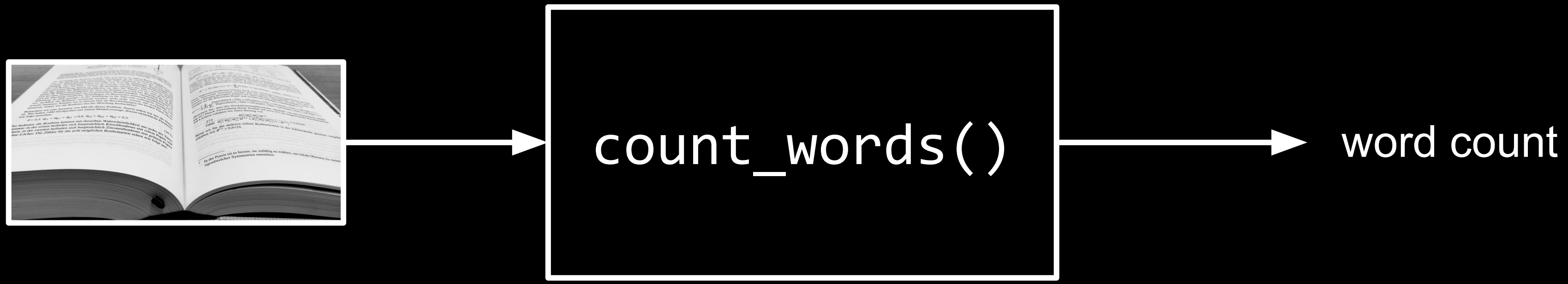
~~2~~, ~~3~~, ~~5~~, ~~7~~, ~~11~~, ~~13~~, ~~17~~, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
~~43~~, ~~47~~, ~~53~~, ~~59~~, ~~61,~~ 67, ~~71~~, ~~73~~, ~~79~~, ~~83~~, ~~89~~, ~~97~~

⬆

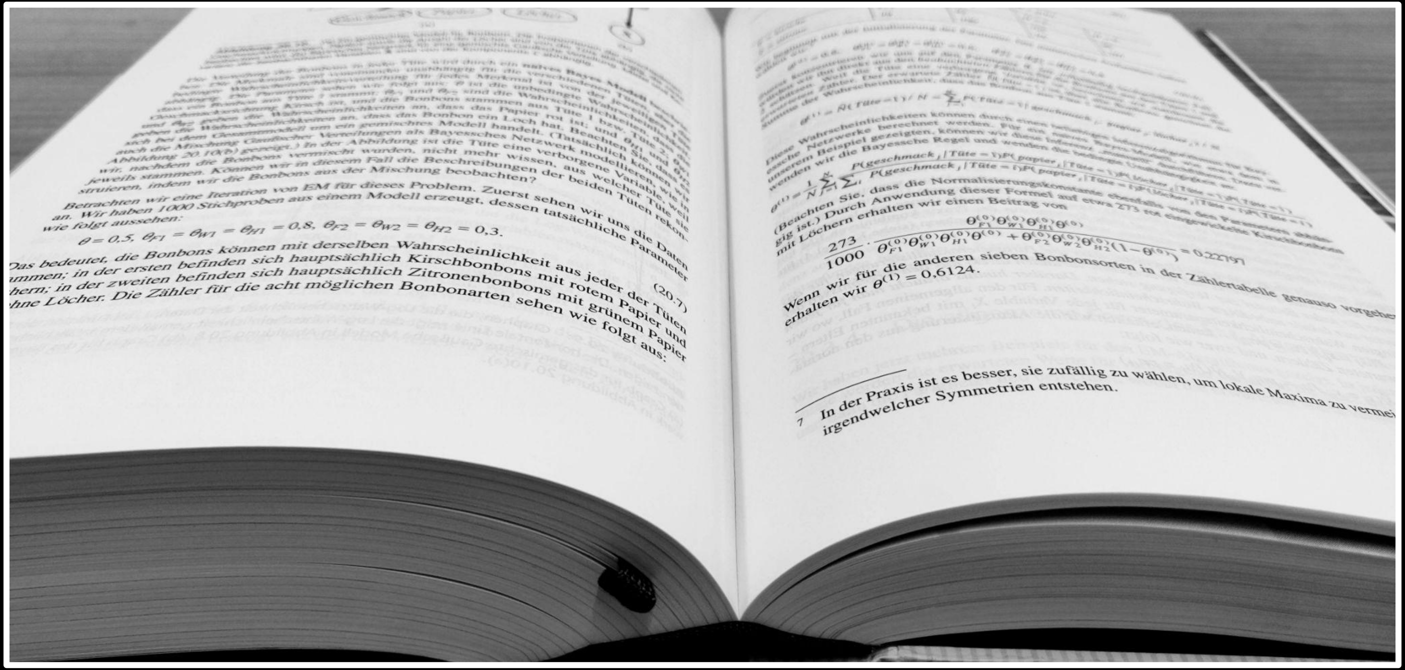
67 = 67

🤔

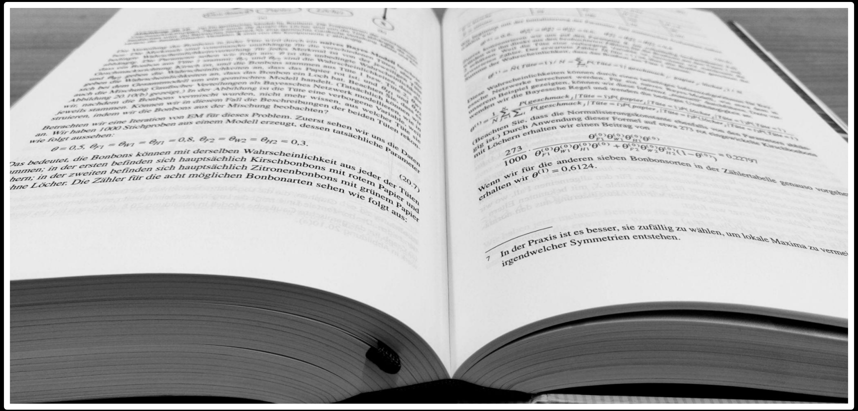
how efficient are linear and
binary search in general?


count_words()
word count

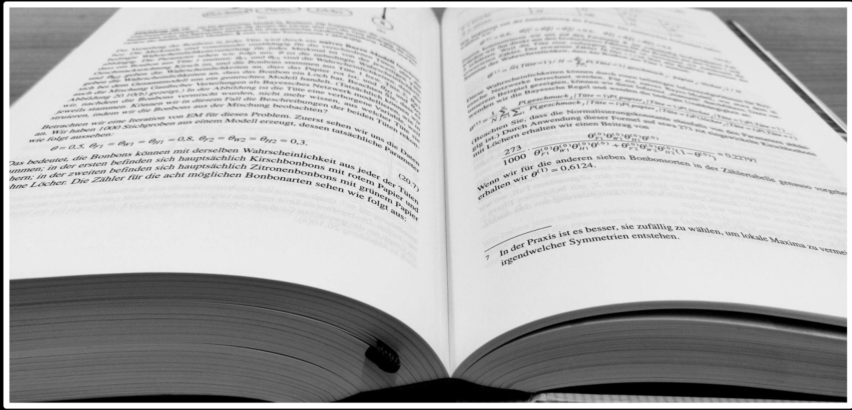
how many words are in the book?



strategies, anyone?

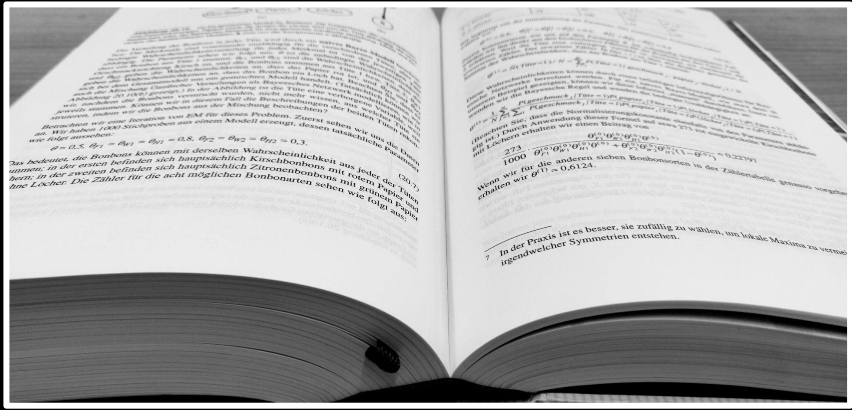


Betrachten wir eine Iteration von EM für dieses Problem. Zuerst sehen wir uns die Daten
an. Wir haben 1000 Stichproben aus einem Modell erzeugt, dessen tatsächliche Parameter
wie folgt aussehen:
erhalten wir $\theta^{(1)} = 0,6124$.
irgendwelcher Symmetrien entstehen.

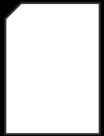


In der Praxis ist es besser, sie zufällig zu wählen, um lokale Maxima zu
irgendwelcher Symmetrien entstehen.

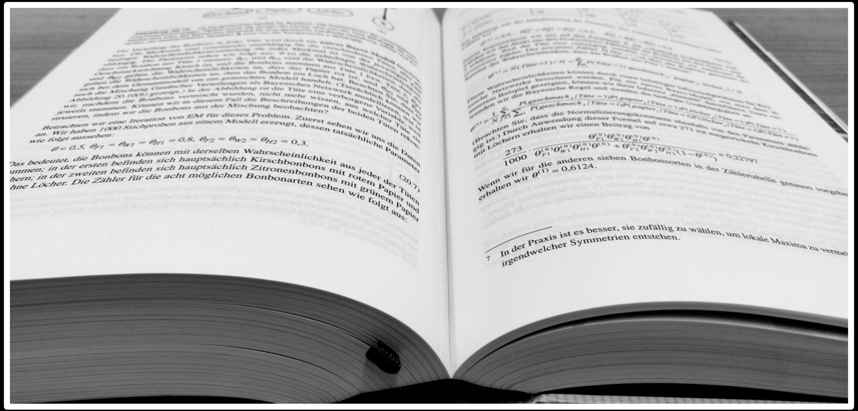
page 1

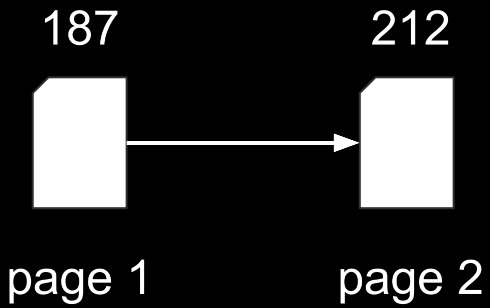


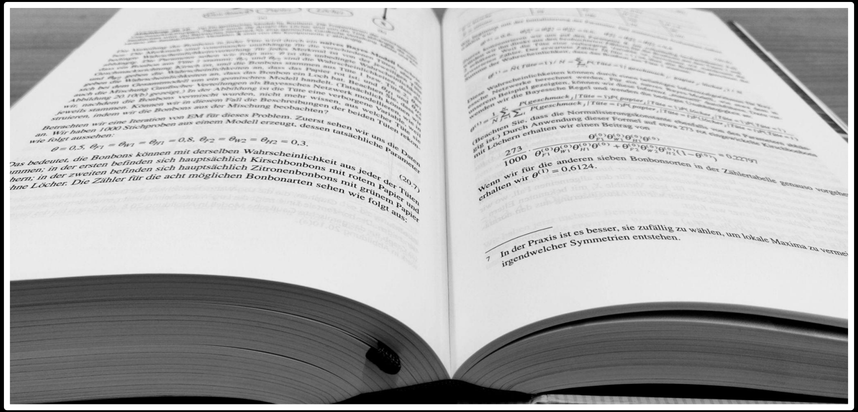
187

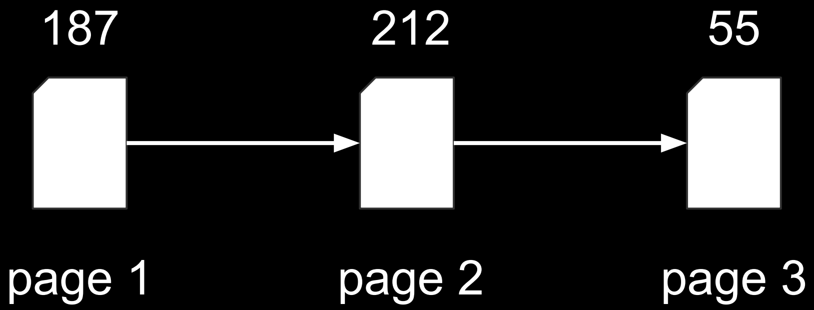


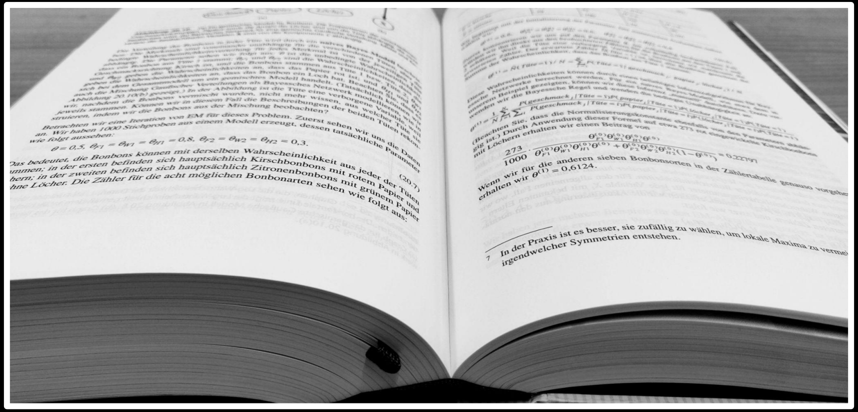
page 1

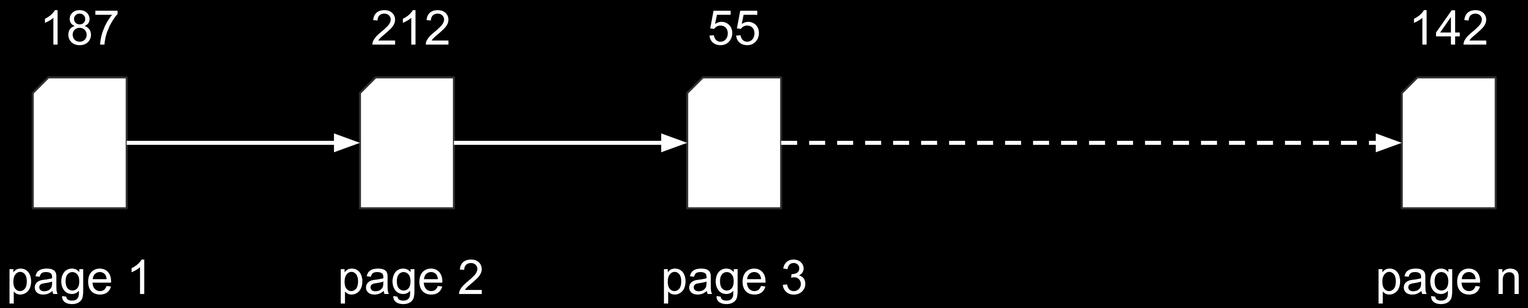


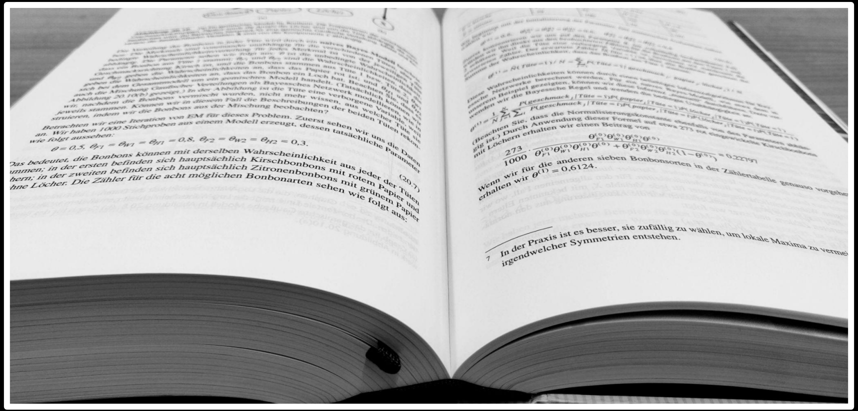

187
212
page 1
page 2




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

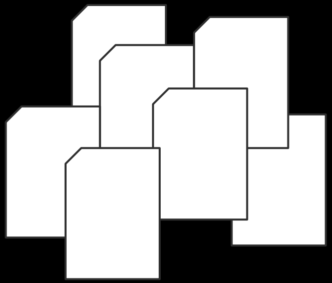
~ 52.34 hours

divide and conquer

+

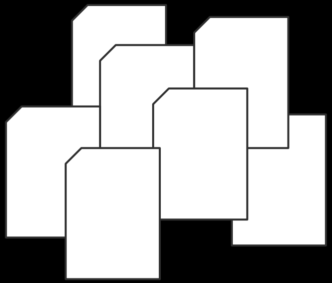
?

pages 1 - 700

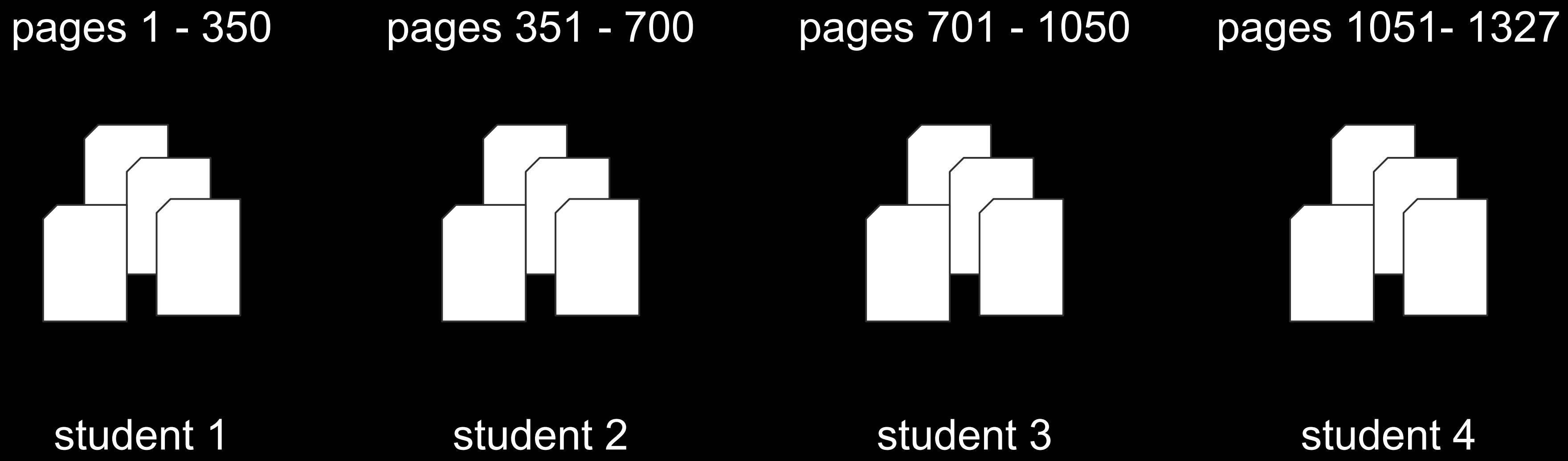


student 1

pages 701 - 1327



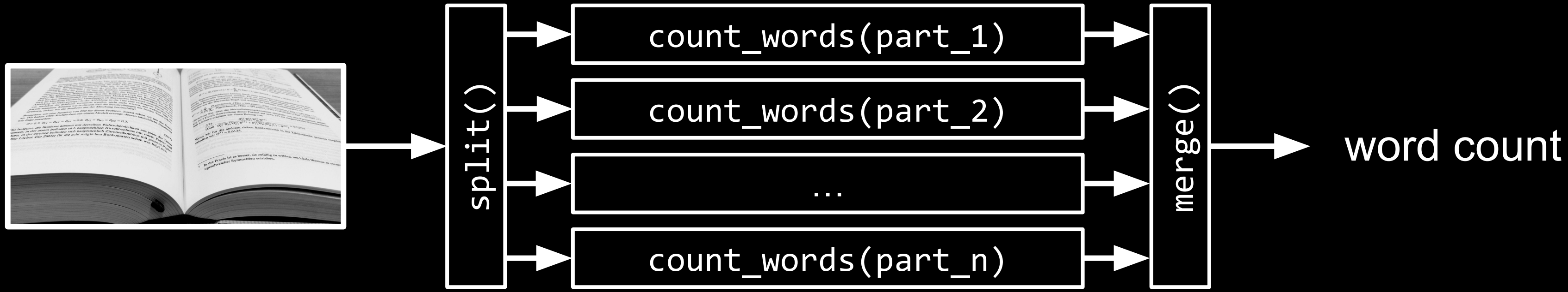
student 2


pages 1 - 350
pages 351 - 700
pages 701 - 1050
pages 1051- 1327
student 1
student 2
student 3
student 4

divide and conquer

+

distribution and parallelization


split()
count_words(part_1)
count_words(part_2)
...
count_words(part_n)
merge()
word count

INFORMATION

“Information is that which allows you to make a correct prediction with accuracy better than chance.”

Adami, Christoph. “What Is Information?” Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 374, no. 2063, Mar. 2016, p. 20150230, https://doi.org/10.1098/rsta.2015.0230.

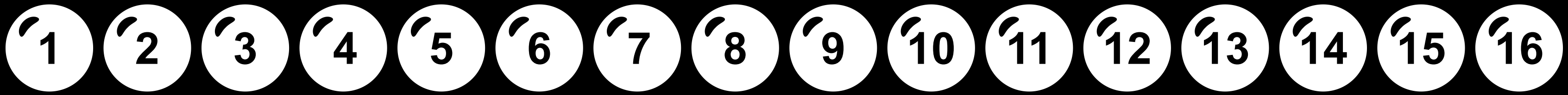
“Information is that which allows you to make a correct prediction with accuracy better than chance.”

Adami, Christoph. “What Is Information?” Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 374, no. 2063, Mar. 2016, p. 20150230, https://doi.org/10.1098/rsta.2015.0230.

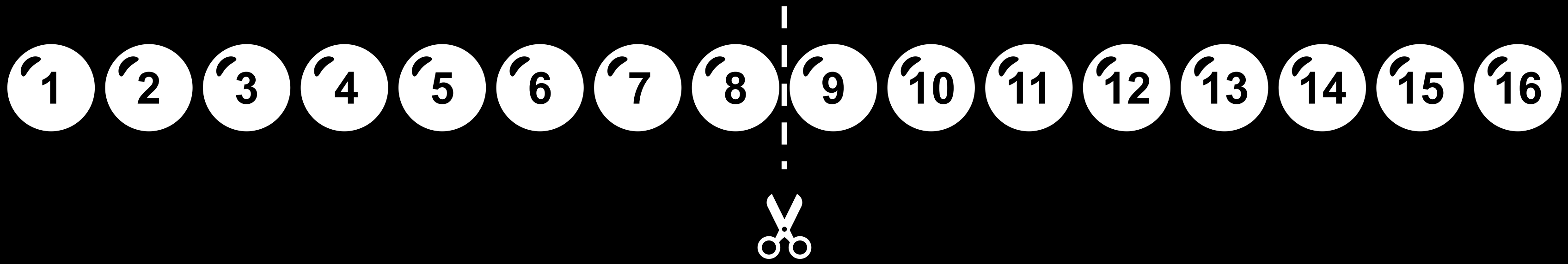
defining and measuring information

guess the number am I thinking of!

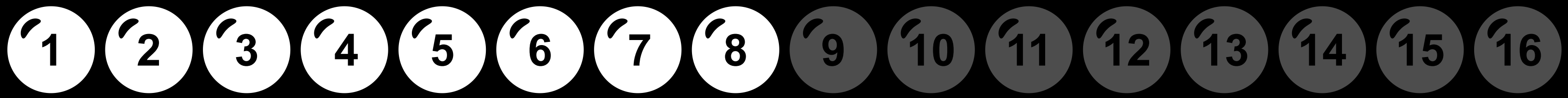
what is the most efficient approach?



is it > 8?

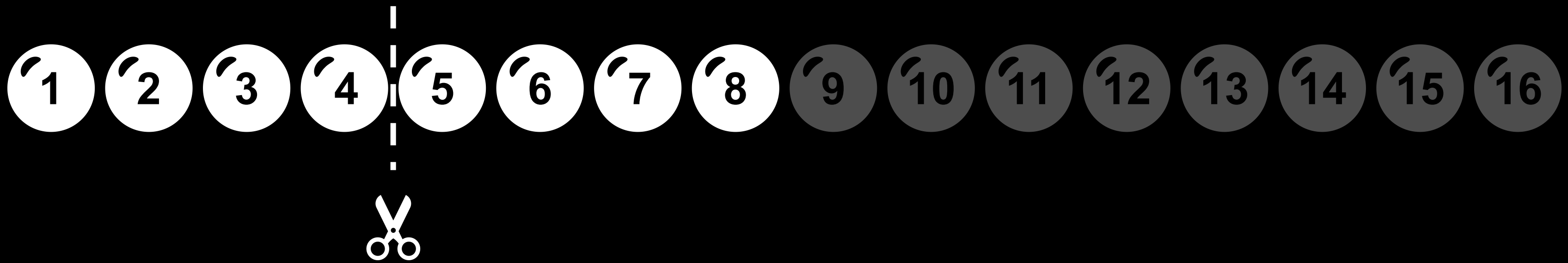


is it > 8? ✗



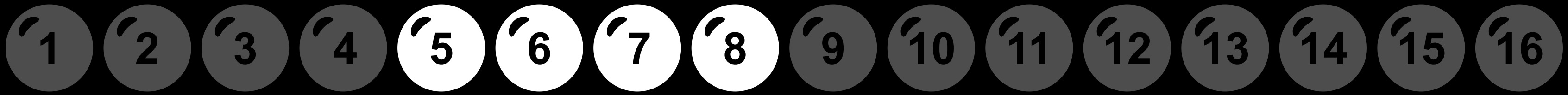
is it > 8? ✗

is it > 4?



is it > 8? ✗

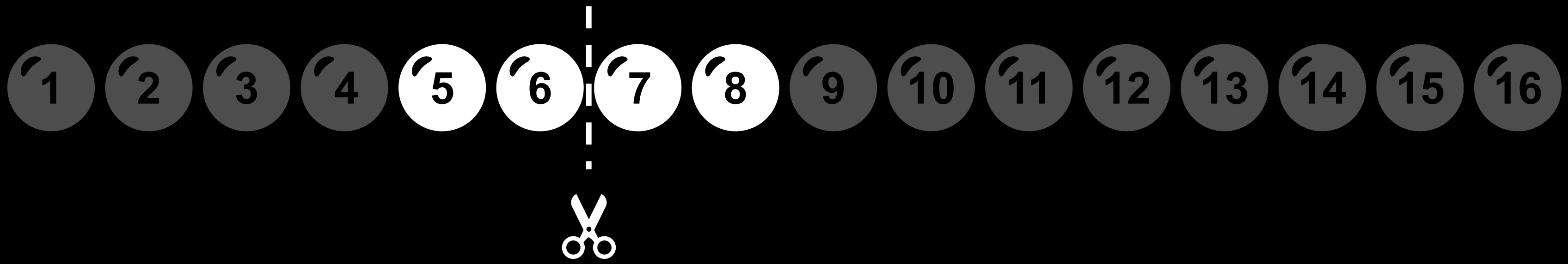
is it > 4? ✓



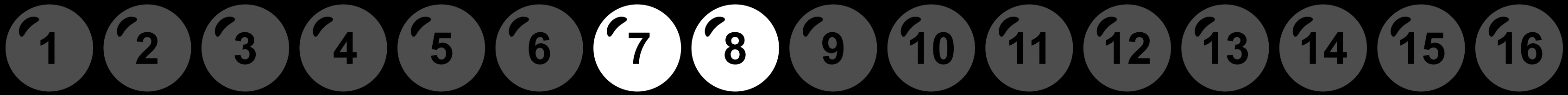
is it > 8? ✗

is it > 4? ✓

is it > 6?



is it > 8? ✗
is it > 4? ✓
is it > 6? ✓

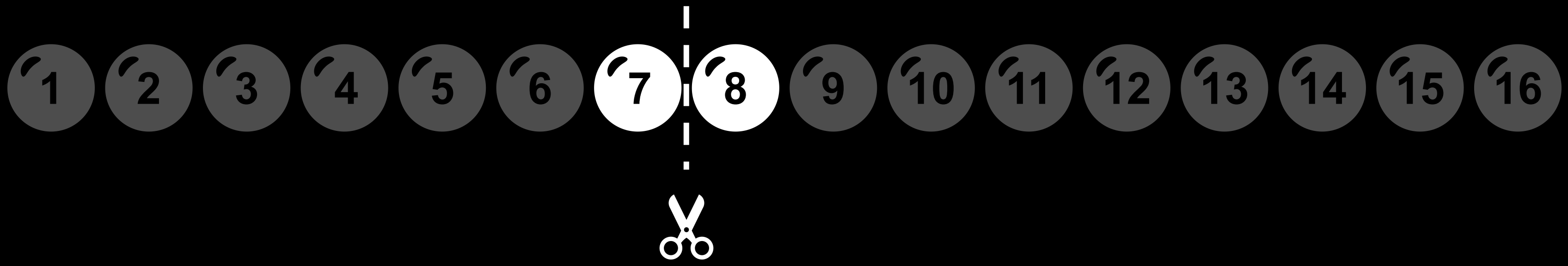


is it > 8? ✗

is it > 4? ✓

is it > 6? ✓

is it > 7?

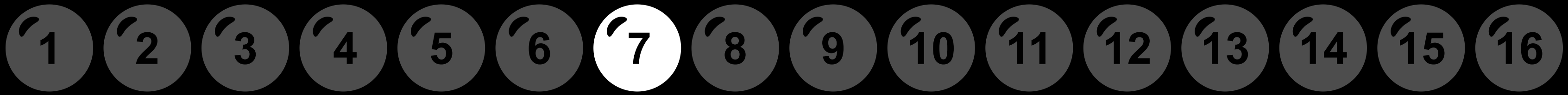


is it > 8? ✗

is it > 4? ✓

is it > 6? ✓

is it > 7? ✗

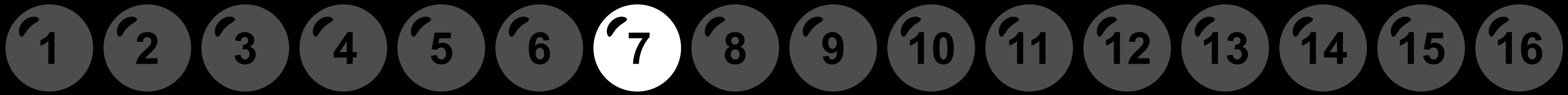


is it > 8? ✗

is it > 4? ✓

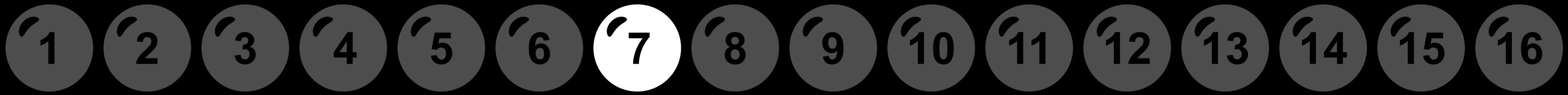
is it > 6? ✓

is it > 7? ✗



with 4 questions from 16 to 1

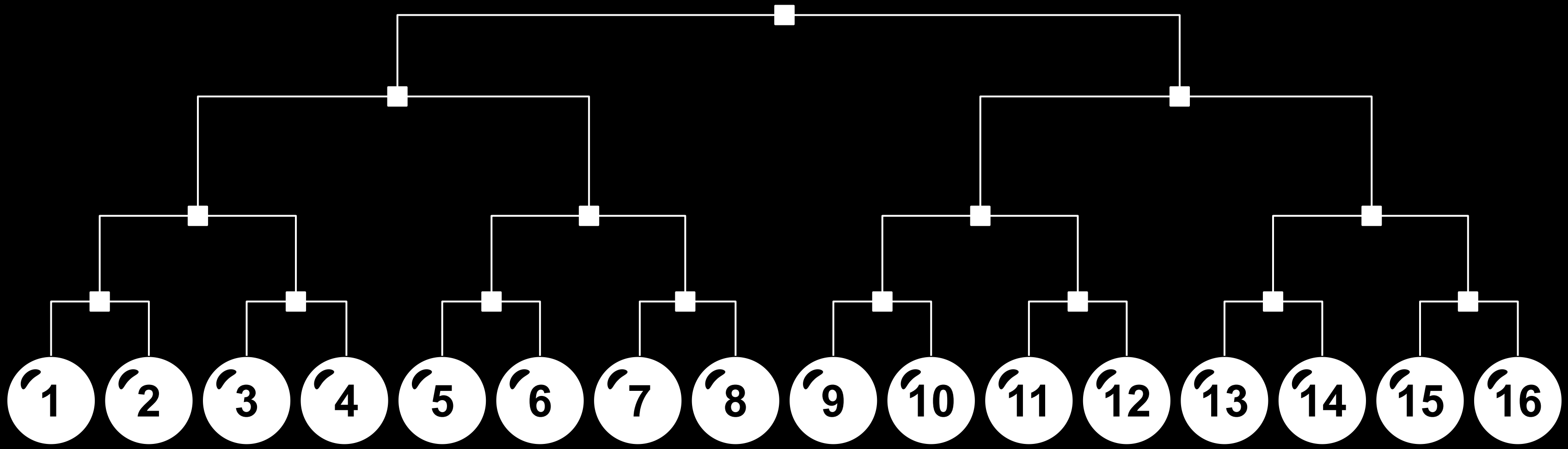
is it > 8? ✗ 0
is it > 4? ✓ 1
is it > 6? ✓ 1
is it > 7? ✗ 0



with 4 questions from 16 to 1

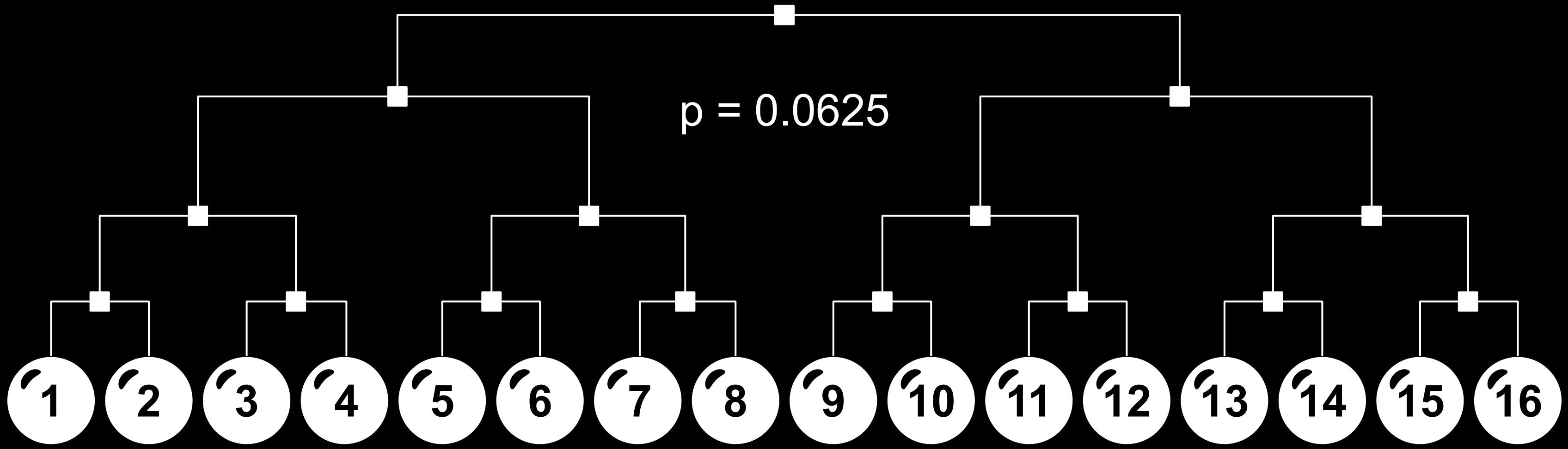
where is the information?

where is the information?



where is the information?

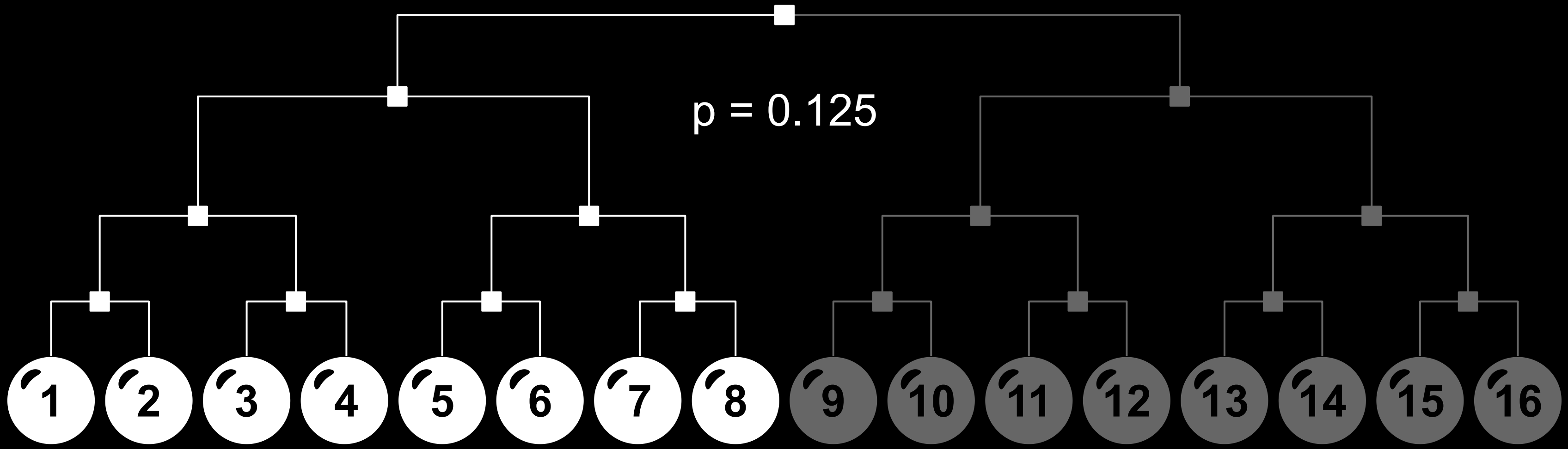
N = 16



where is the information?

~~N = 16~~

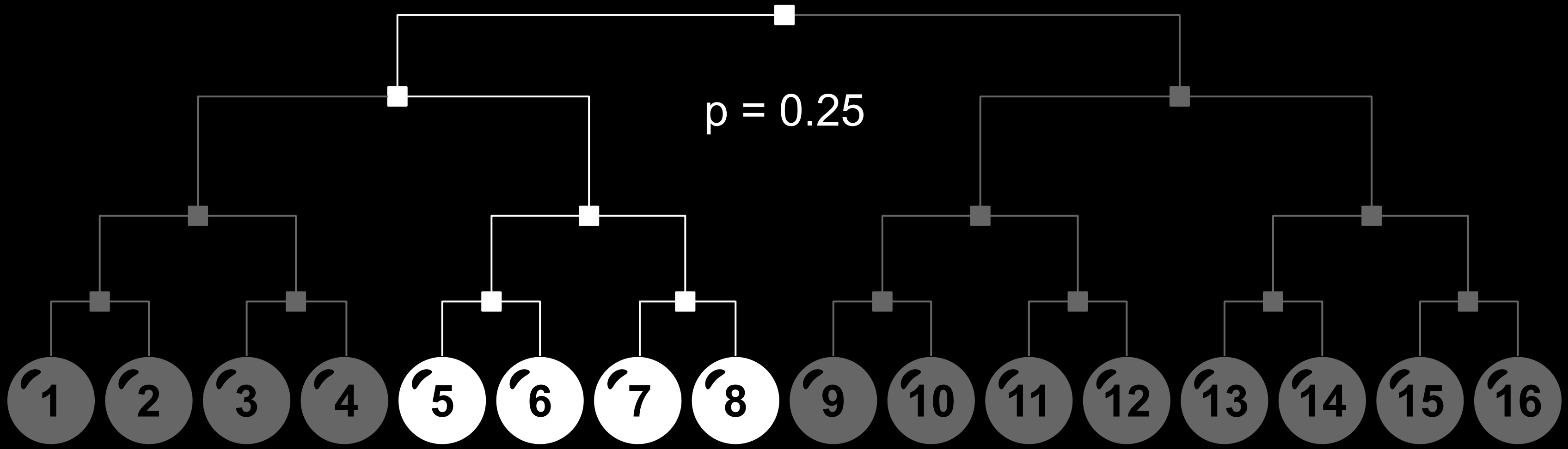
N = 8



where is the information?

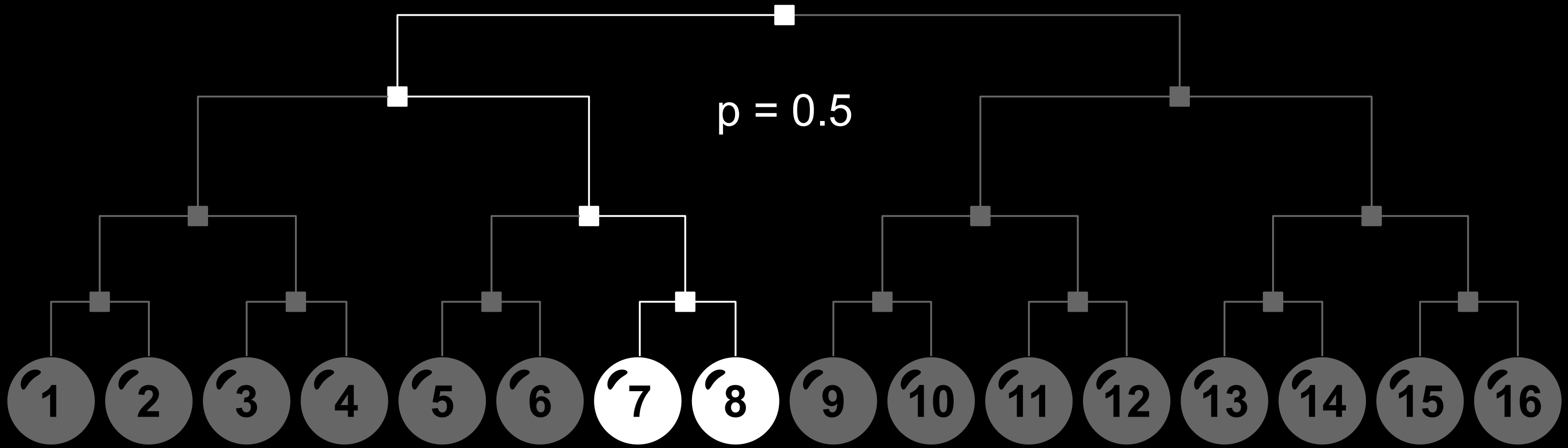
~~N = 16~~
~~N = 8~~
N = 4

p = 0.25



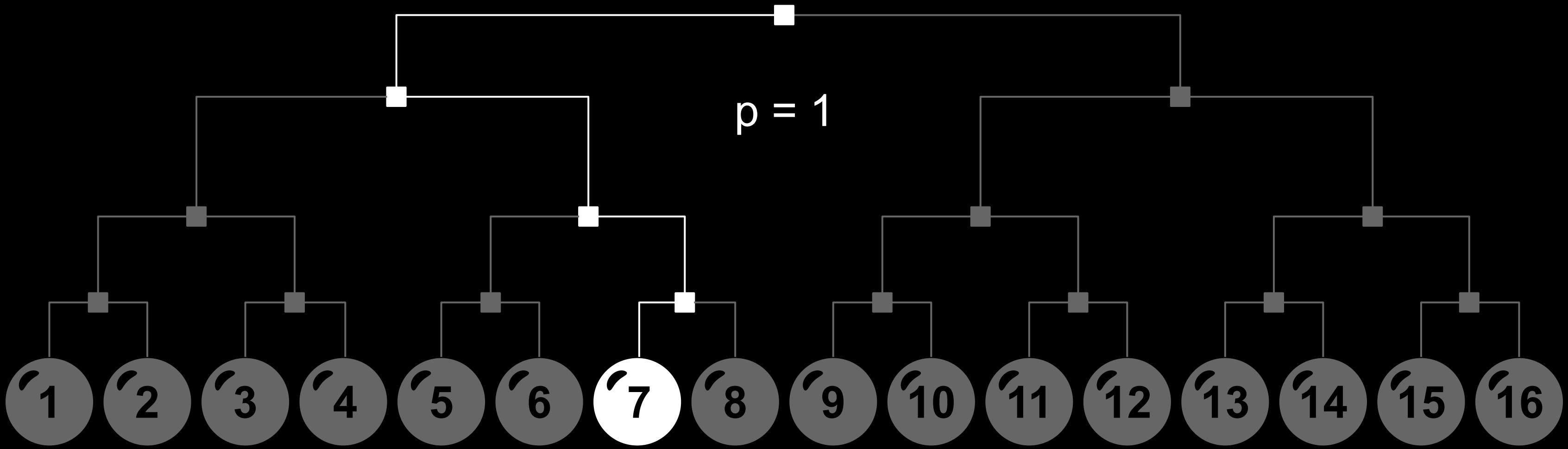
where is the information?

~~N = 16~~
~~N = 8~~
~~N = 4~~
N = 2



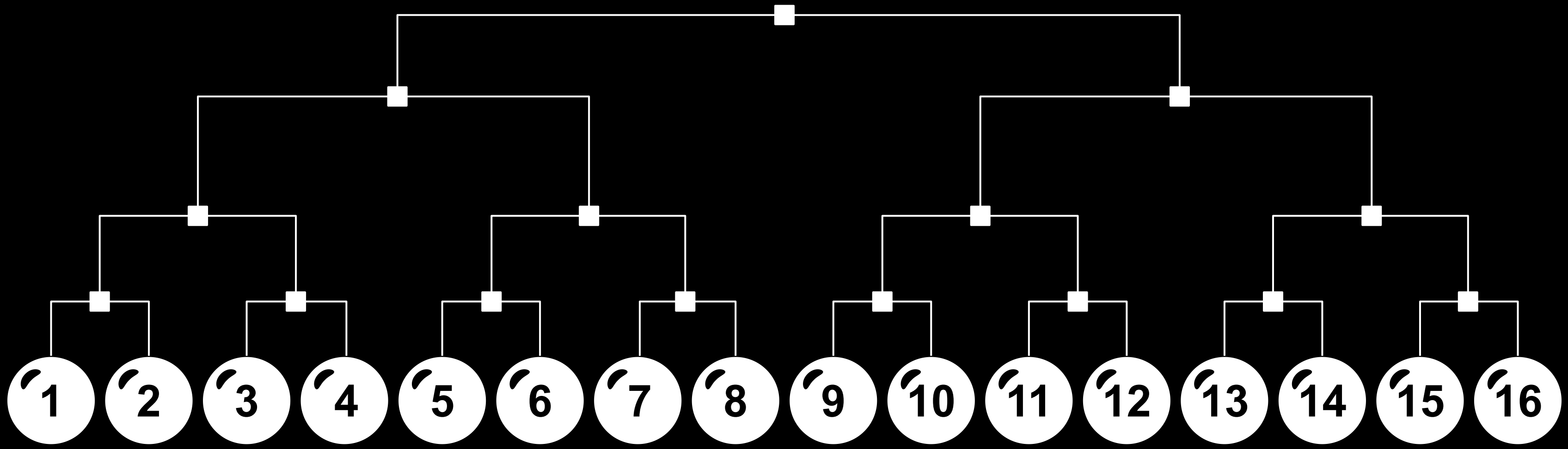
where is the information?

~~N = 16~~
~~N = 8~~
~~N = 4~~
~~N = 2~~
N = 1



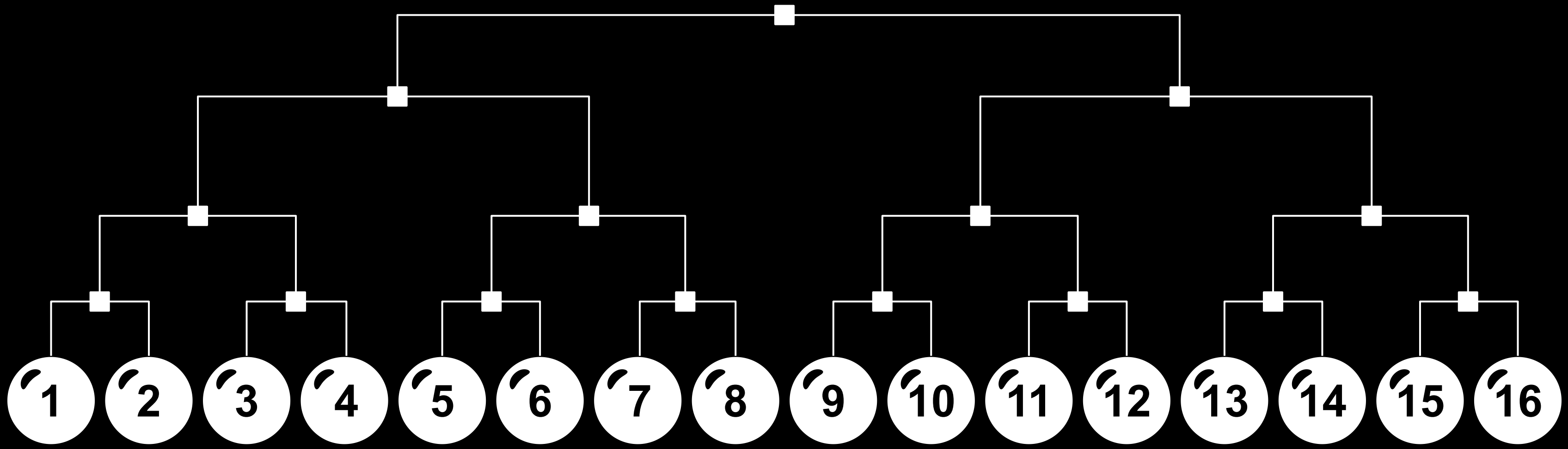
information = reduced uncertainty
uncertainty is measured with the logarithm of N

$$H = log_2(N)$$

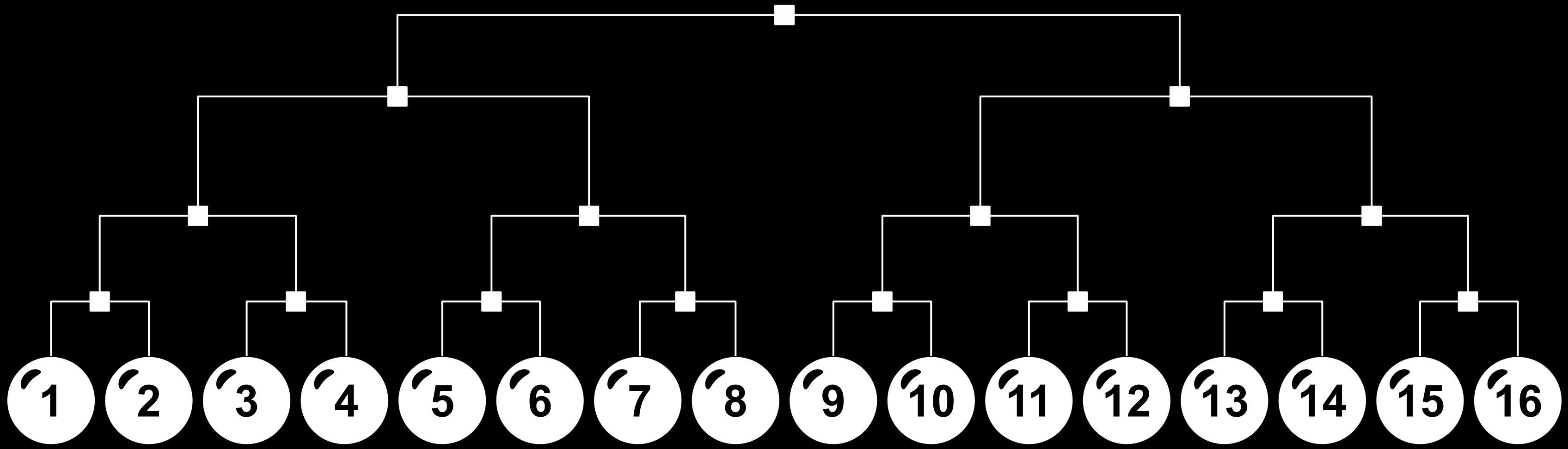


or: how often can we cut the remaining possibilities in half?

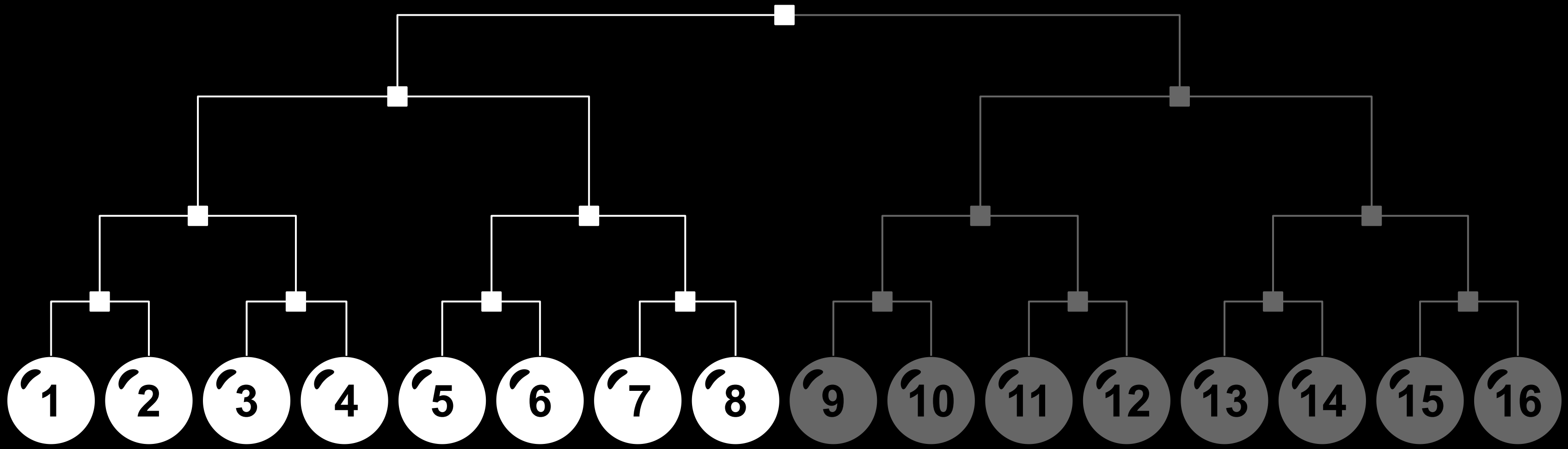
$$H = log_2(N)$$



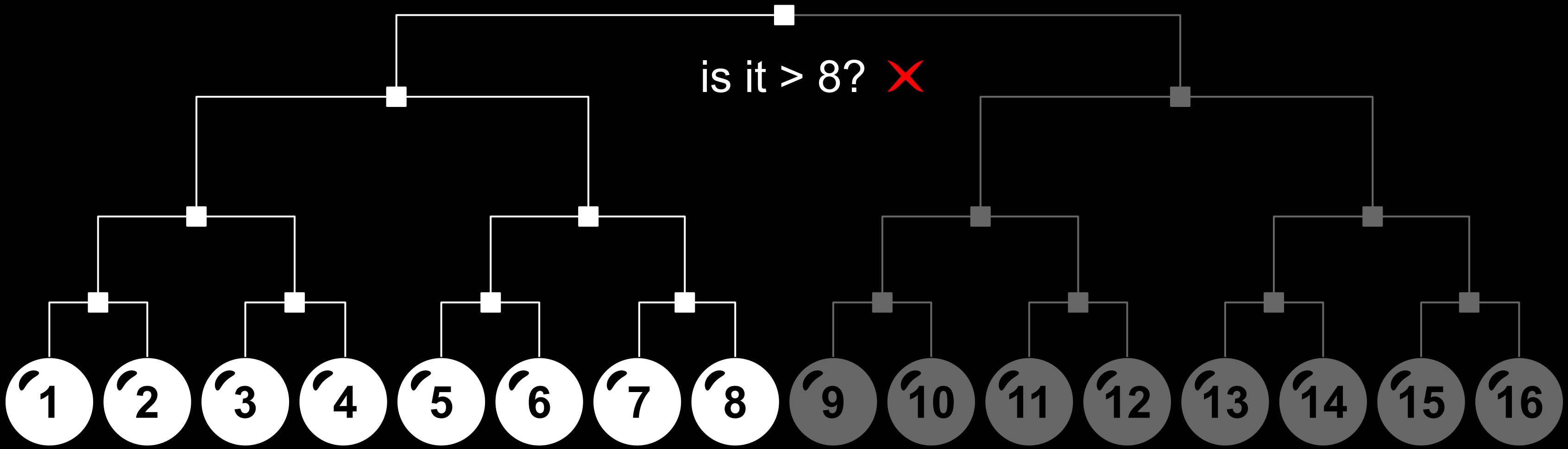
$$H_0 = log_2(16) = 4$$



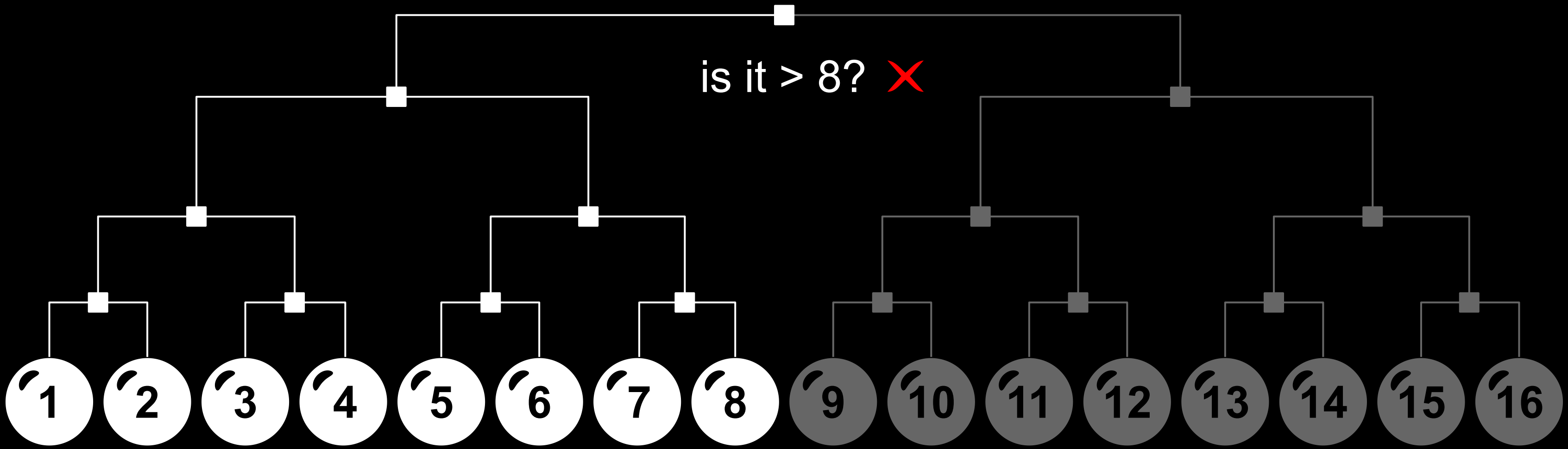
$$H_1 = log_2(8) = 3$$



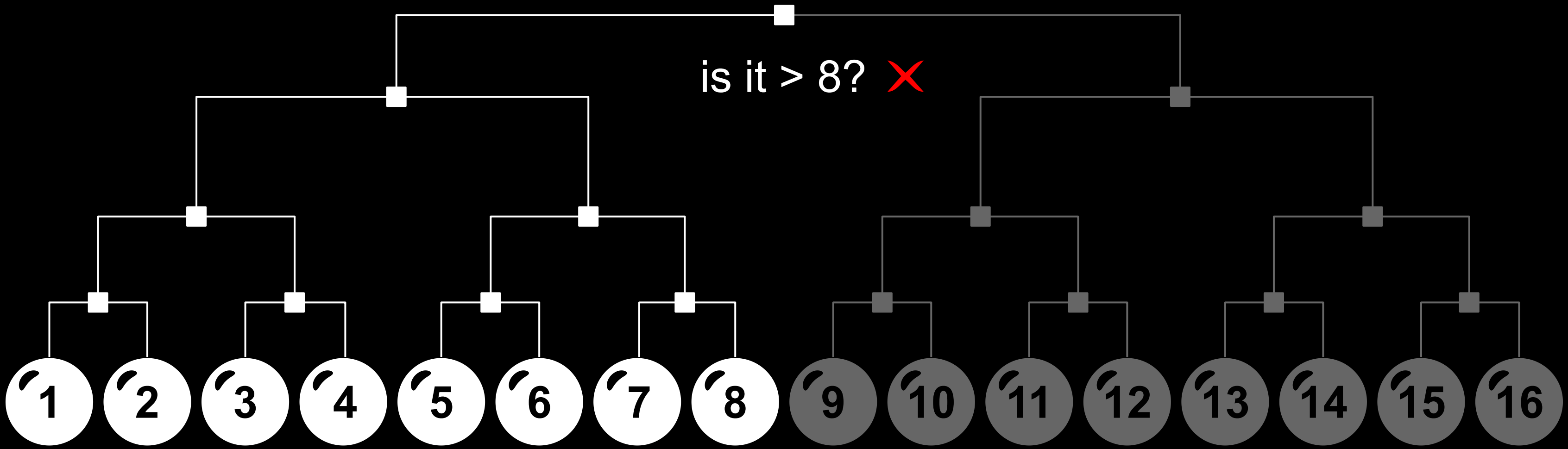
$$I = H_0 - H_1$$



$$I = log_2(16) - log_2(8)$$



$$I = 4 - 3 = 1$$



uncertainty and information are measured in **bits** how many yes/no questions to reduce uncertainty to zero?

$$H = 0 = log_2(1)$$

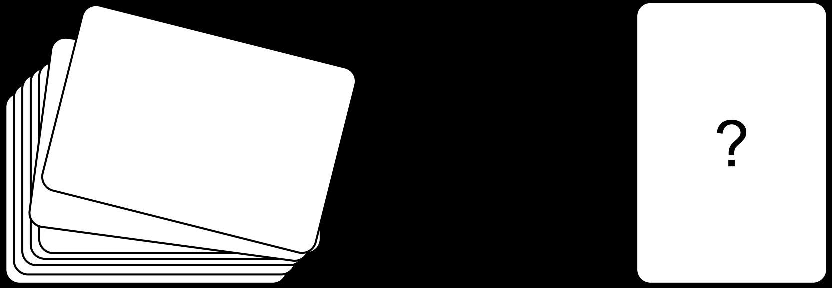
how many yes/no questions to reduce uncertainty to zero?

$$H = 0 = log_2(1)$$

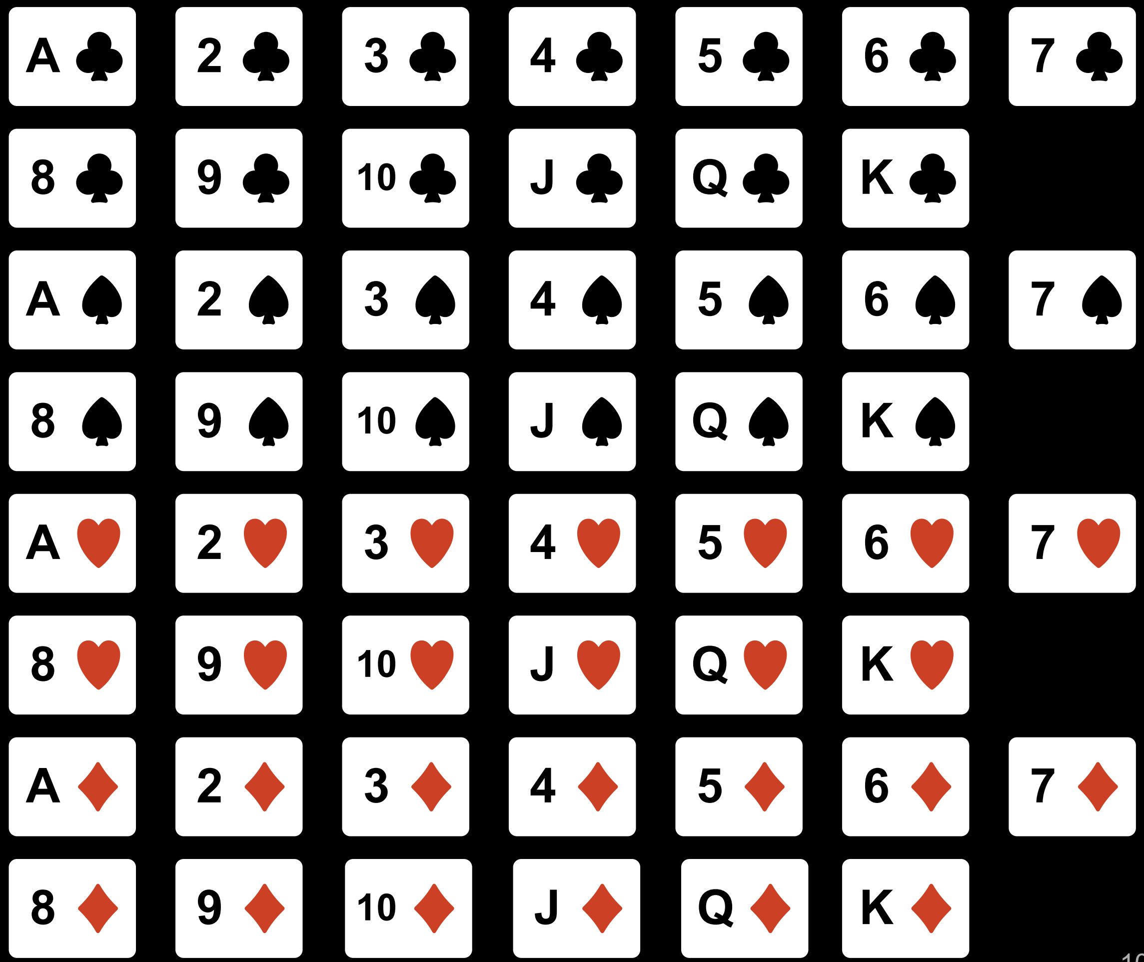
$$H = log_2(N)$$

poker

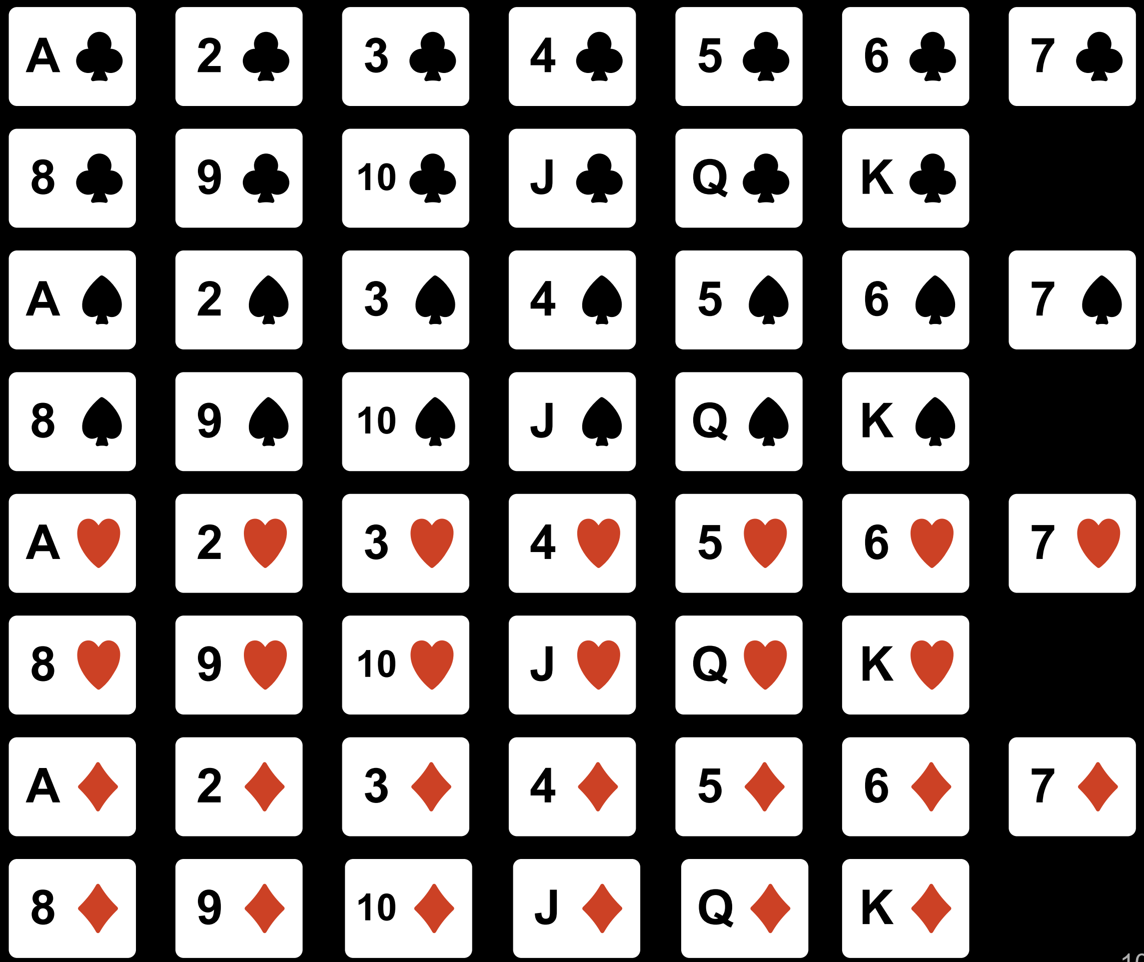
which card am I holding?



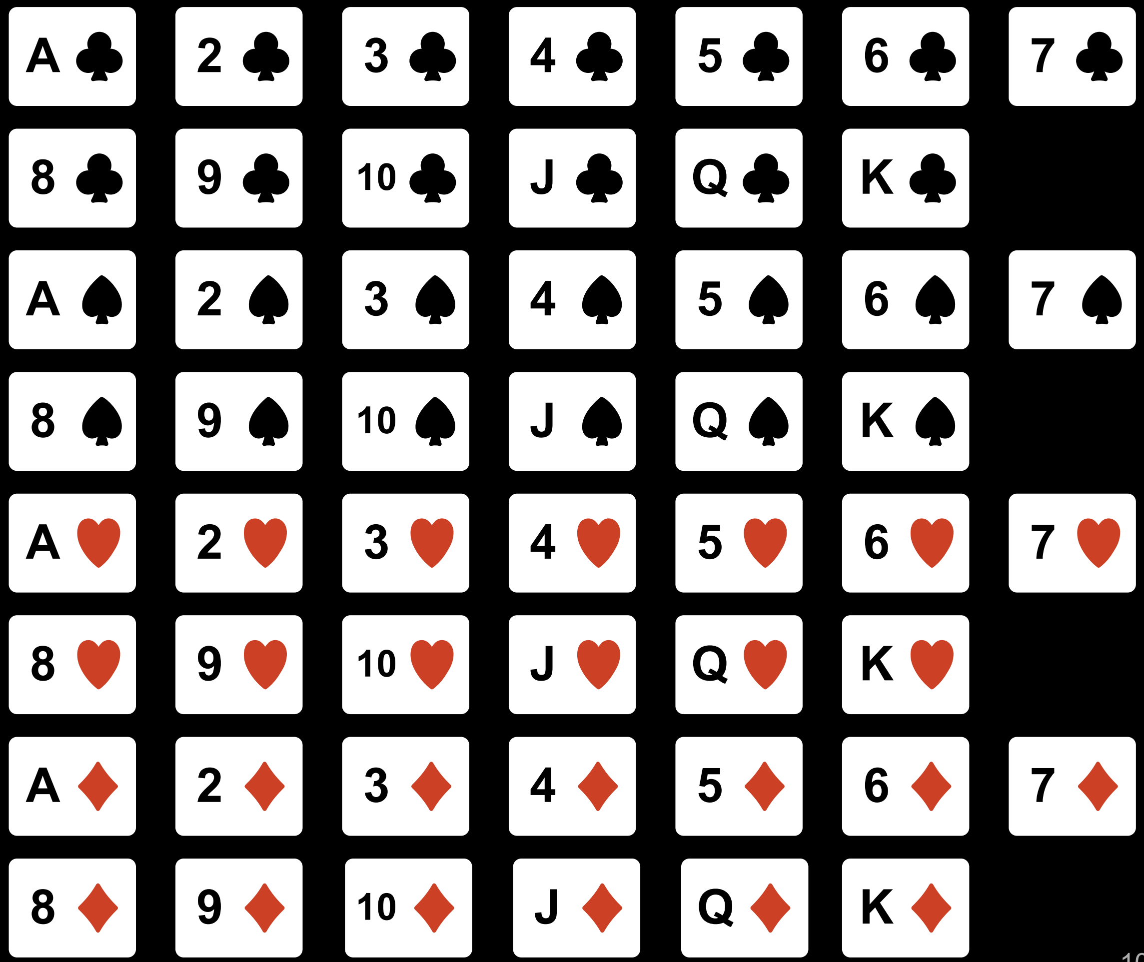
52 card poker deck


A ♣ 2 ♣ 3 ♣ 4 ♣ 5 ♣ 6 ♣ 7 ♣
8 ♣ 9 ♣ 10 ♣ J ♣ Q ♣ K ♣
A ♠ 2 ♠ 3 ♠ 4 ♠ 5 ♠ 6 ♠ 7 ♠
8 ♠ 9 ♠ 10 ♠ J ♠ Q ♠ K ♠
A ♥ 2 ♥ 3 ♥ 4 ♥ 5 ♥ 6 ♥ 7 ♥
8 ♥ 9 ♥ 10 ♥ J ♥ Q ♥ K ♥
A ♦ 2 ♦ 3 ♦ 4 ♦ 5 ♦ 6 ♦ 7 ♦
8 ♦ 9 ♦ 10 ♦ J ♦ Q ♦ K ♦

52 possible cards

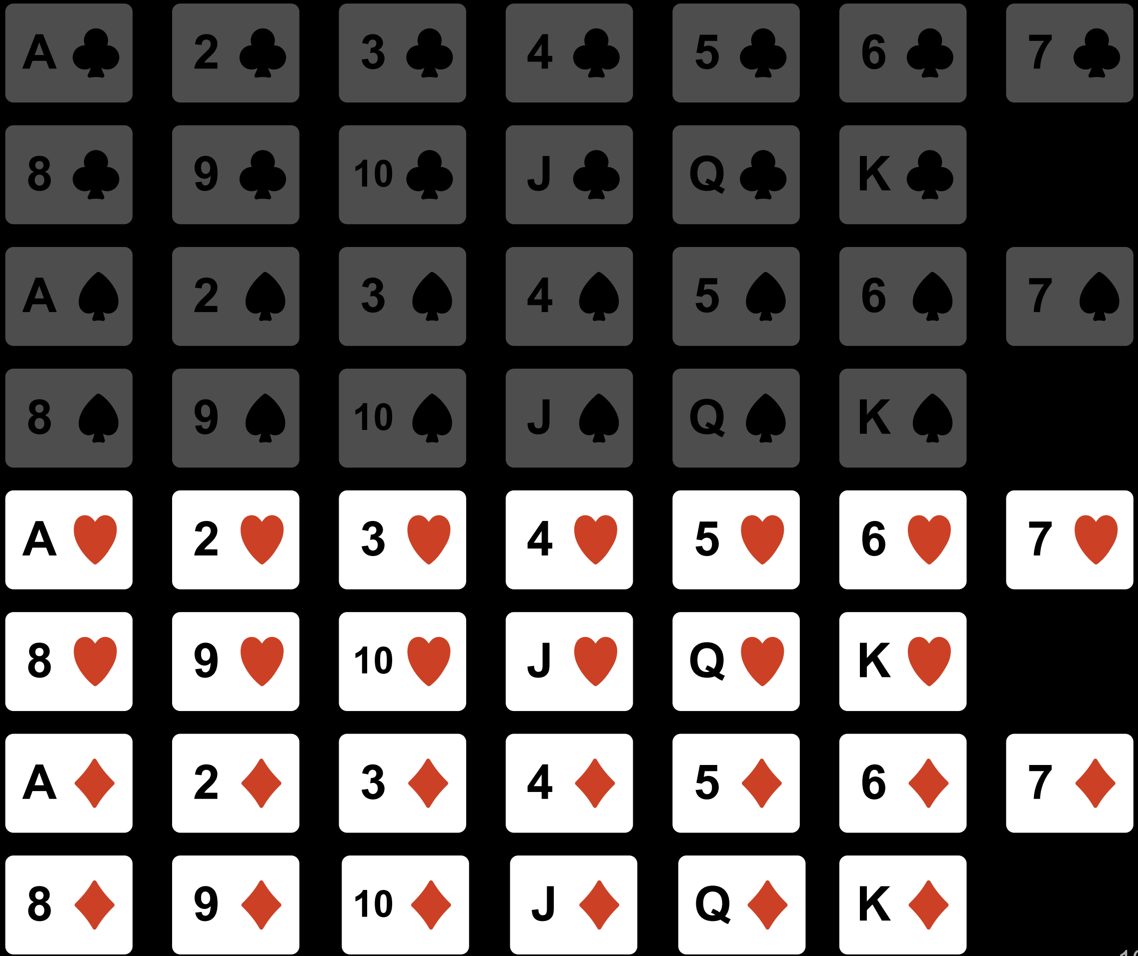


is the card black?



is the card black?

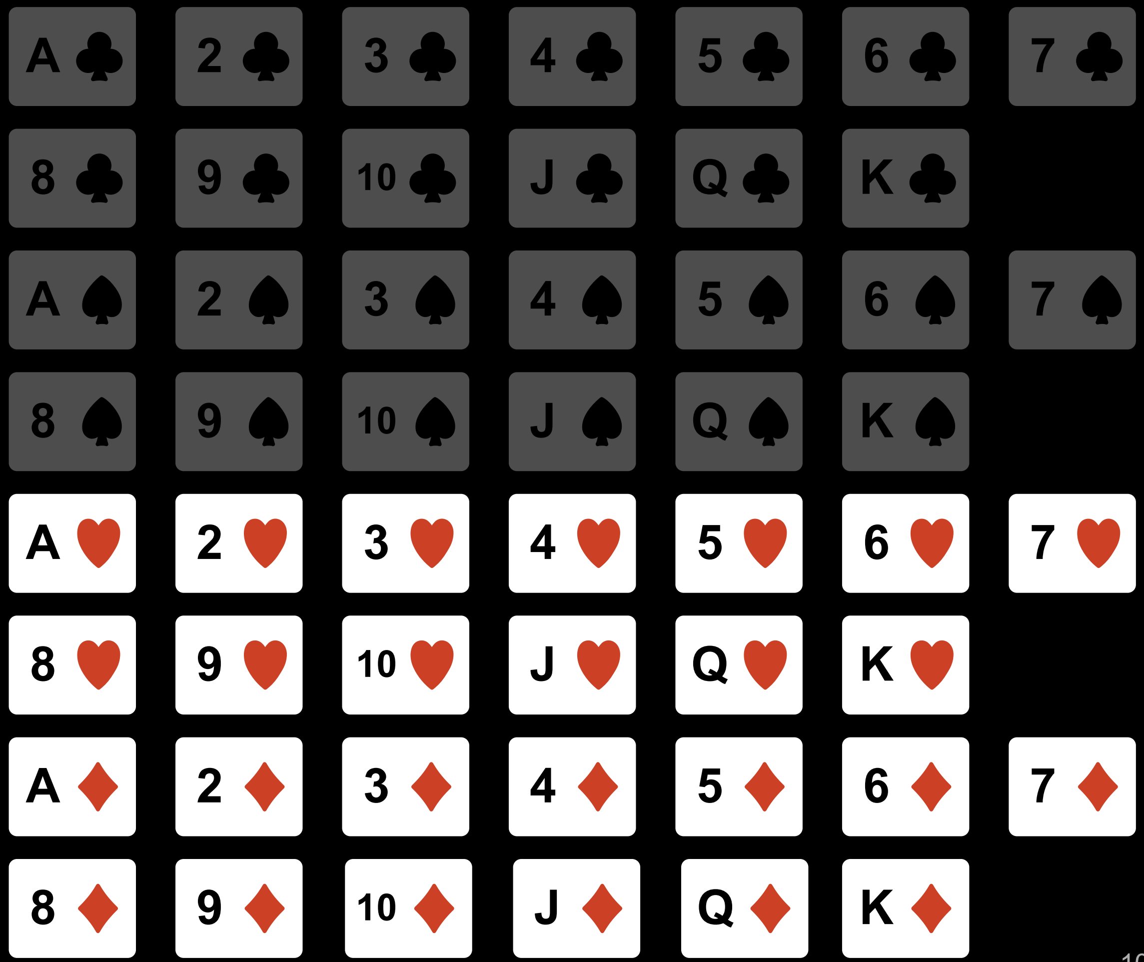
no



is the card black?

no

is it hearts?

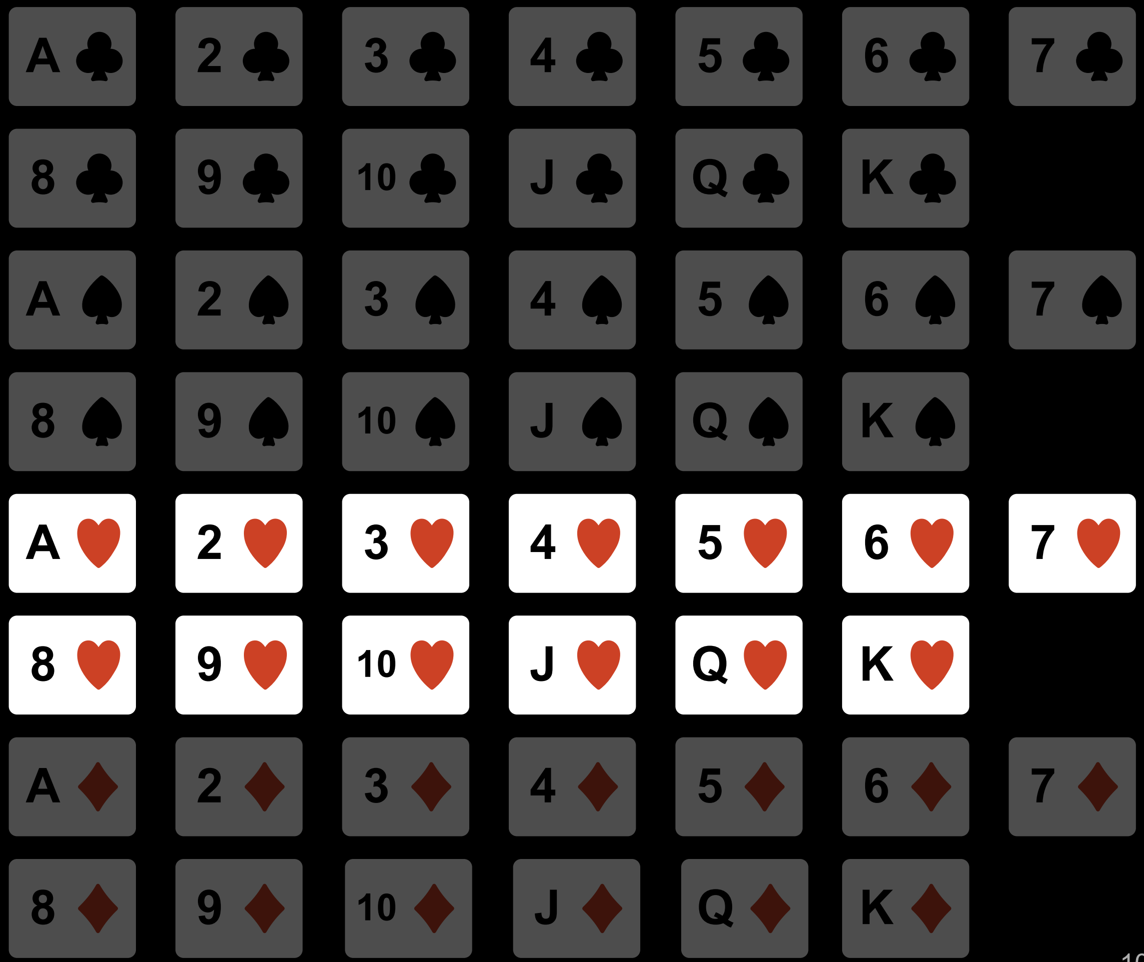


is the card black?

no

is it hearts?

yes



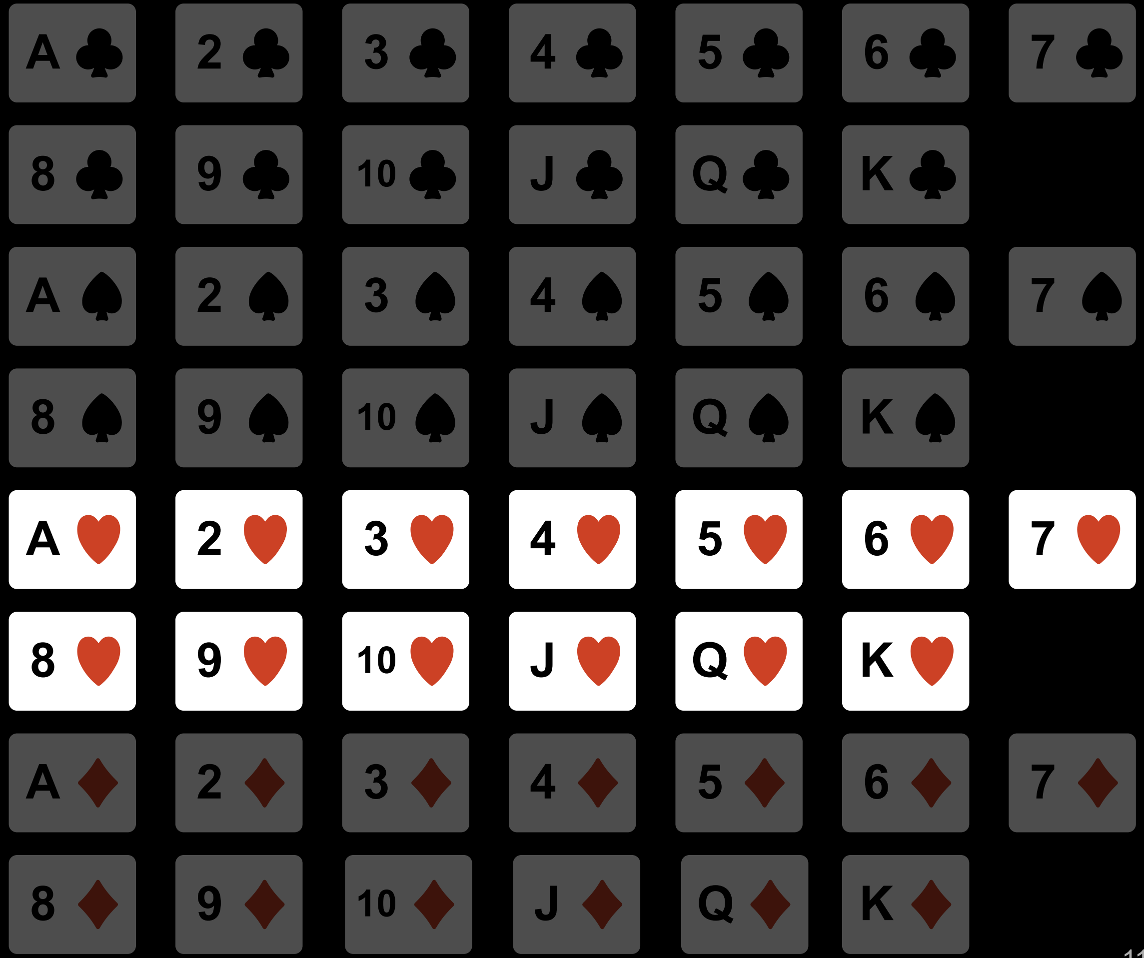
is the card black?

no

is it hearts?

yes

is it 8 or above?



is the card black?

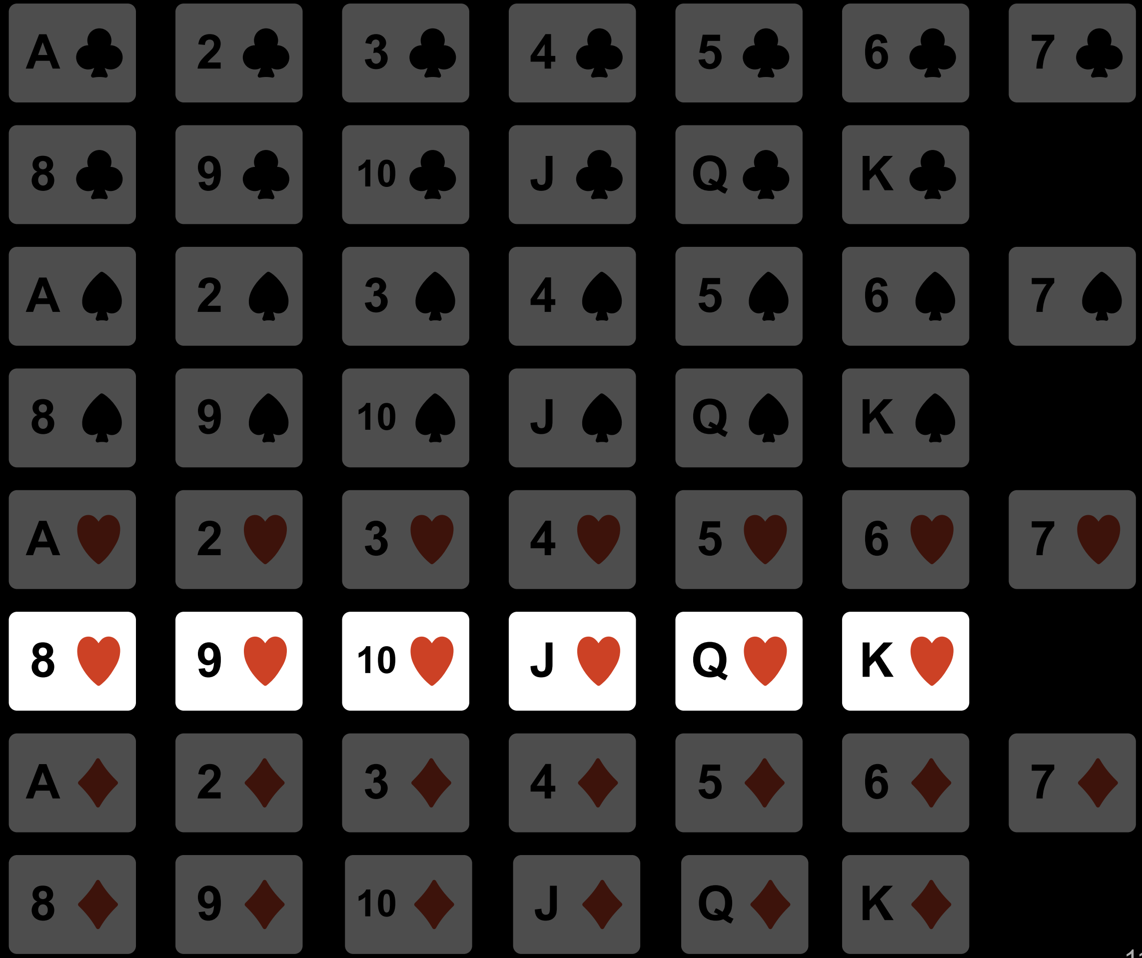
no

is it hearts?

yes

is it 8 or above?

yes



is the card black?

no

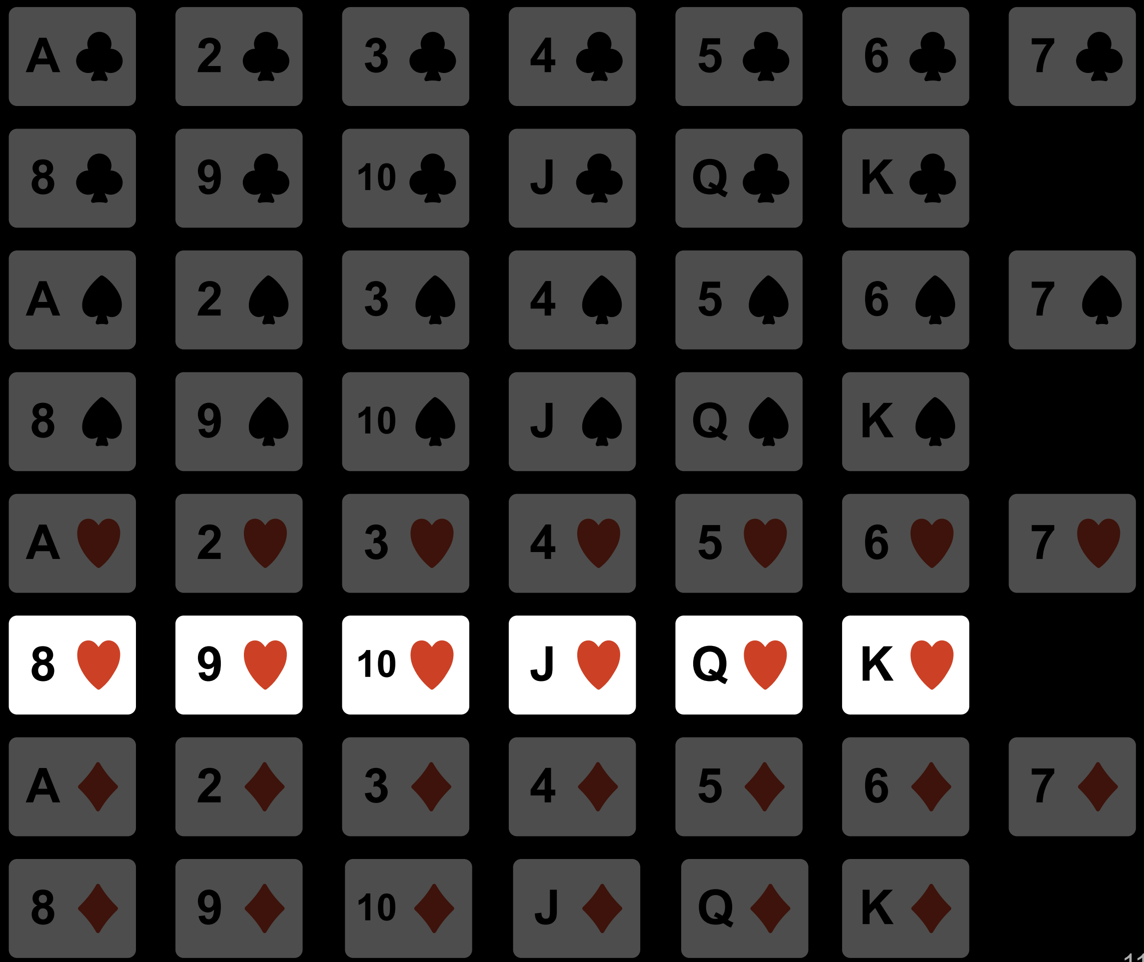
is it hearts?

yes

is it 8 or above?

yes

is it jack or above?



is the card black?

no

is it hearts?

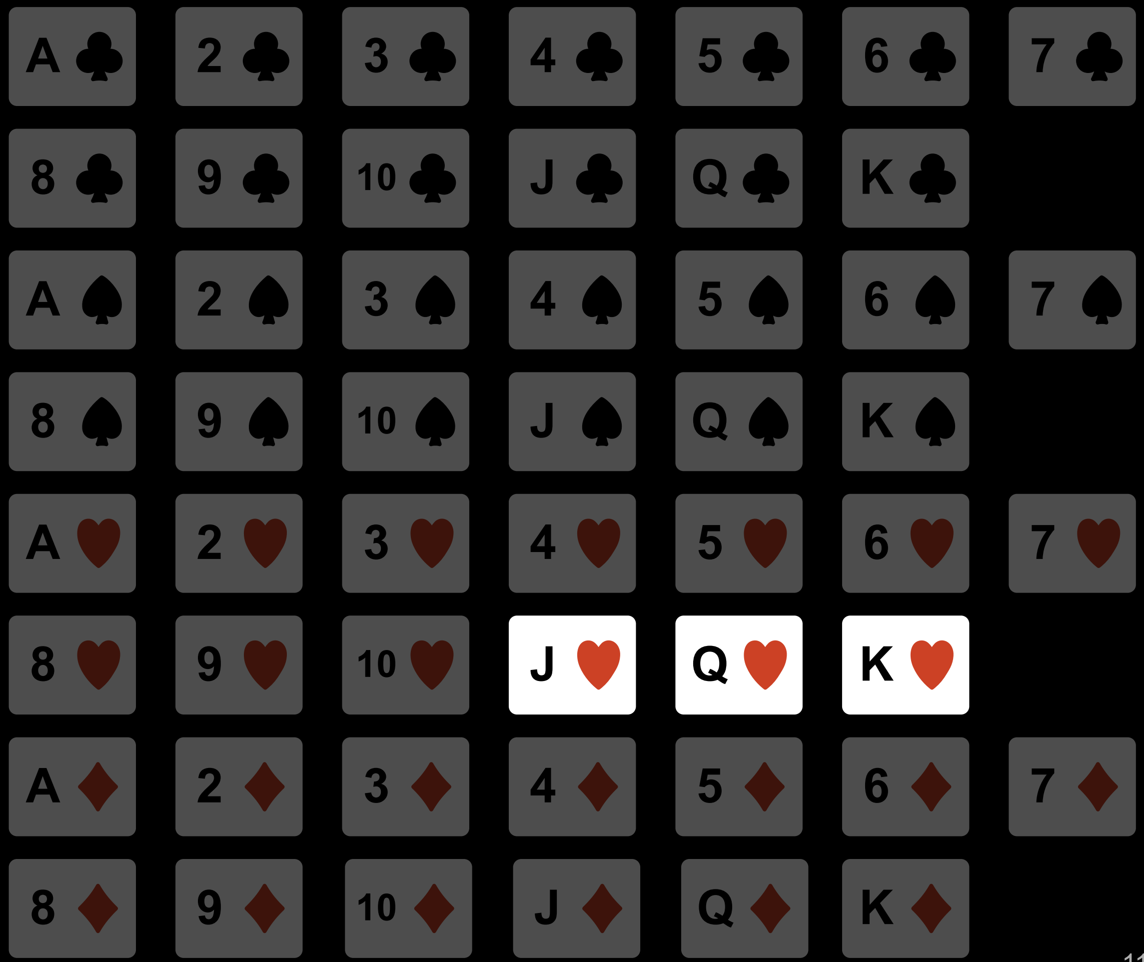
yes

is it 8 or above?

yes

is it jack or above?

yes



is the card black?

no

is it hearts?

yes

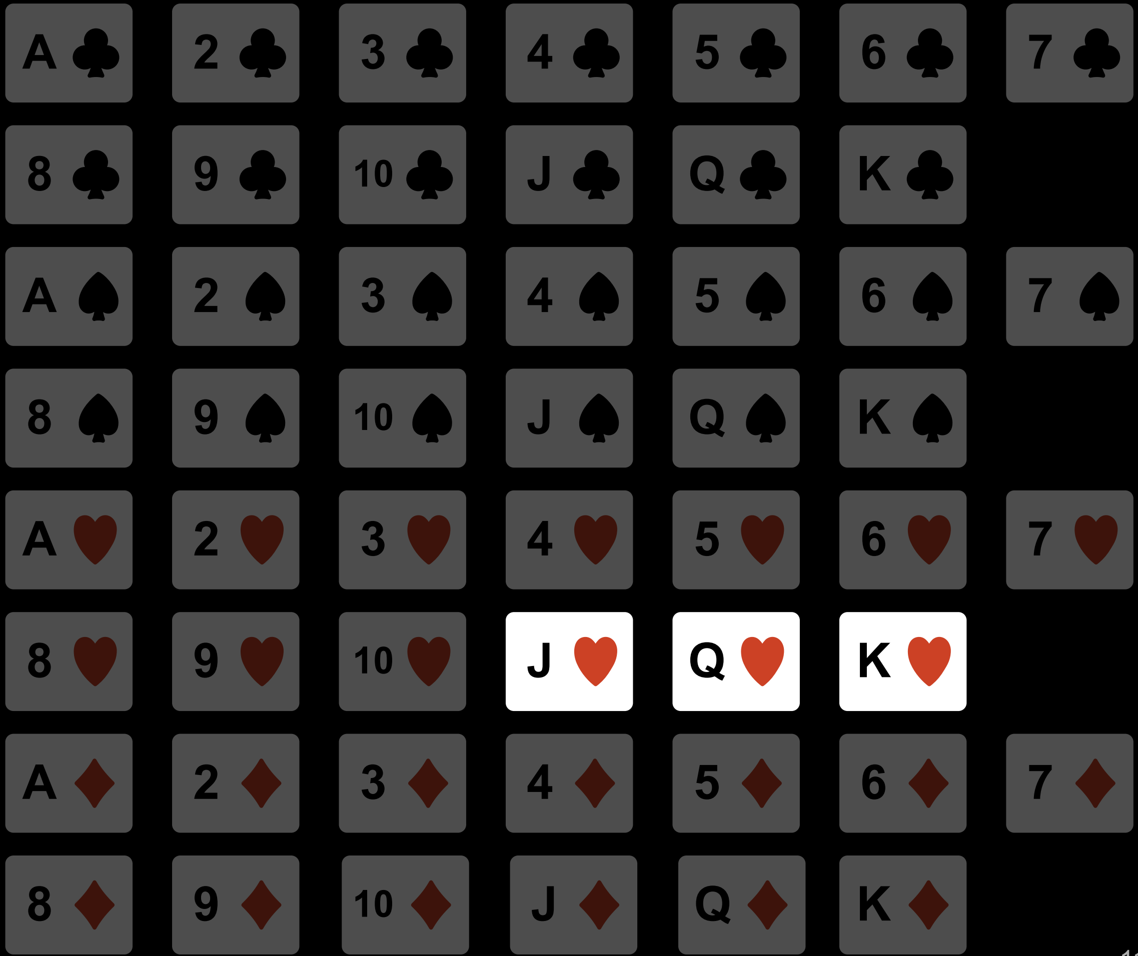
is it 8 or above?

yes

is it jack or above?

yes

is it queen or above?



is the card black?

no

is it hearts?

yes

is it 8 or above?

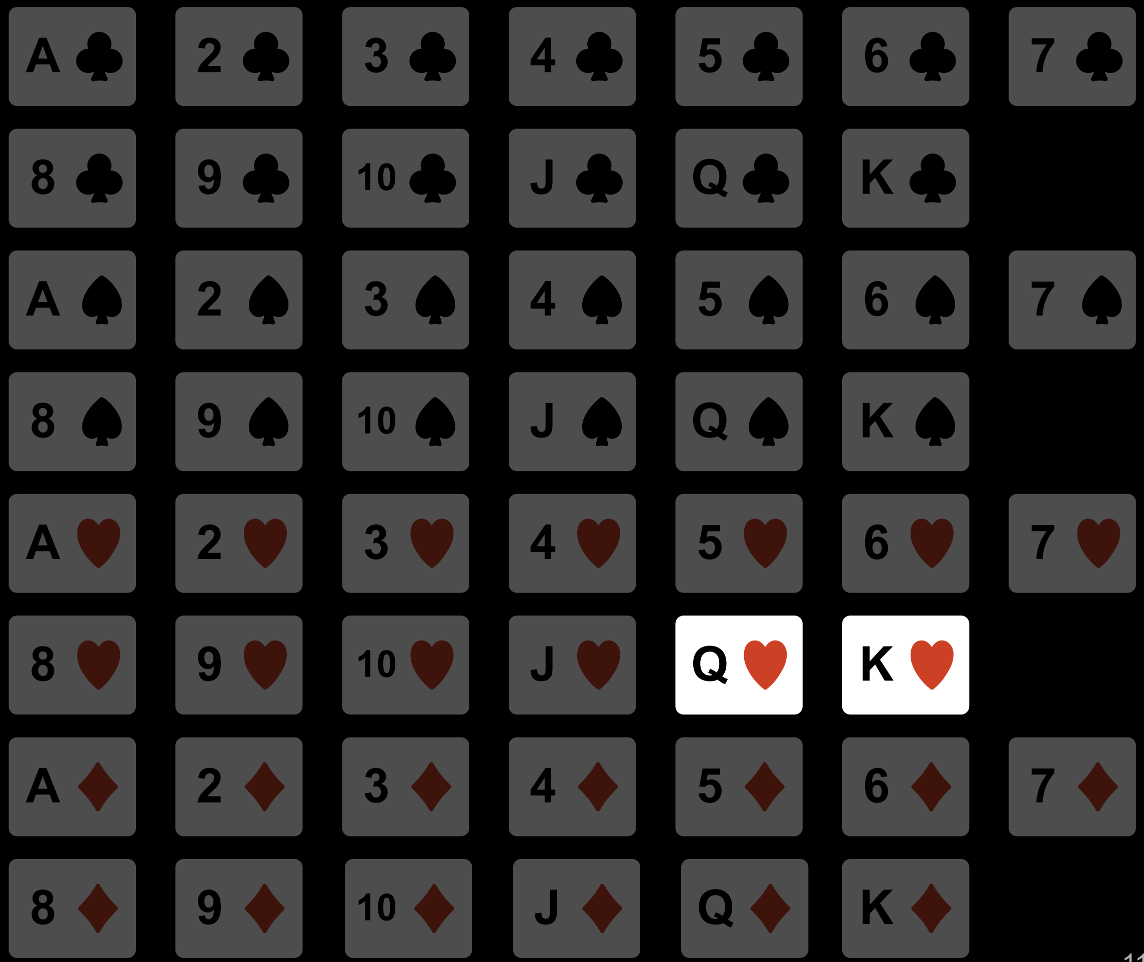
yes

is it jack or above?

yes

is it queen or above?

yes



is the card black?

no

is it hearts?

yes

is it 8 or above?

yes

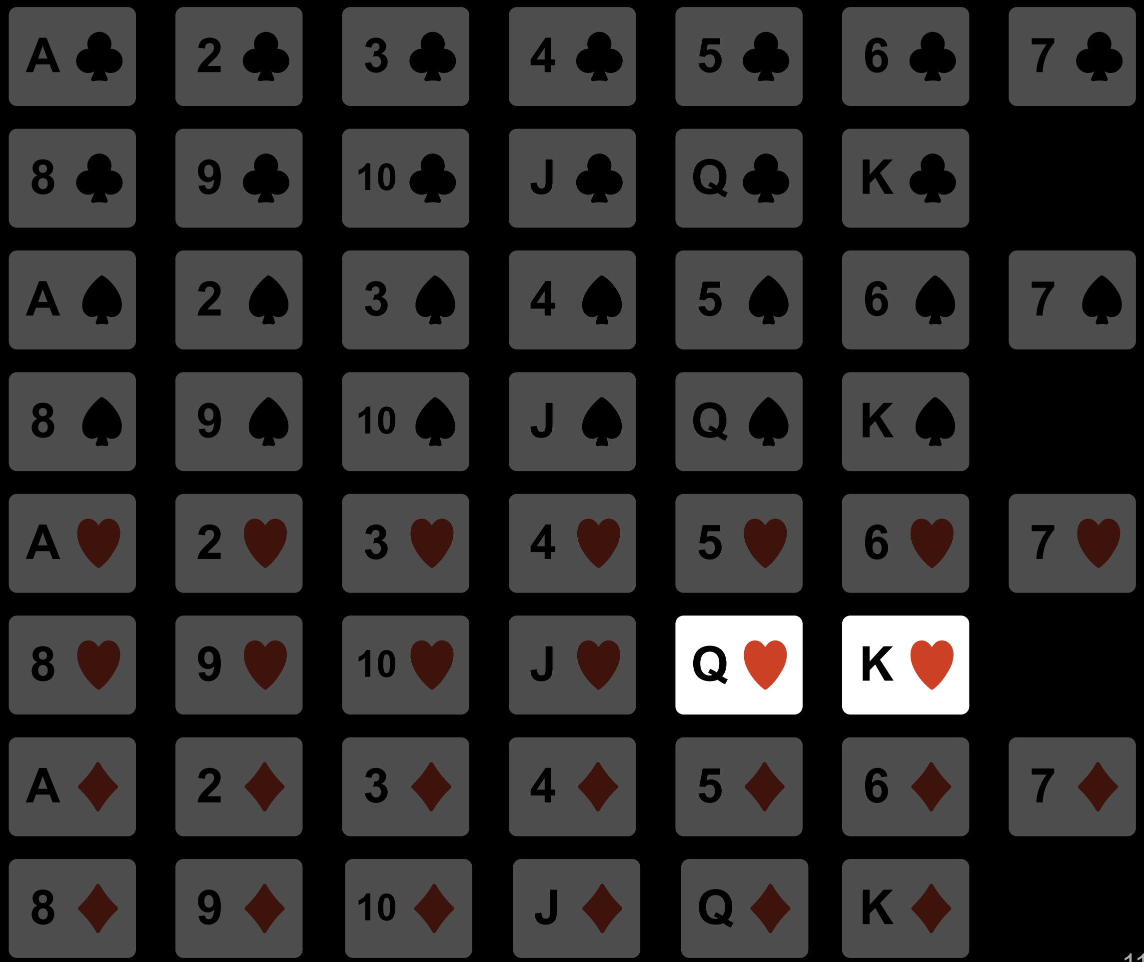
is it jack or above?

yes

is it queen or above?

yes

is it king?



is the card black?

no

is it hearts?

yes

is it 8 or above?

yes

is it jack or above?

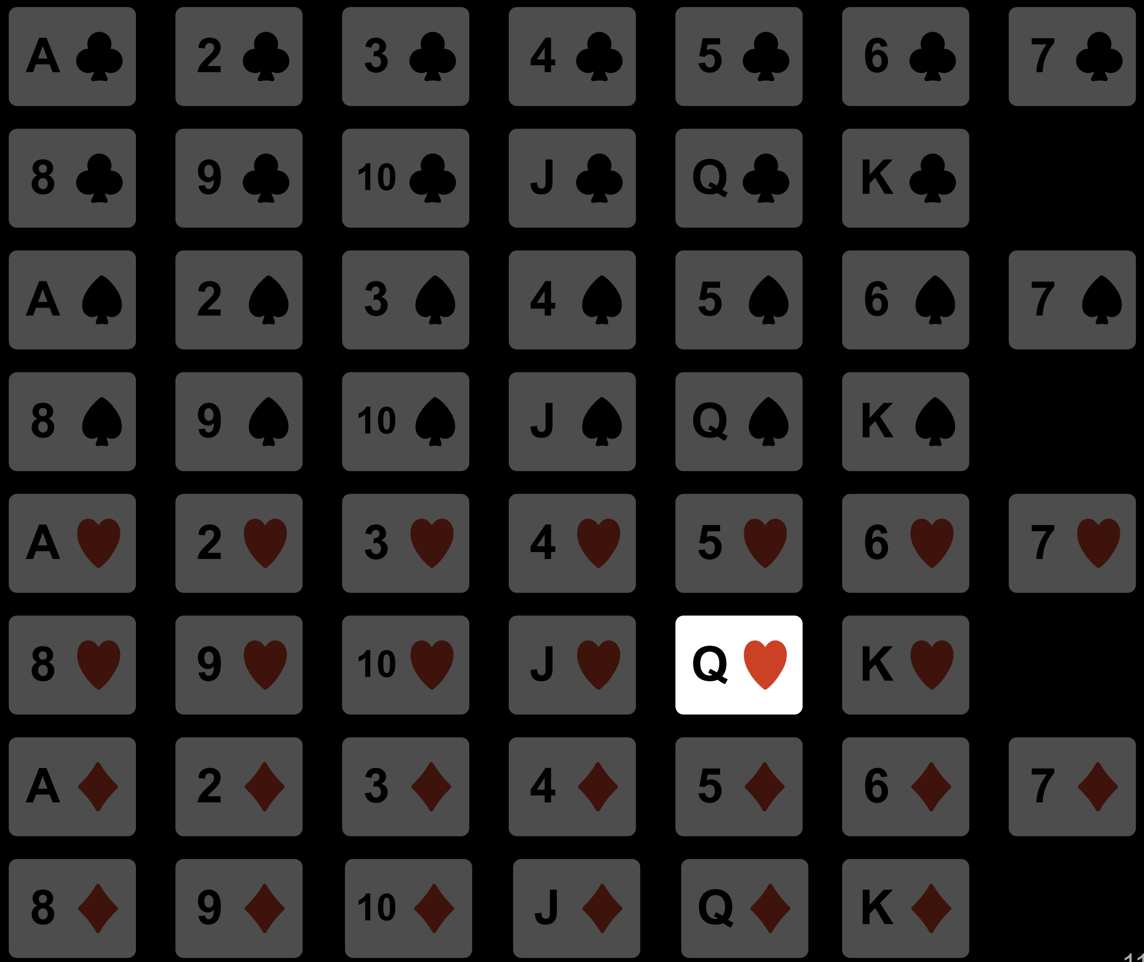
yes

is it queen or above?

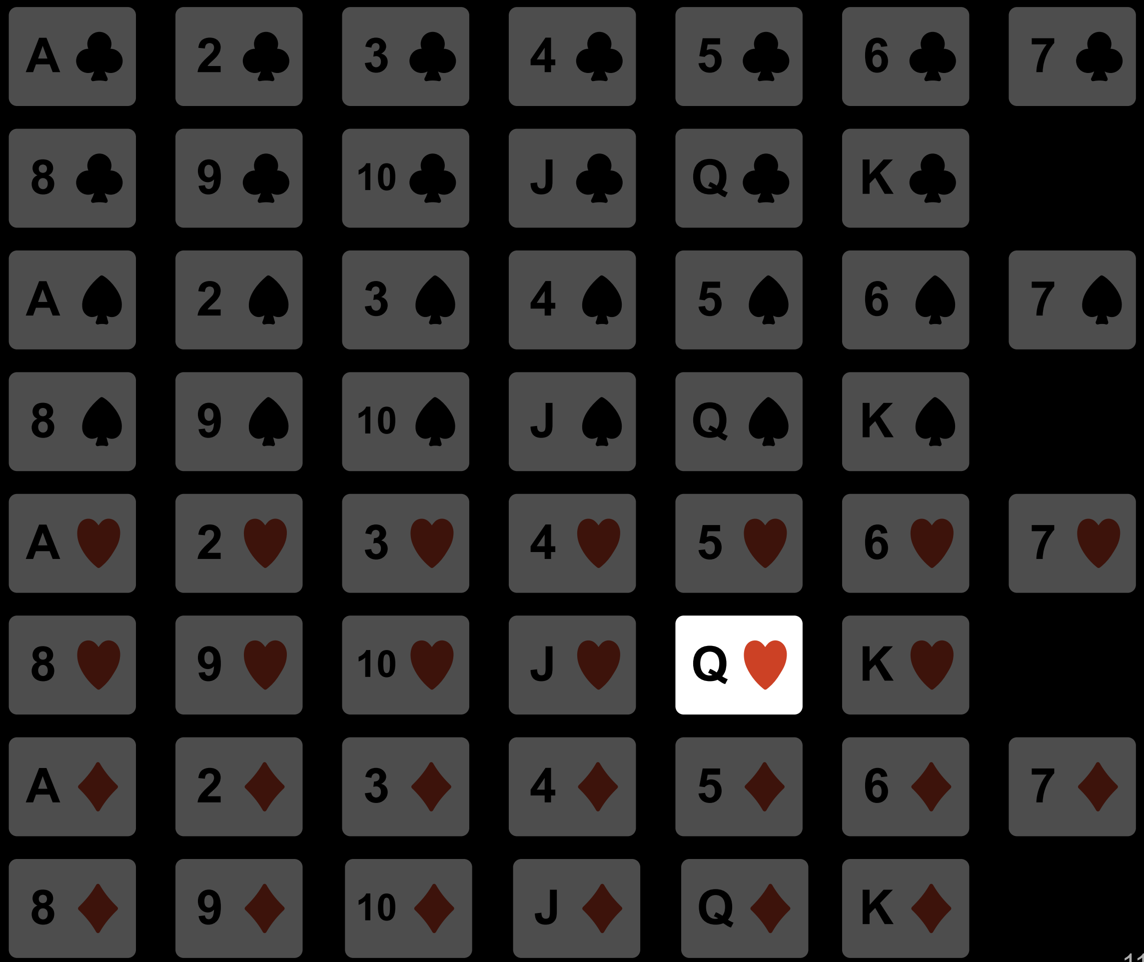
yes

is it king?

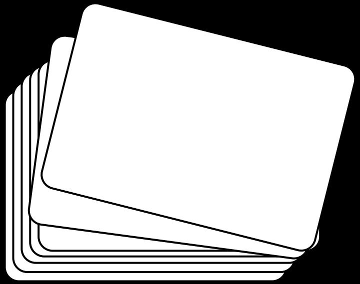
no



with 6 questions
from 52 to 1

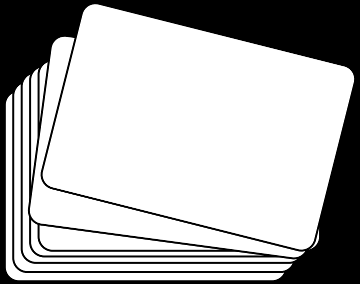


uncertainty with N = 52 possibilities?



$$H = log_2(52) \approx 5.7$$

uncertainty with N = 52 possibilities?



average # of yes/no questions

$$H = log_2(52) \approx 5.7$$

one bit of information with each answer…

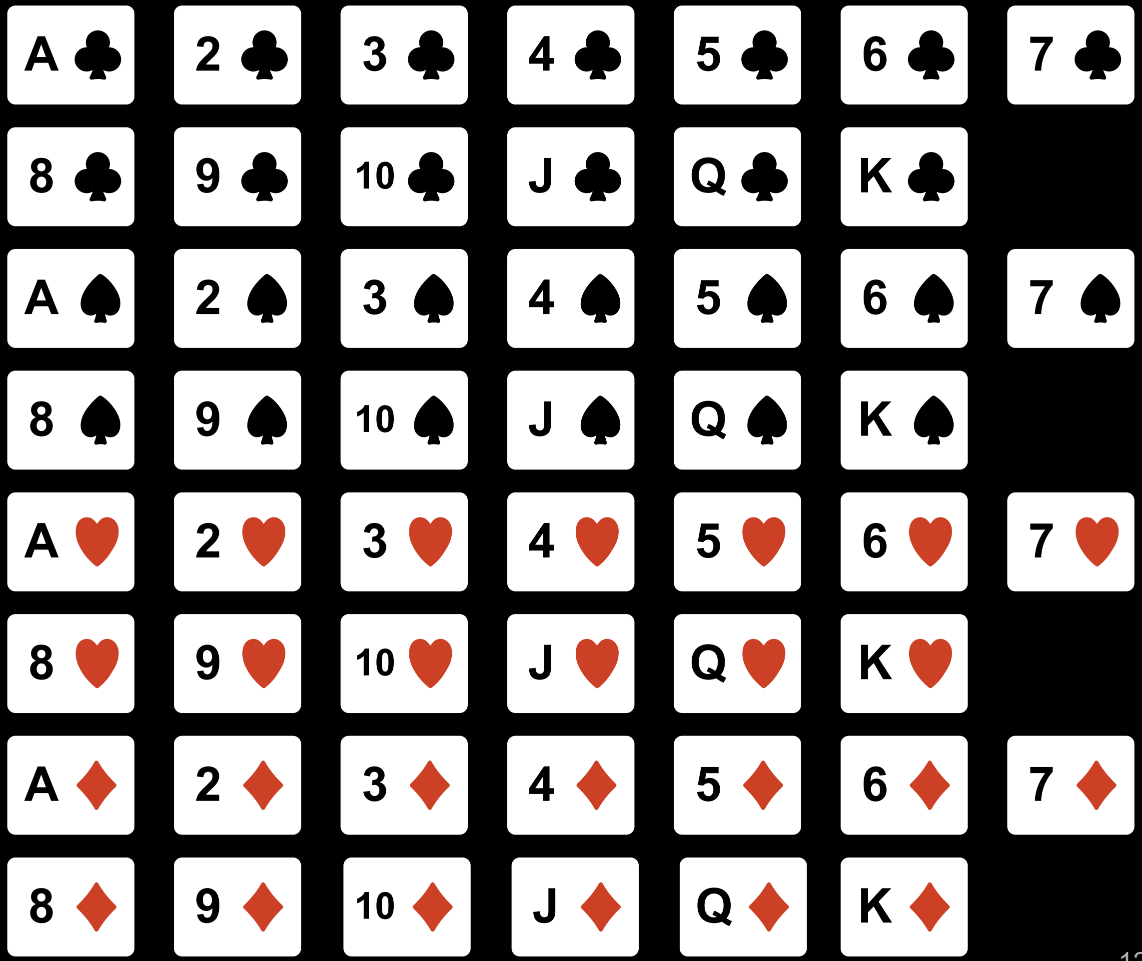
$$log_2(52) - log_2(26) = 1$$

one bit of information with each answer…

$$log_2(52) - log_2(26) = 1$$

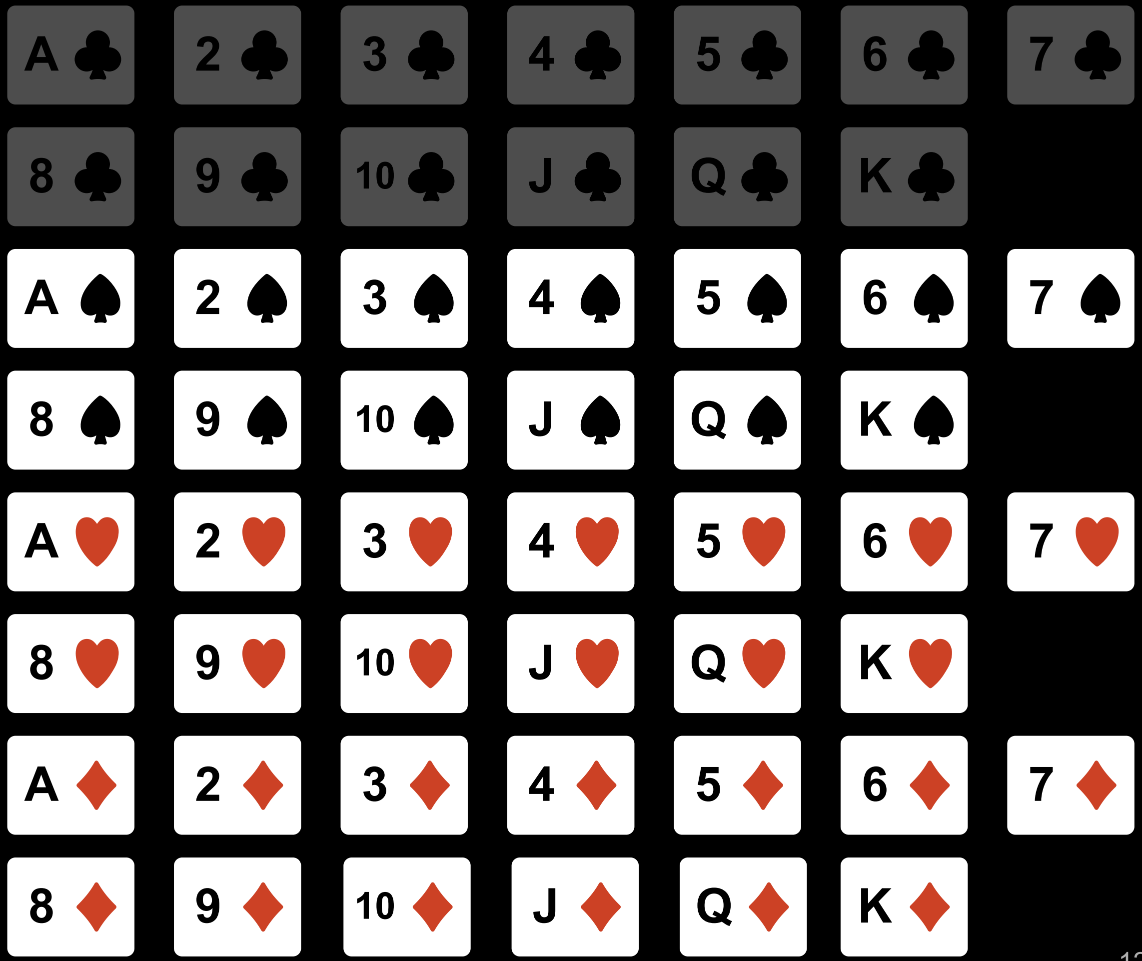
…that cuts the remaining options in half

is it a spades card?



is it a spades card?

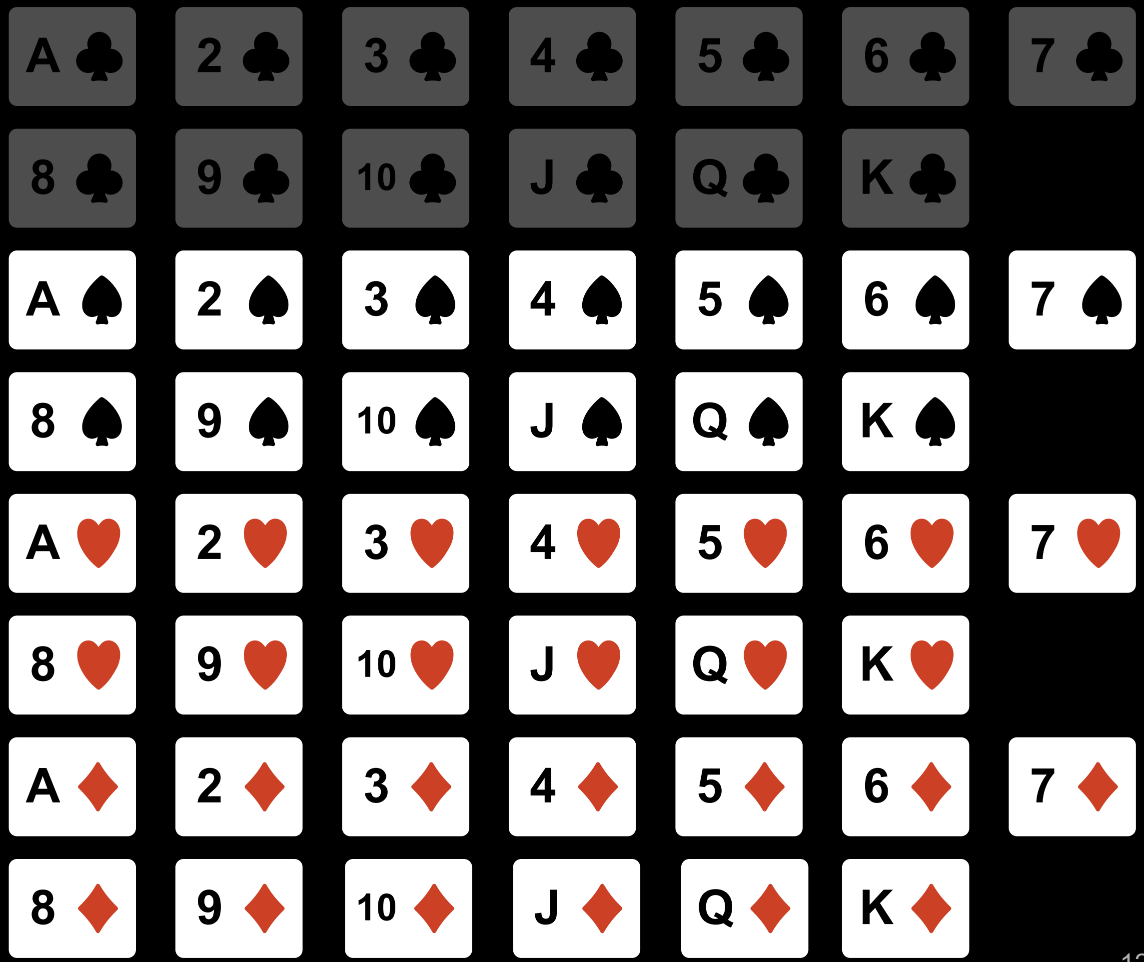
no



is it a spades card?

no

how much information?



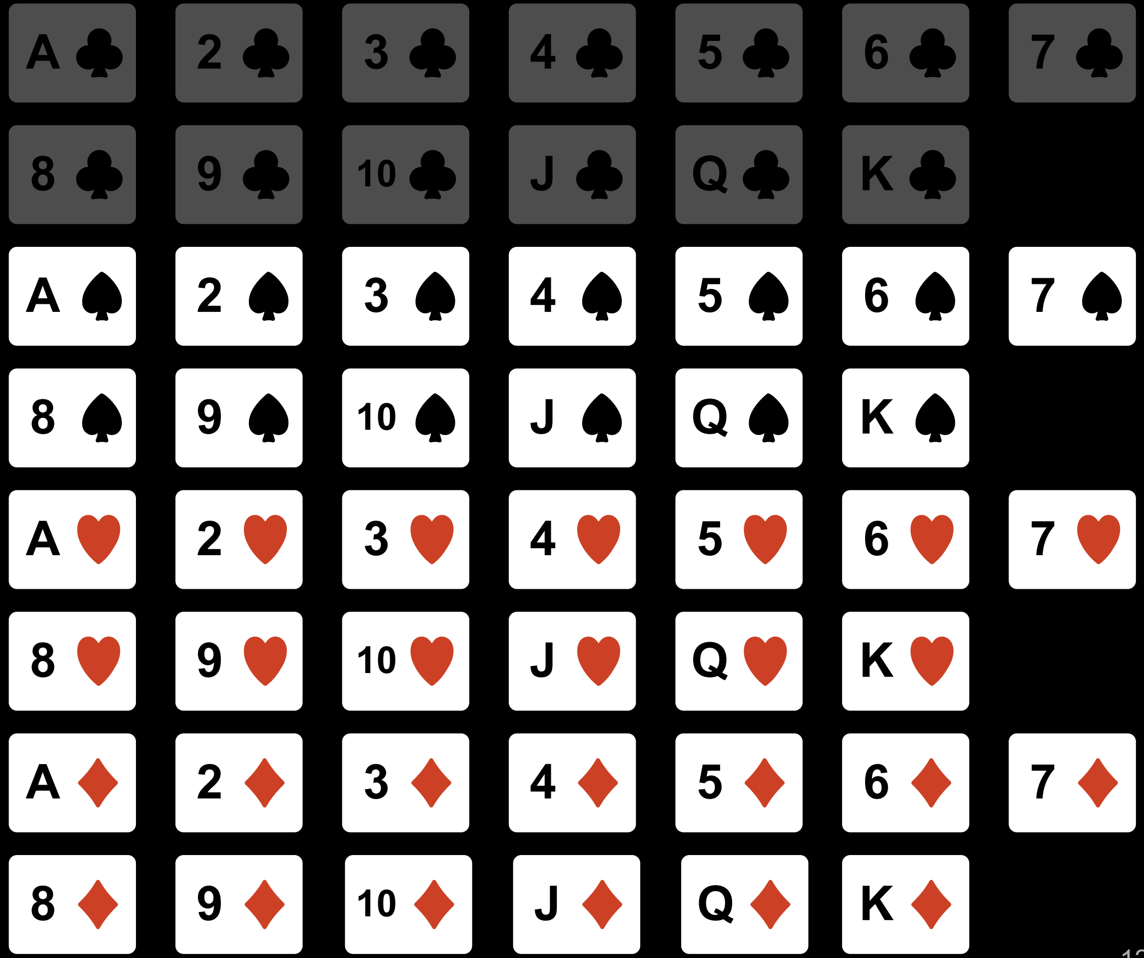
is it a spades card?

no

how much information?

$$H_0 = log_2(52) \approx 5.7$$

$$H_1 = log_2(39) \approx 5.29$$



is it a spades card?

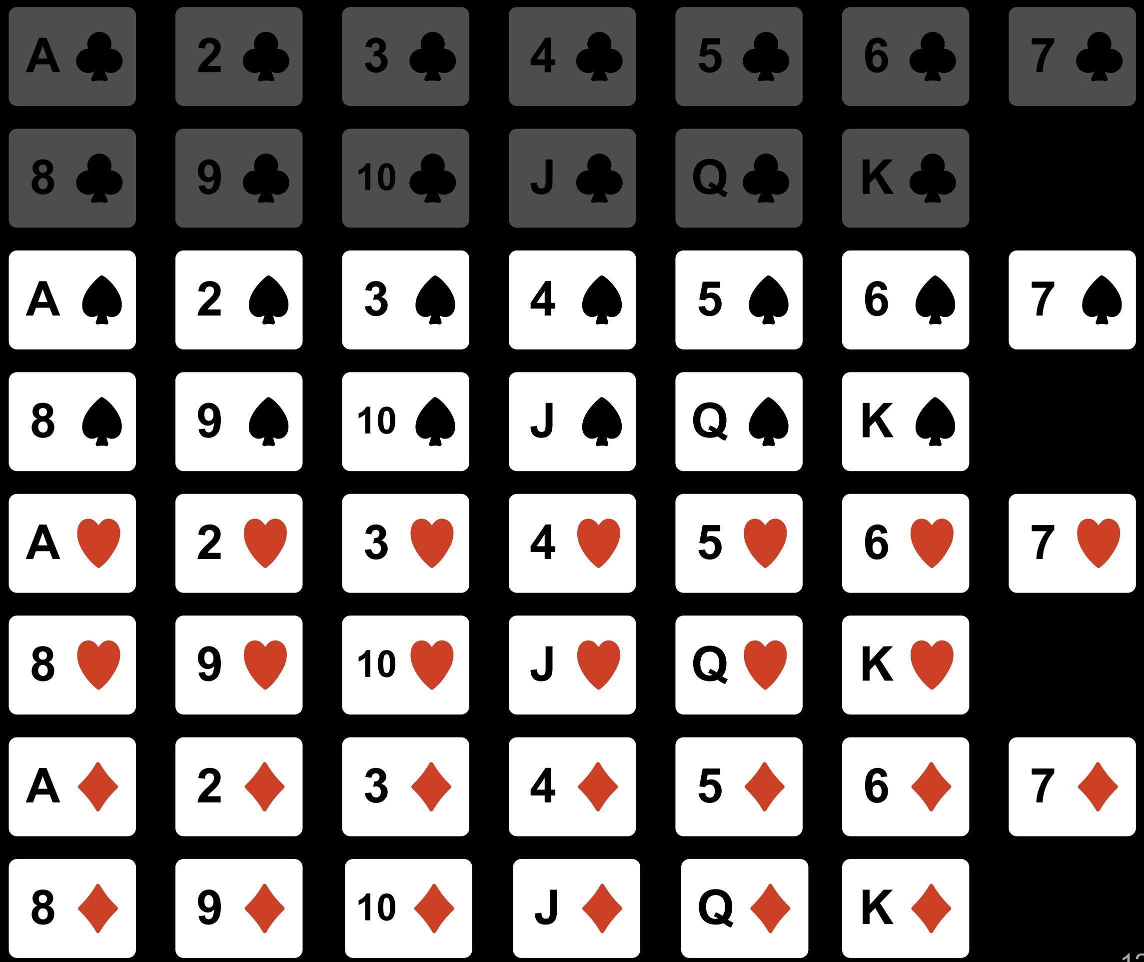
no

how much information?

$$H_0 = log_2(52) \approx 5.7$$

$$H_1 = log_2(39) \approx 5.29$$

$$H_0 - H_1 \approx 0.41$$



is it a spades card?

no

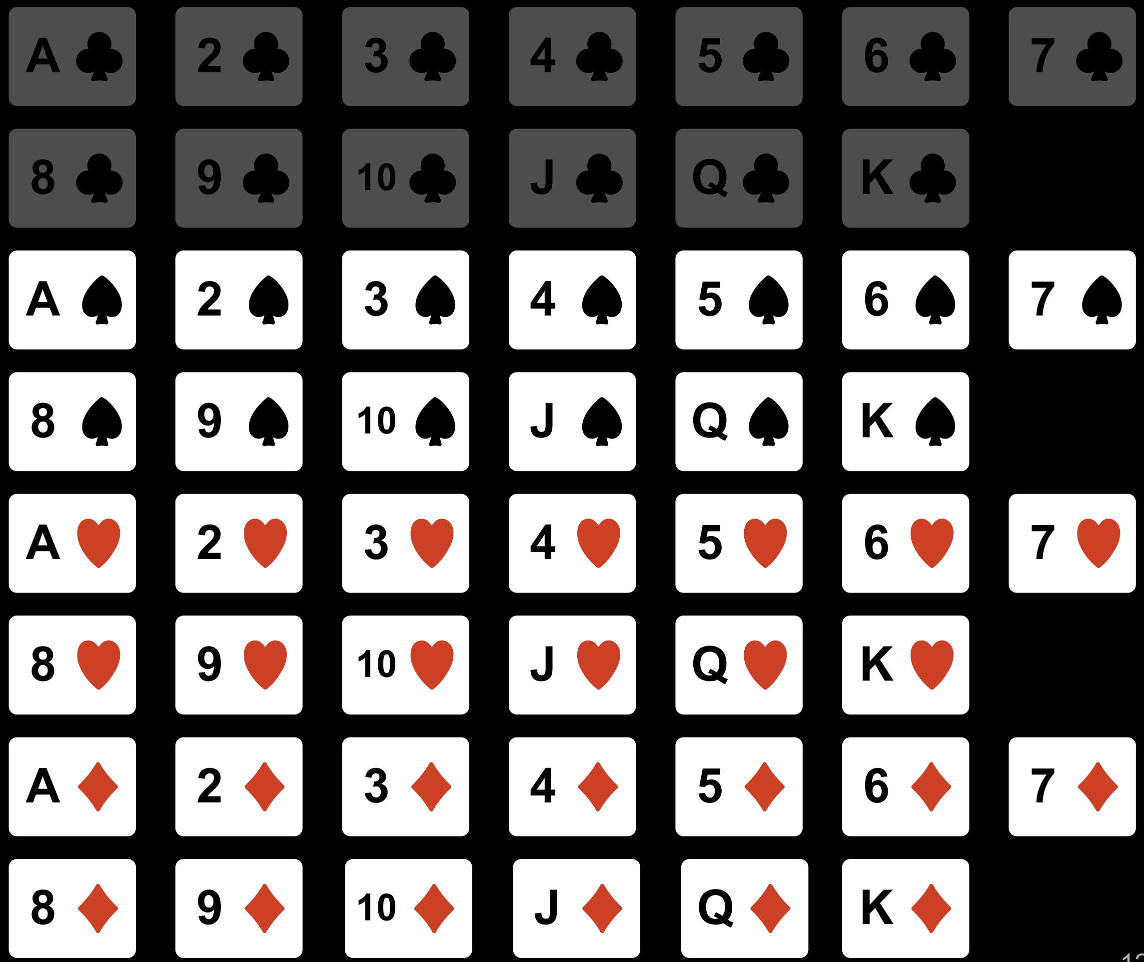
how much information?

$$H_0 = log_2(52) \approx 5.7$$

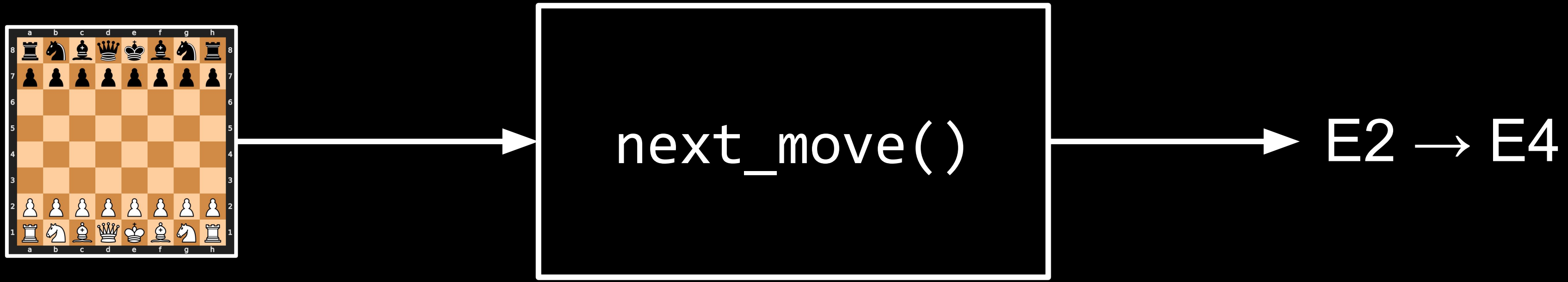
$$H_1 = log_2(39) \approx 5.29$$

$$H_0 - H_1 \approx 0.41$$

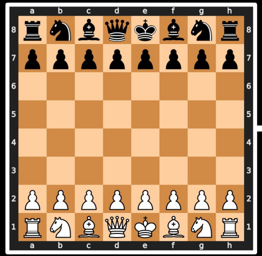
that's less than 1 bit



chess

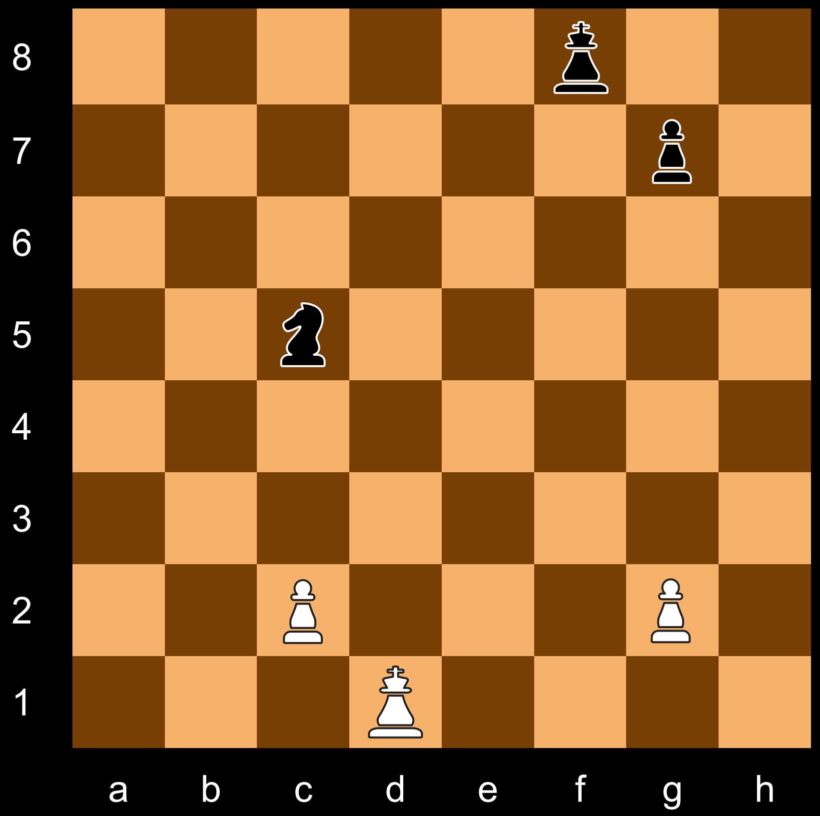

next_move()
E2 → E4

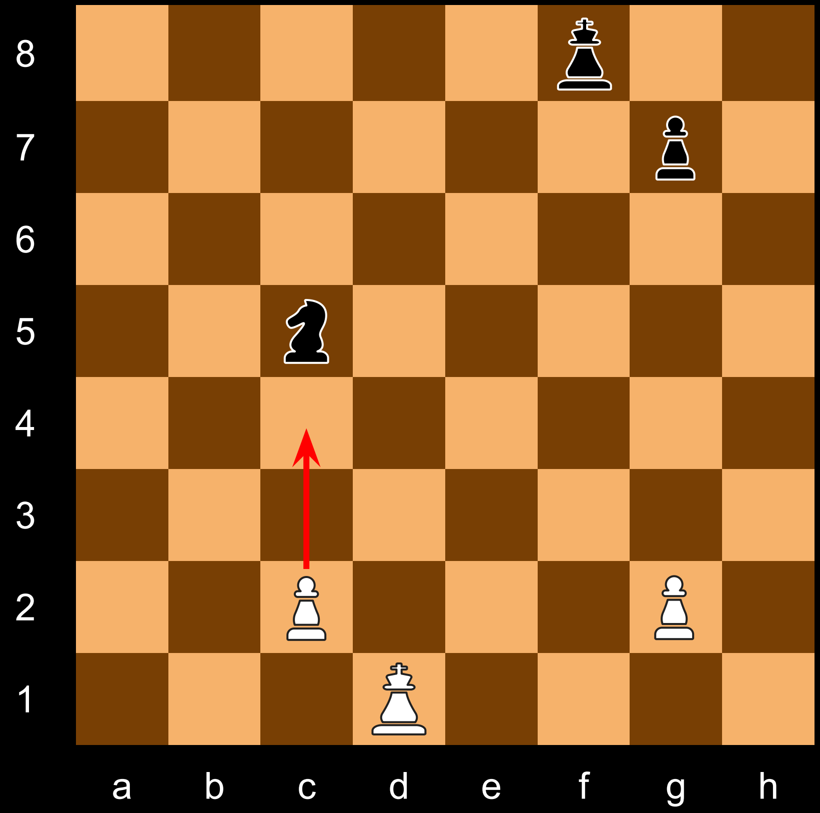
how much information is one move?

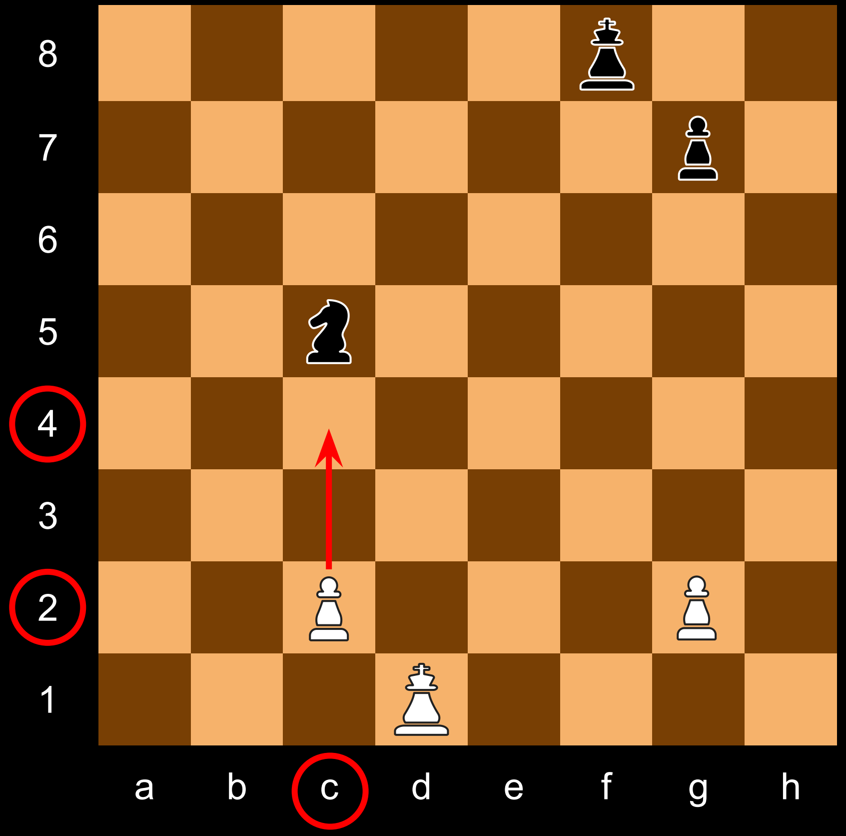


`next_move()`

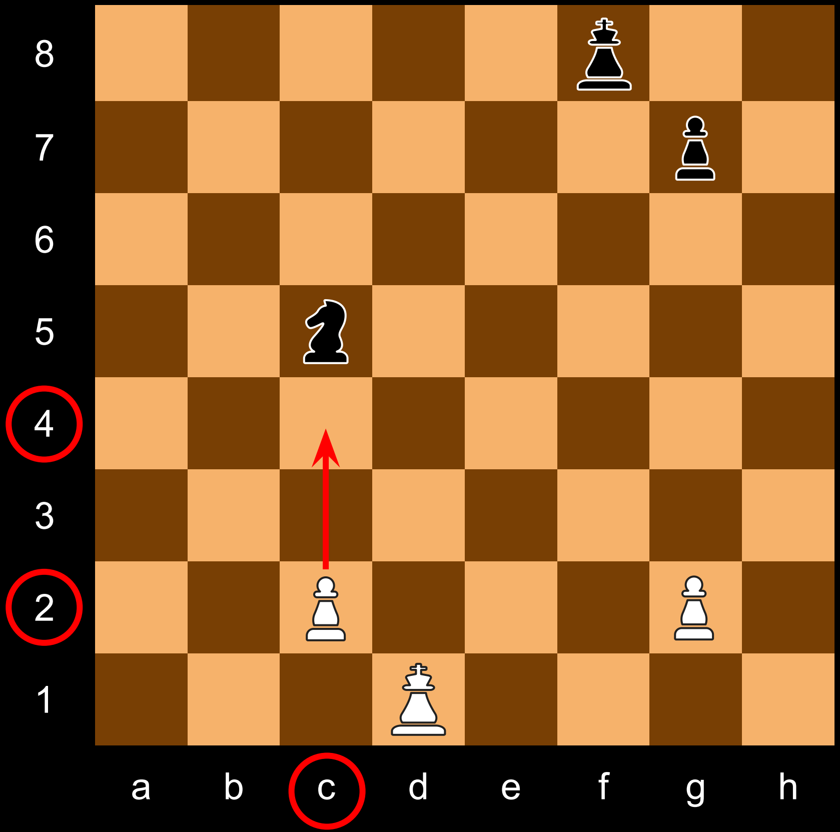
E2 → E4


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

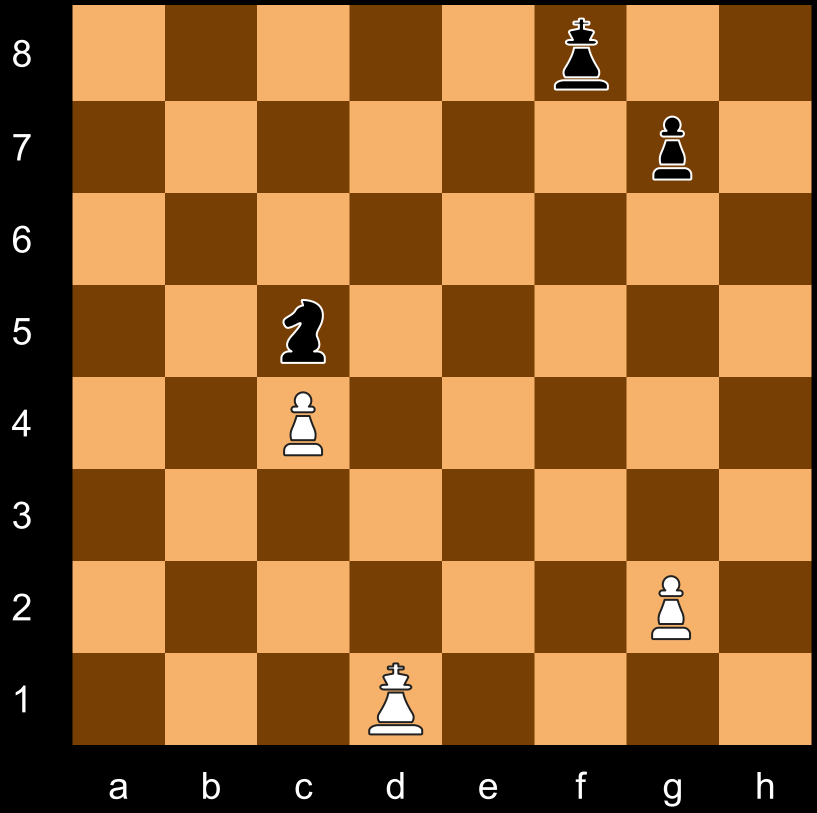

8
7
6
5
4
3
2
1
a b c d e f g h


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

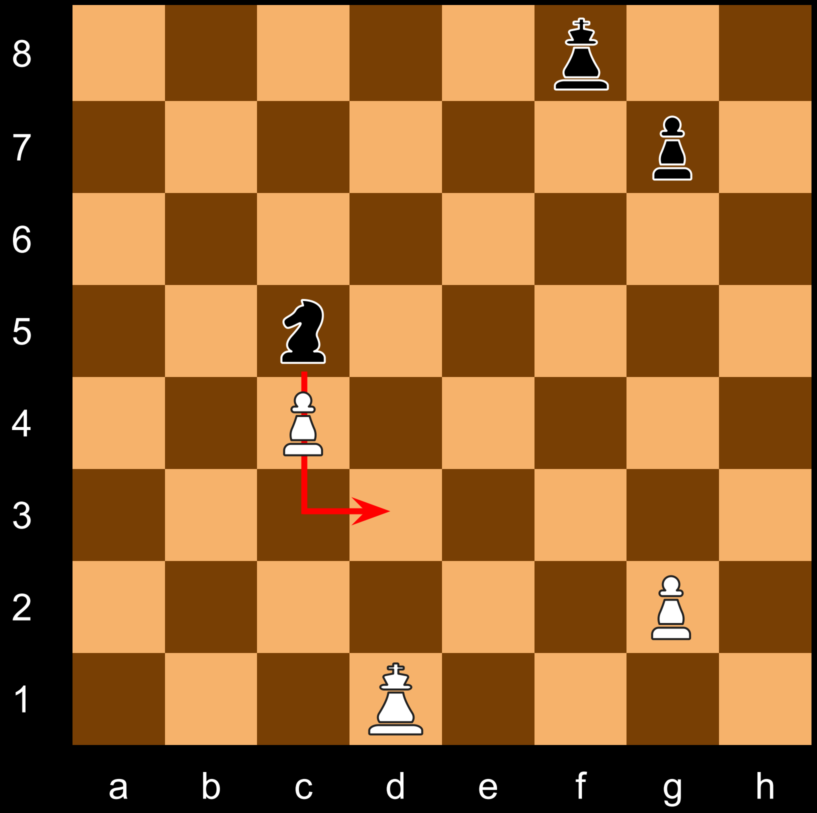
$c2 \rightarrow c4$



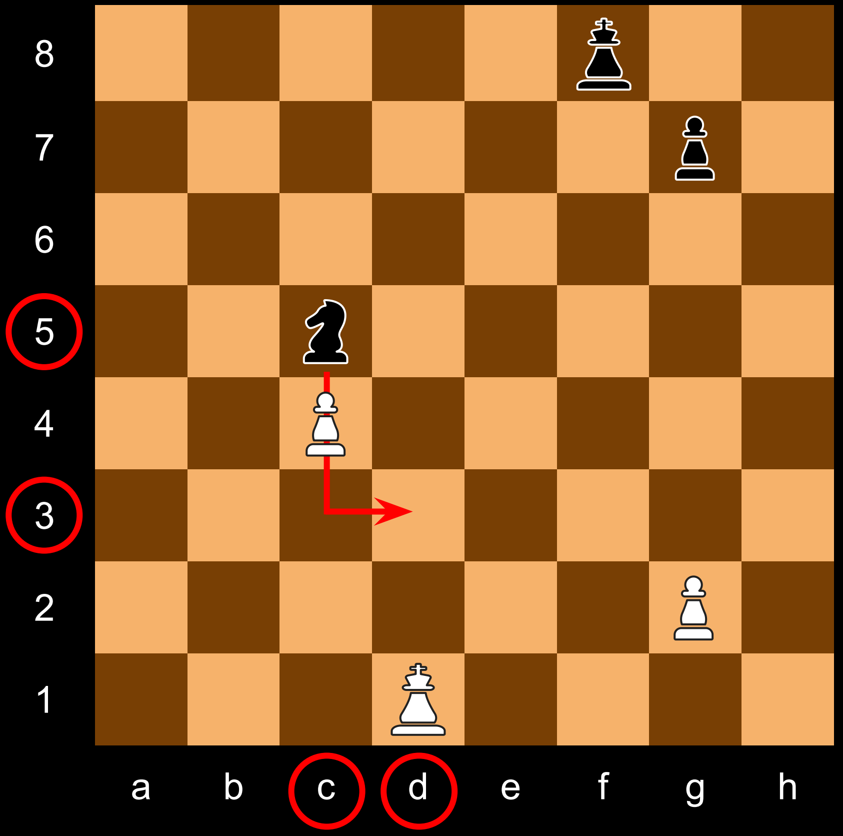
c2 → c4



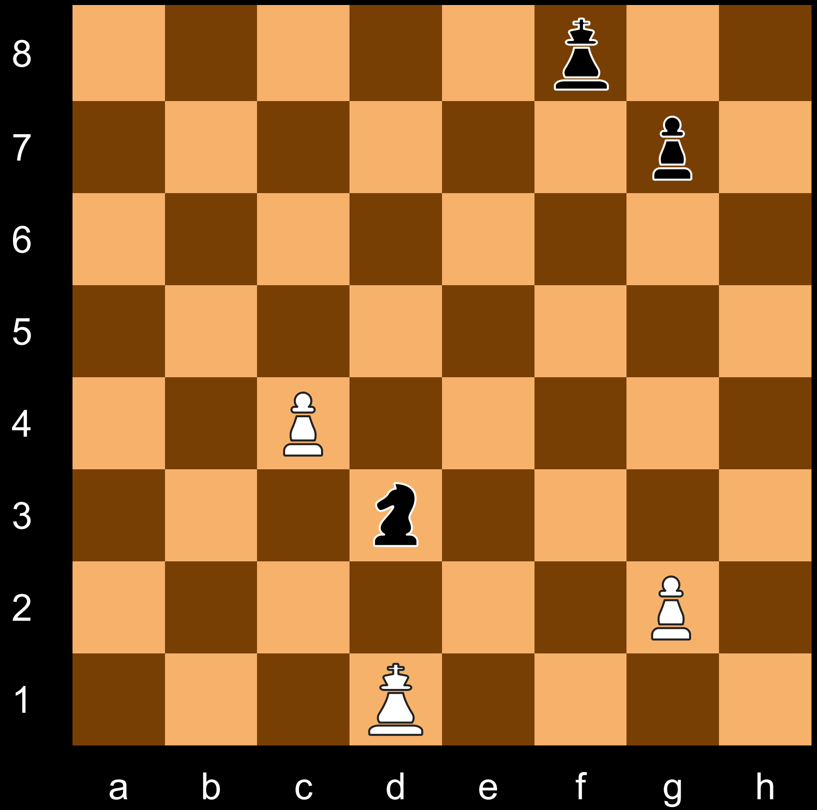
c2 → c4



c2 → c4
c5 → d3



c2 → c4
c5 → d3
…



c2 → c4

how many possibilities?

64 fields x 64 fields

how many possibilities?

$$64 \times 64 = 4096$$

possible moves*

*disregarding impossible moves

64 x 64 = 4096

$$H = log_2(4096) = 12$$

64 x 64 = 4096

$$H = log_2(4096) = 12$$

one chess move is 12 bits of information an alternative way to calculate # bits

c 2 c 4

4 digits

c 2 c 4

4 digits

8 possible symbols per digit

c 2 c 4

4 digits

8 possible symbols per digit

how many bits per digit?

c 2 c 4

4 digits

8 possible symbols per digit

how many bits per digit?

$$H_{digit} = log_2(8) = 3$$

c 2 c 4

4 digits

8 possible symbols per digit

how many bits per digit?

$$H_{digit} = log_2(8) = 3$$

$$H_{move} = log_2(8) \times 4 = 12$$

$$H_{avg} = log_2(S) \times n$$

S: number of possible symbols

n: number of digits in our message

$$H_{max} = \lceil log_2(S) \rceil \times n$$

when calculating bits for storage, we must always consider the worst case

digits and # symbols

{A}

_ _

{A}

A A

$\{A, B\}$

$\_\ \_$

$\{A, B\}$

AA, AB, BA, BB

$\{A, B, C\}$

_ _

{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 # digits n = 2

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

with length n = 2

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

and more digits?

$\{A, B\}$

___ ___ ___

{A, B}

AAA, AAB, ABA, ABB,
BBB, BBA, BAA, BAB

$\{A, B\}$

_ _ _ _

{A, B}

AAAA, AAAB, AABA, AABB,
ABAA, ABAB, ABBA, ABBB,
BAAA, BAAB, BABA, BABB,
BBAA, BBAB, BBBA, BBBB

with # symbols S = 2

# digits	# messages
1	2
2	4
3	8
4	16
5	32

with # symbols S = 2

# digits		# messages
1		2
2		4
3	$\xrightarrow{f(x)}$	8
4		16
5		32

with # symbols S = 2

# digits		# messages
1		2
2	f(x)	4
3	$\longrightarrow$	8
4		16
5		32

possible messages with *n* digits and *S* symbols

$$N = S^n$$

COUNTING

1 2 3

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

$$\frac{1 \quad\quad 2 \quad\quad 3}{}$$

$$10^2 \quad\quad 10^1 \quad\quad 10^0$$

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

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

$$= 123$$

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

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

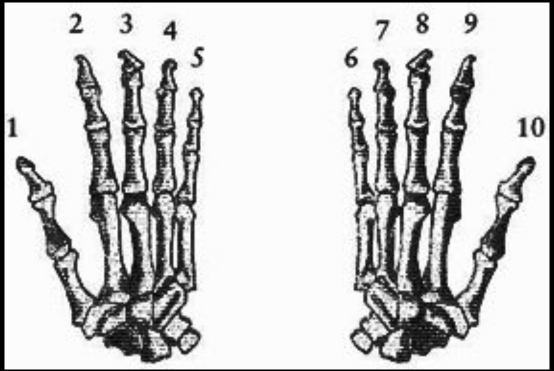
$$= 4 \times 1000 + 1 \times 100 + 2 \times 10 + 3 \times 1$$

$$\begin{array}{cccc} 4 & 1 & 2 & 3 \\ \hline ? & 10^2 & 10^1 & 10^0 \end{array}$$

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

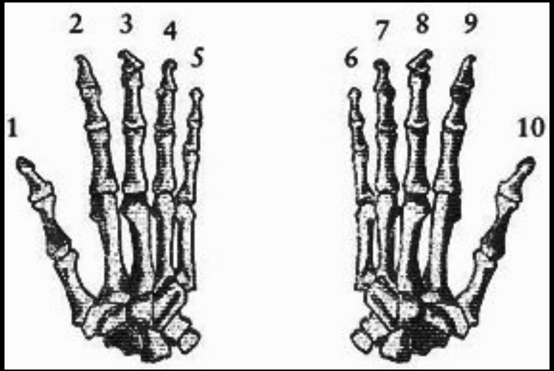
$$= 4 \times 1000 + 1 \times 100 + 2 \times 10 + 3 \times 1$$

$$= 4123$$

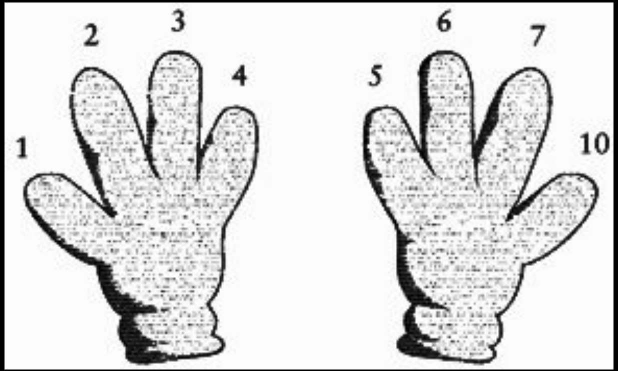


human hand

Image source: Petzold, Charles. Code: The Hidden Language of Computer Hardware and Software. 2nd ed., Microsoft Press, 2022.



human hand



cartoon character’s hand

Image source: Petzold, Charles. Code: The Hidden Language of Computer Hardware and Software. 2nd ed., Microsoft Press, 2022.

1 2 3 (octal)

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

$$1 \quad 2 \quad 3 \quad \text{(octal)}$$

$$8^2 \quad 8^1 \quad 8^0$$

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

$$1 \quad 2 \quad 3$$ (octal)

$$8^2 \quad 8^1 \quad 8^0$$

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

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

$$1 \quad 2 \quad 3 \quad \text{(octal)}$$

$$8^2 \quad 8^1 \quad 8^0$$

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

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

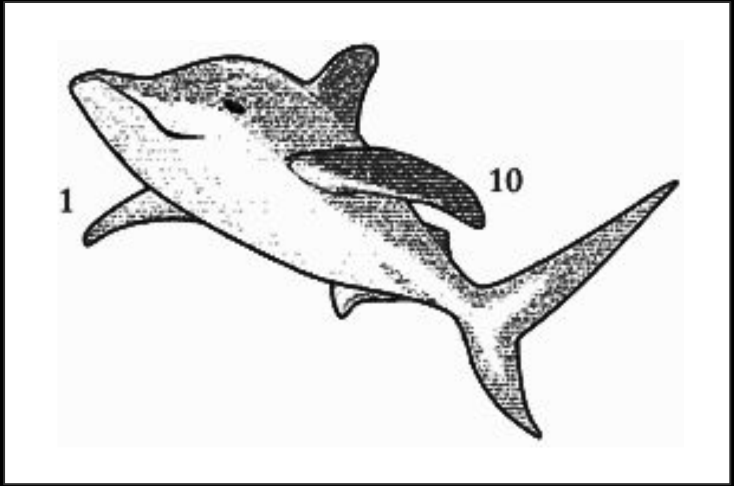
$$= 83 \text{ (decimal)}$$

decimal		octal
8	→	?

decimal		octal
?	←	7

decimal		octal
16	→	?

decimal		octal
?	←	100



what now?

Image source: Petzold, Charles. Code: The Hidden Language of Computer Hardware and Software. 2nd ed., Microsoft Press, 2022.

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	

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

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

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

$$1 \quad 1 \quad 0 \quad \text{(binary)}$$

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

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

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

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

$$2^2 \qquad 2^1 \qquad 2^0$$

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

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

$= 6$ (decimal)

2 3 4 5 6

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

place value systems

$$N = d_n * R^{n-1} + \ldots + d_2 * R^1 + d_1 * R^0$$

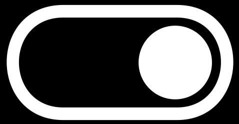
$d \in \{ 0, 1, \ldots R-1 \}$

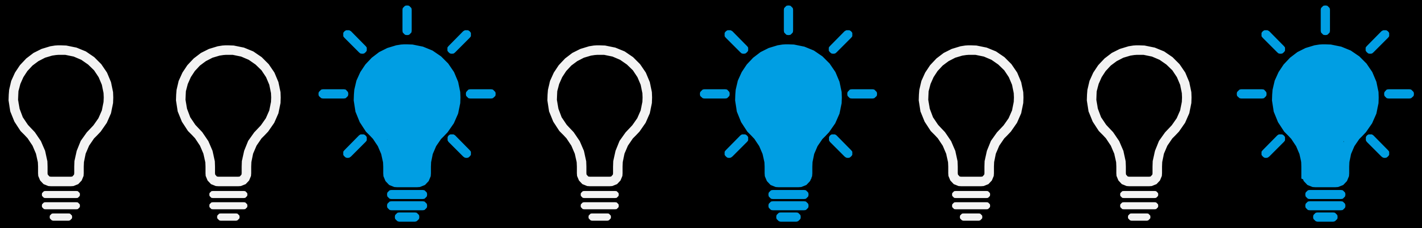
n = number of digits
R = base

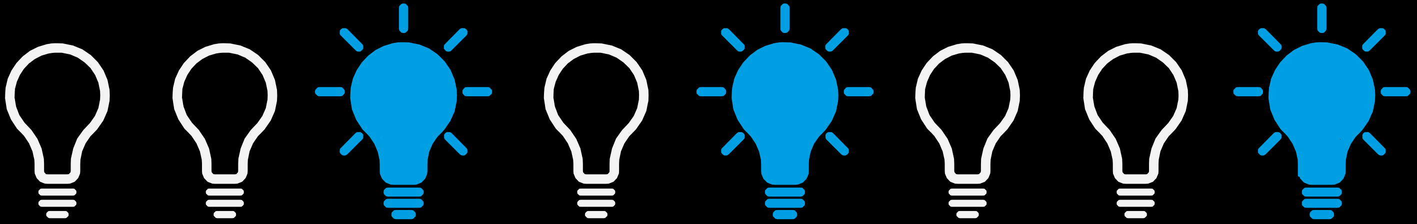
$$R \geq 2$$

BITS

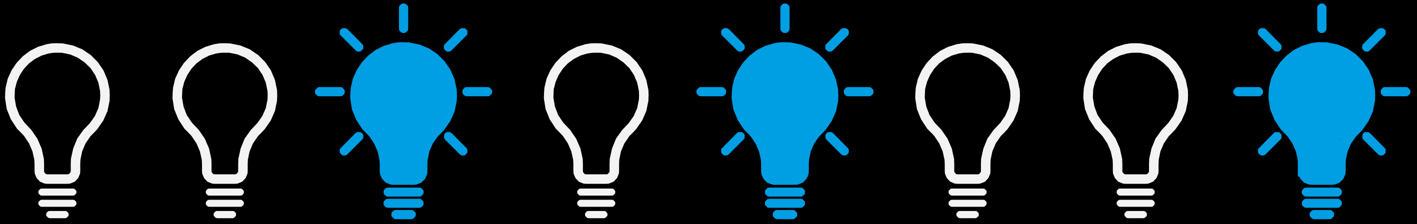
why do computers think binary?





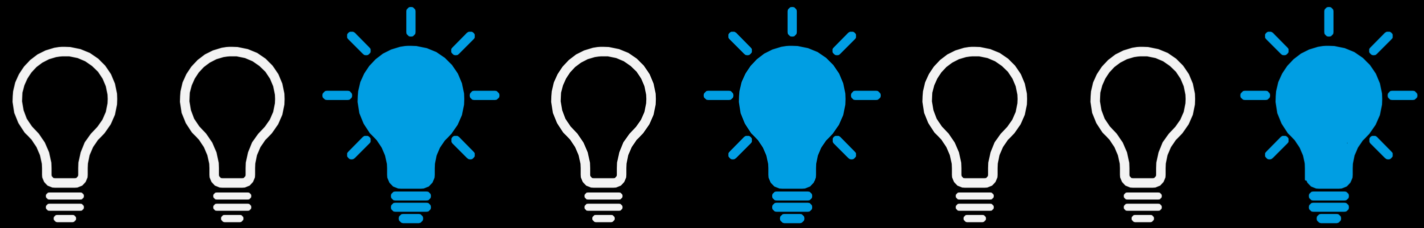


0 0 1 0 1 0 0 1



0 0 1 0 1 0 0 1

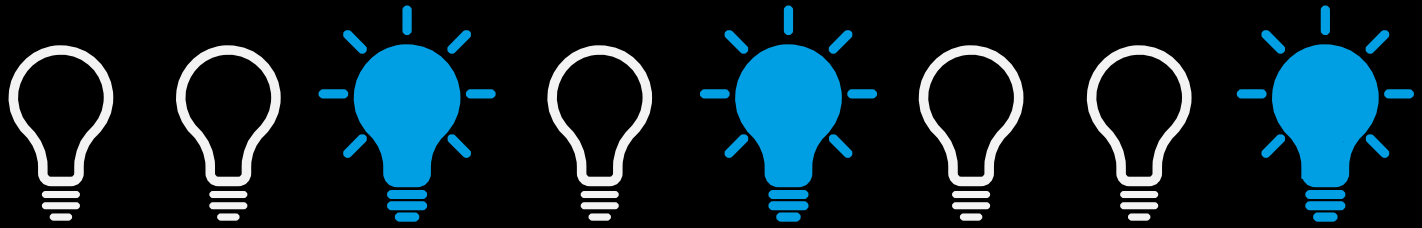
a **bit** (binary digit)



0 0 1 0 1 0 0 1

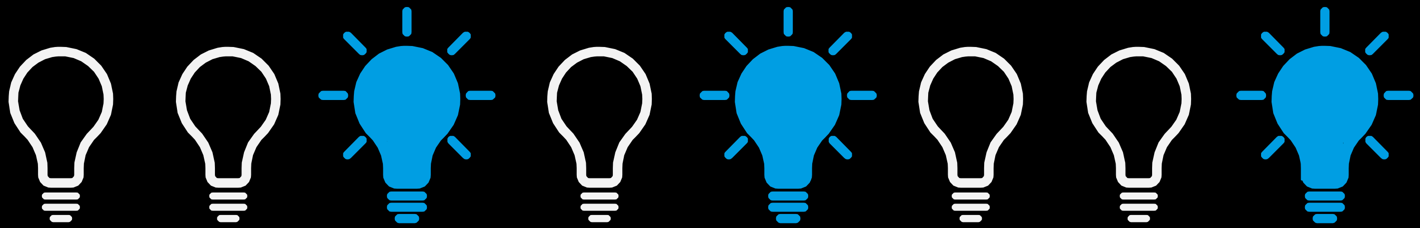
a **bit** (binary digit)

a **byte** (8 bits)

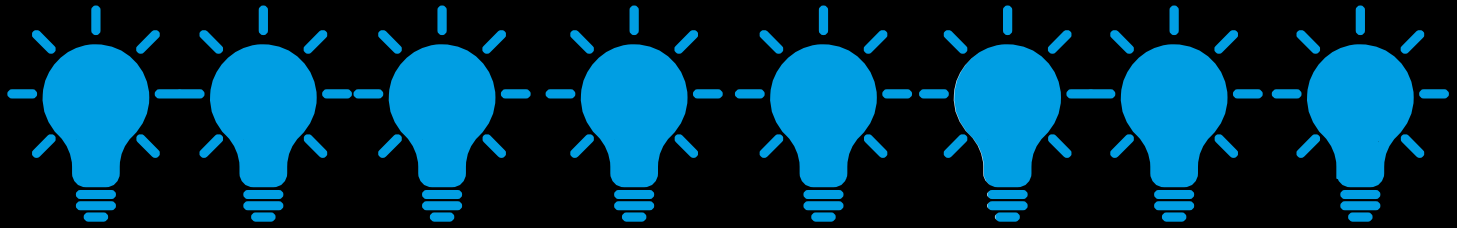


0 0 1 0 1 0 0 1

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?

what comes after the byte?

2^{10} bytes = 1.024 bytes = 1 Kibibyte (KiB)

2^{20} bytes = 1.048.576 bytes = 1 Mebibyte (MiB)

2^{30} bytes = 1.073.741.824 bytes = 1 Gibibyte (GiB)

10^3 bytes = 1.000 bytes = 1 Kilobyte (KB)

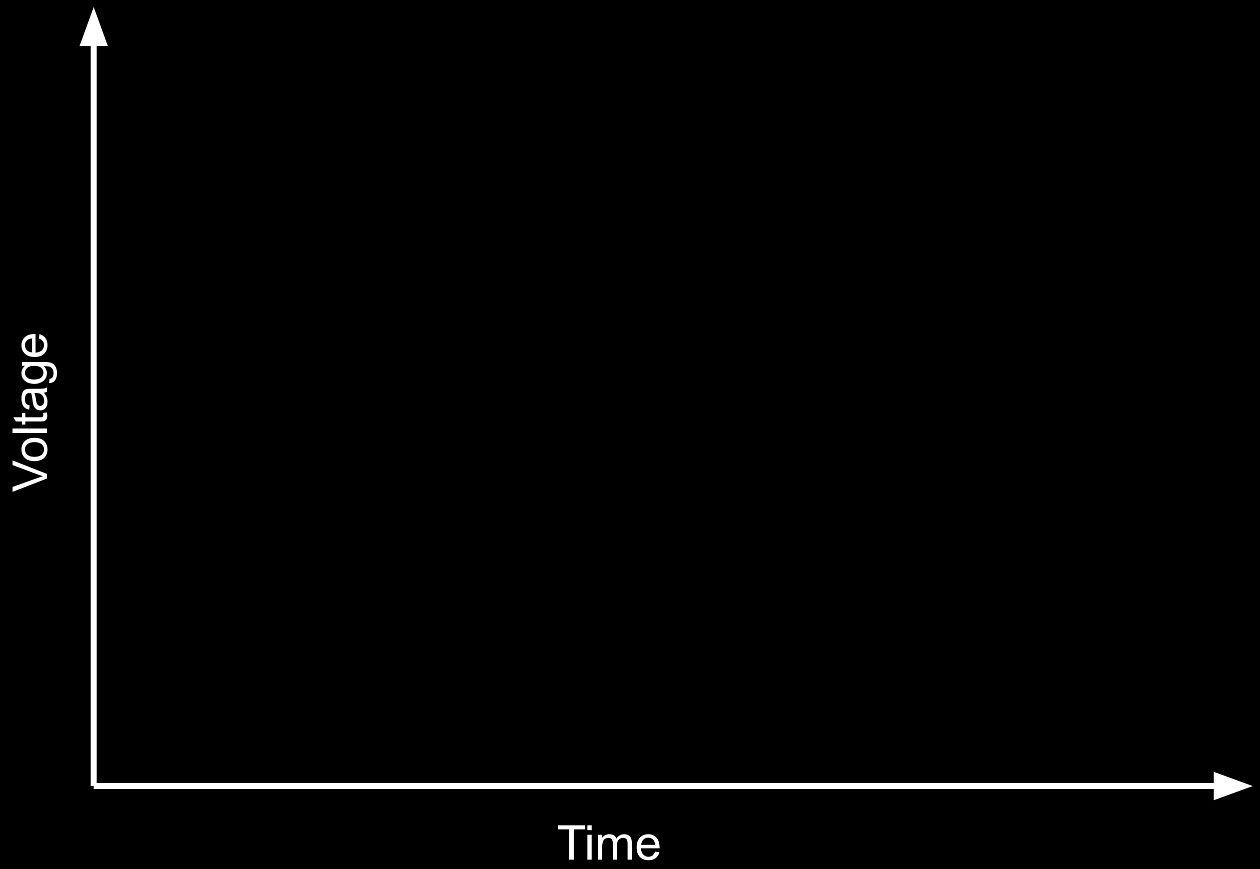
10^6 bytes = 1.000.000 bytes = 1 Megabyte (MB)

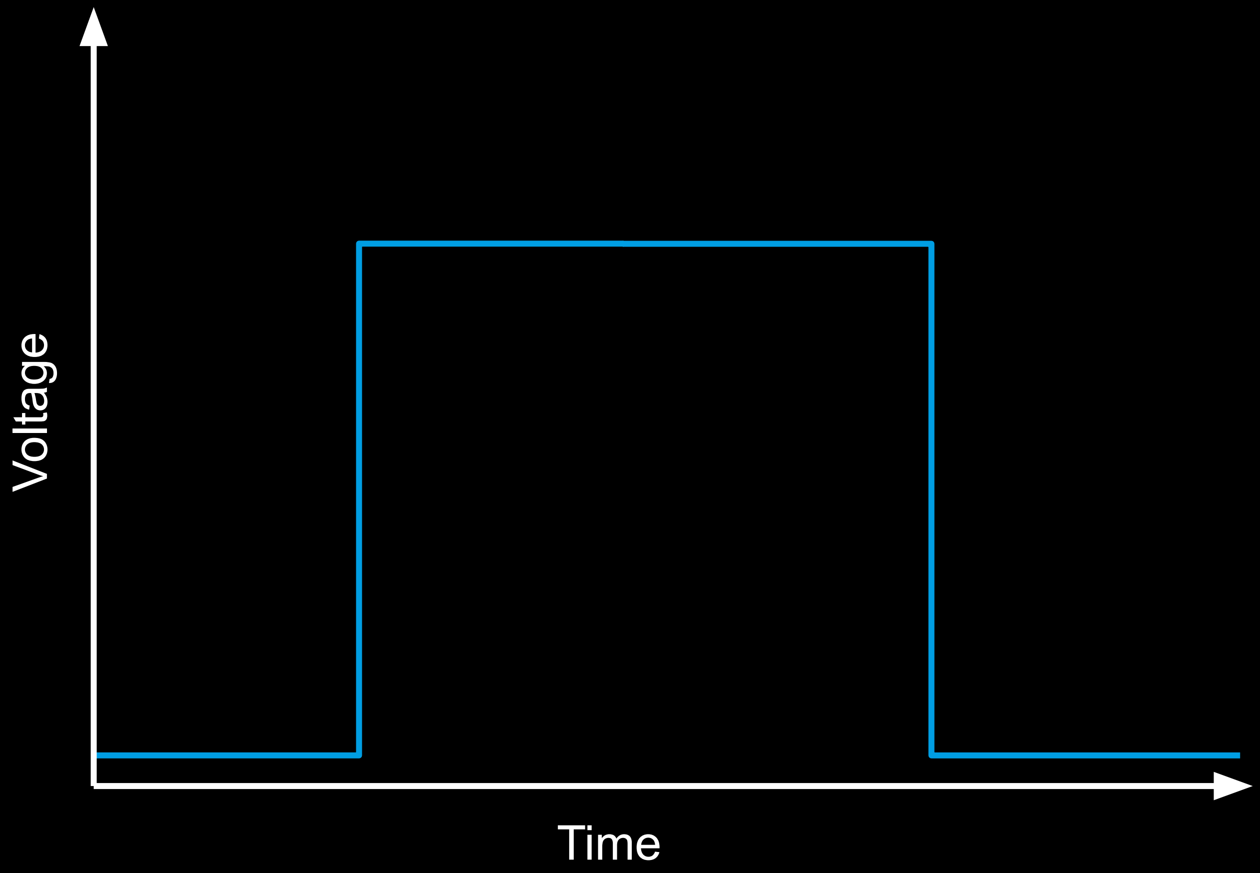
10^9 bytes = 1.000.000.000 bytes = 1 Gigabyte (GB)

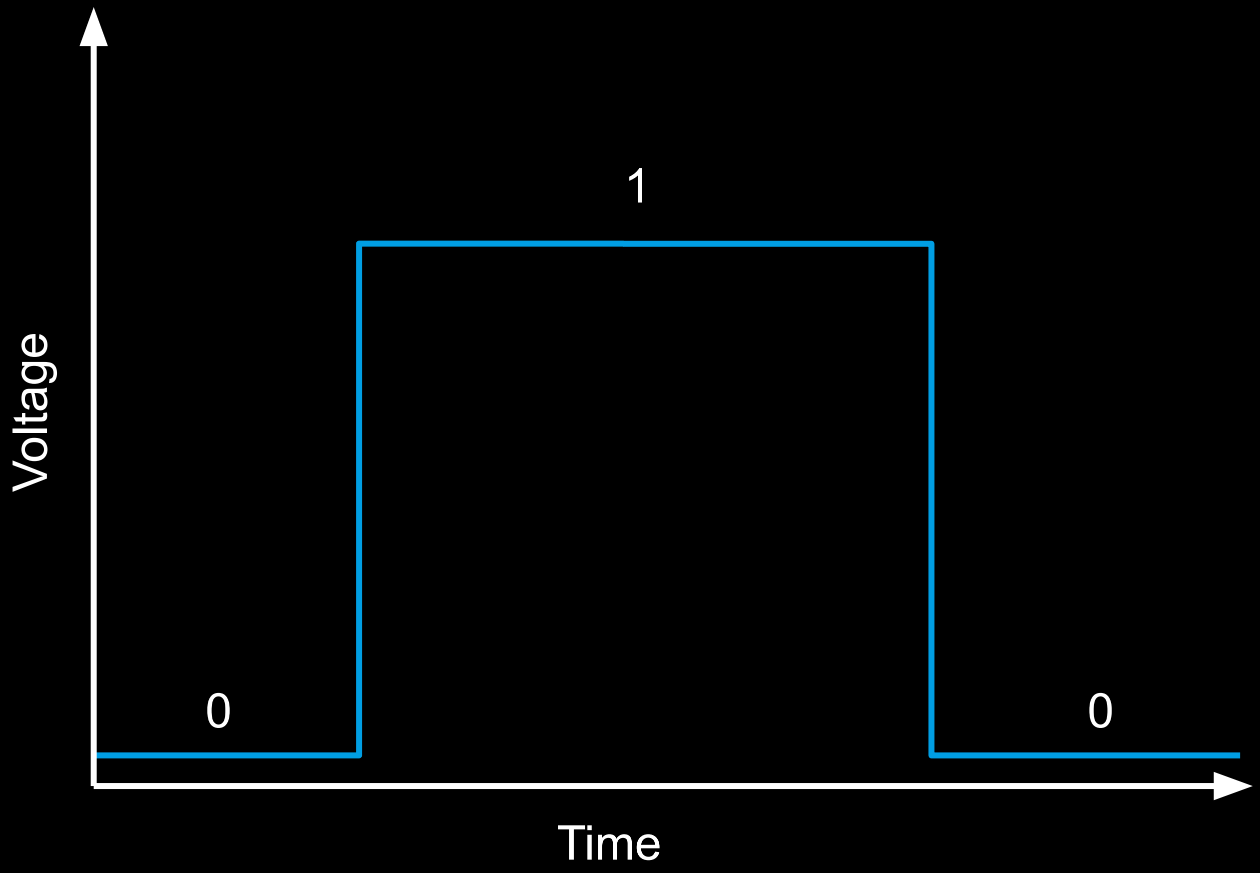
10^{12} bytes = ?

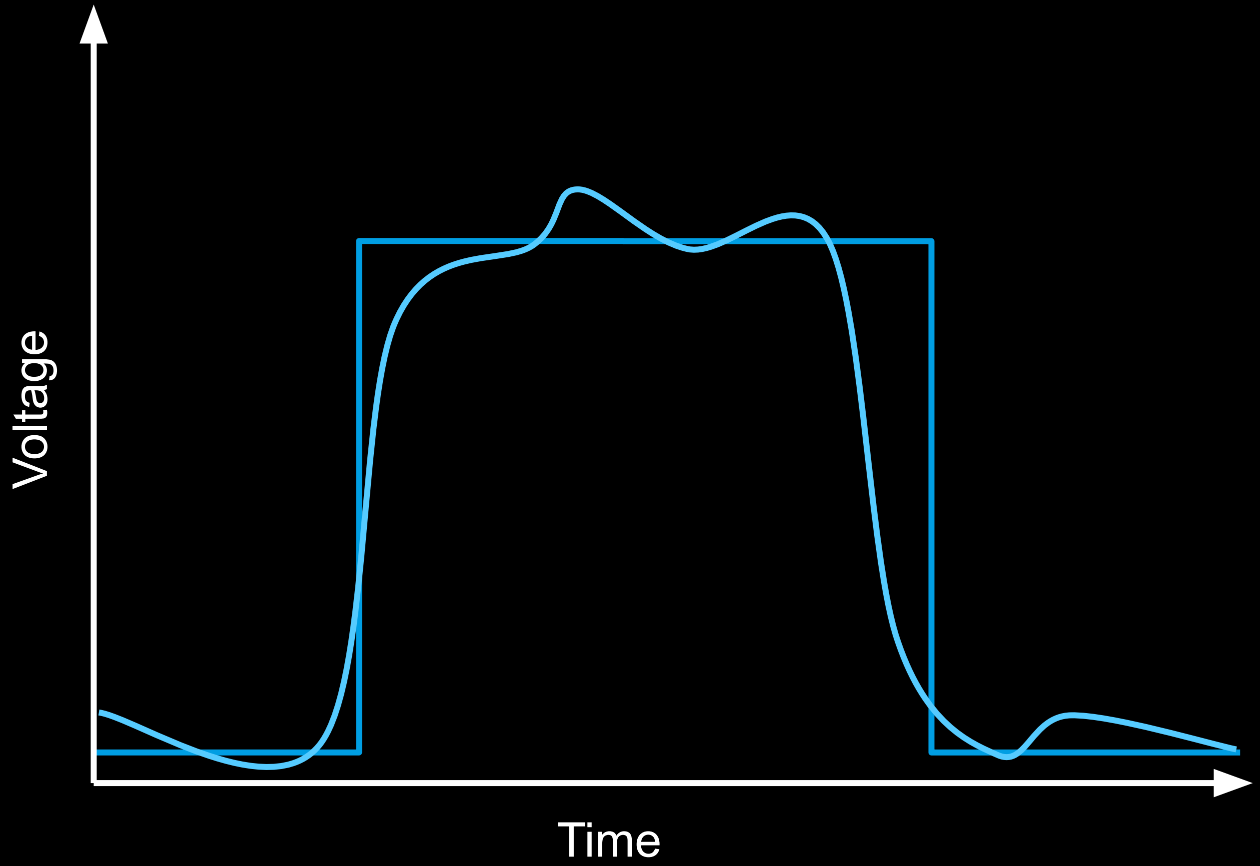
https://www.matheretter.de/wiki/zehnerpotenzen-namen how many bits are on a DVD with 4.7 GB capacity?

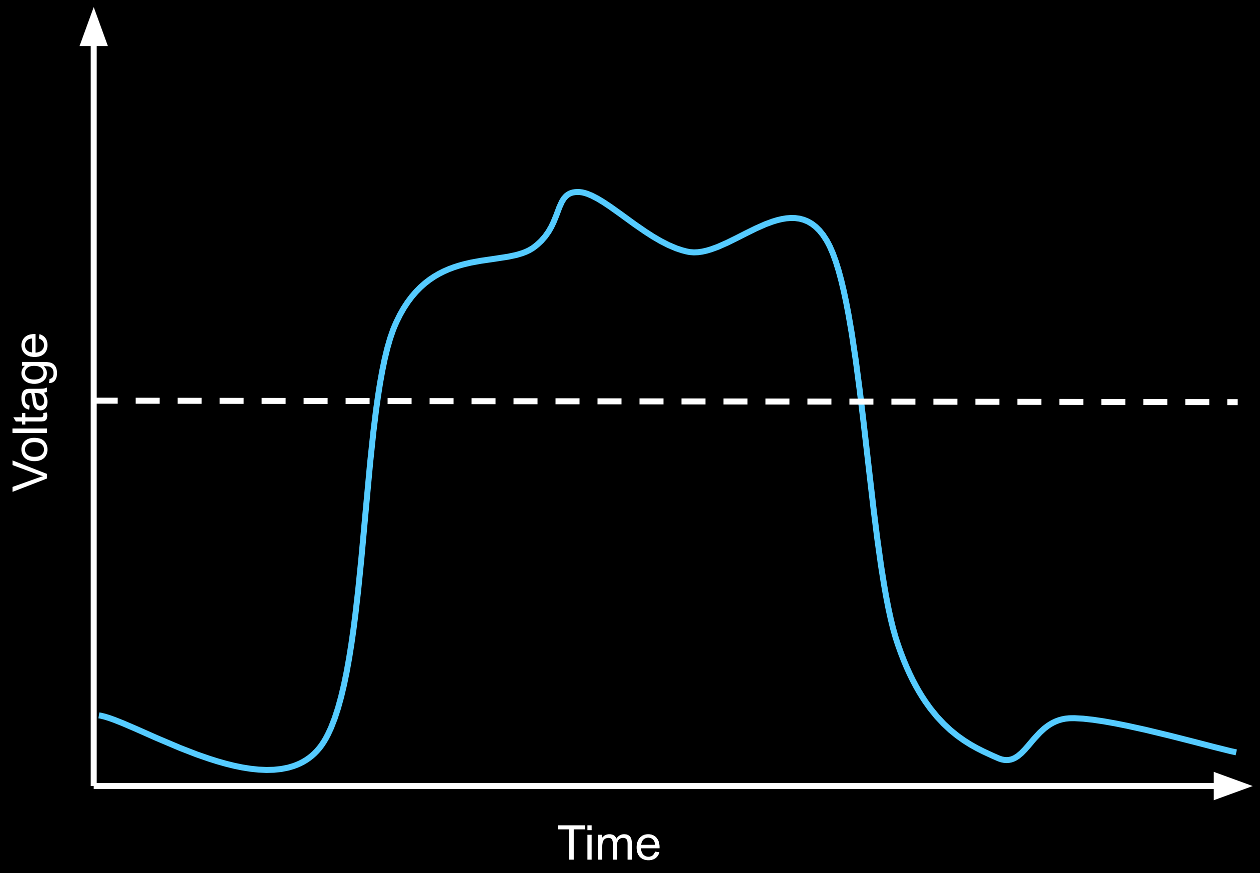
are we stuck with binary?

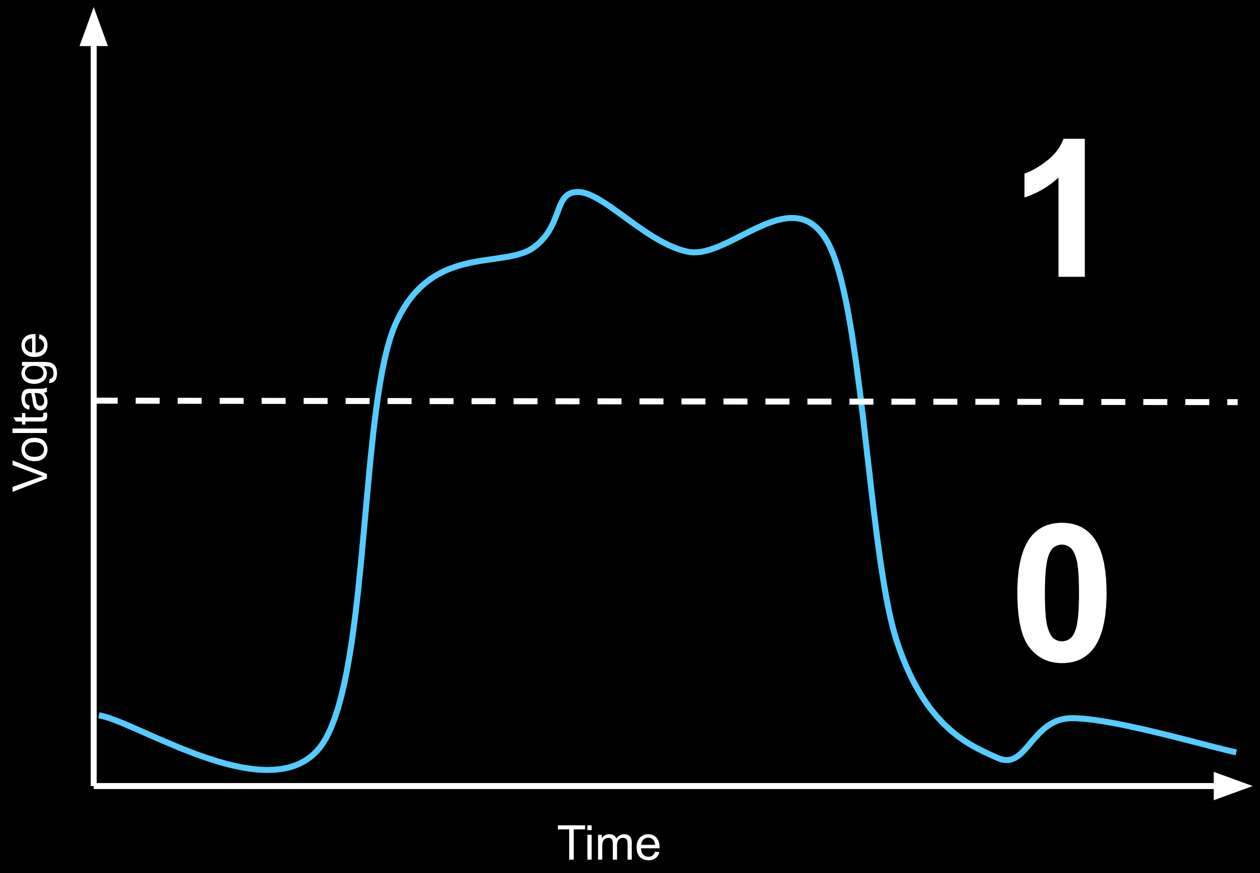

Voltage
Time


Voltage
Time

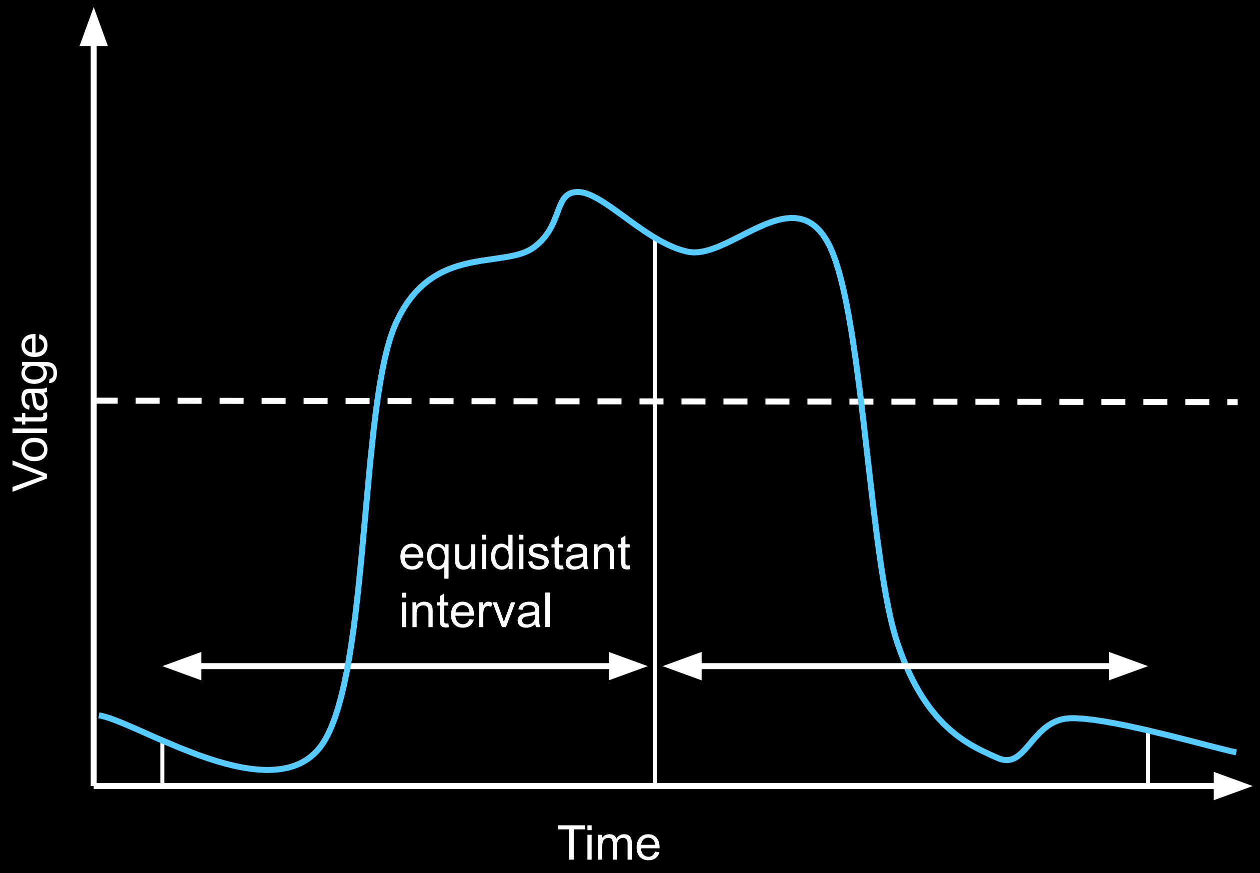

1
0
0
Voltage
Time

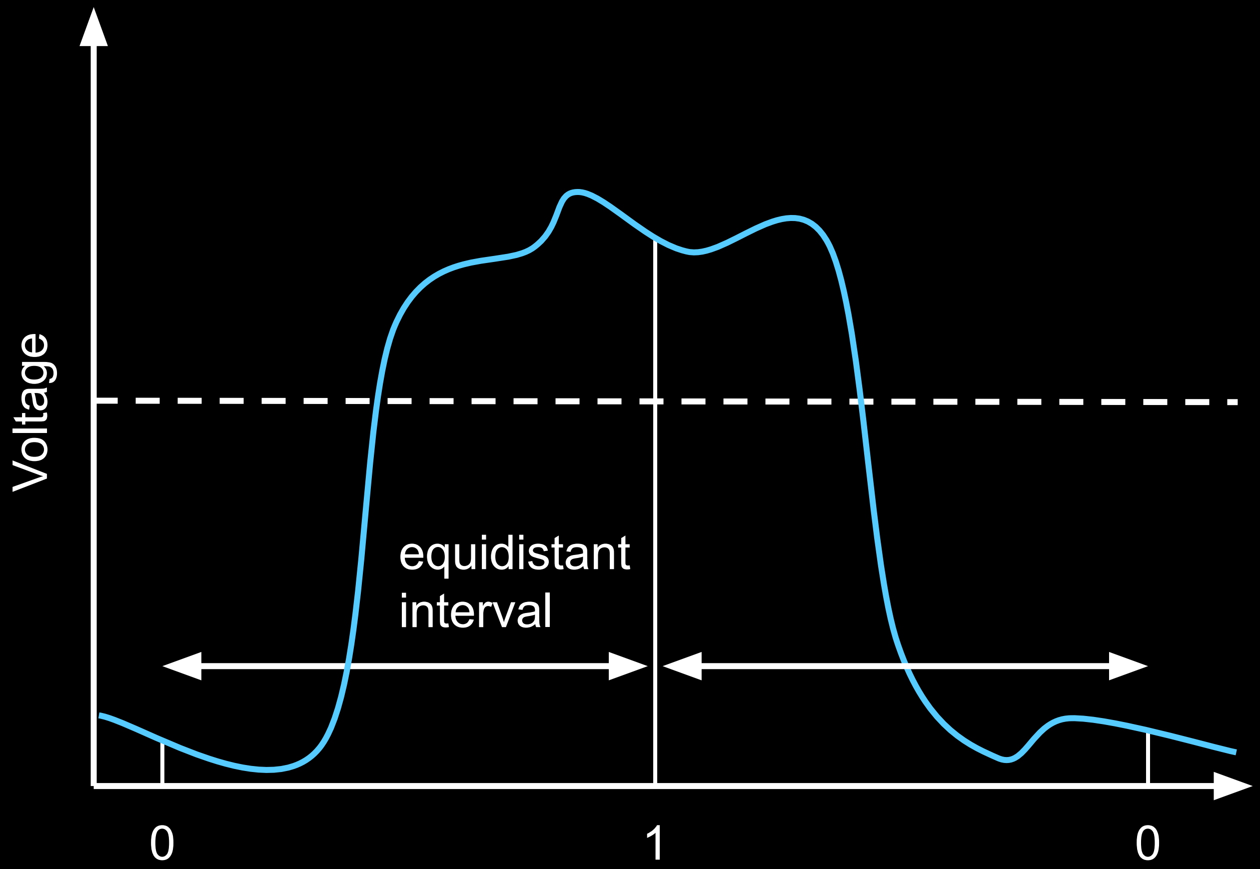

Voltage
Time

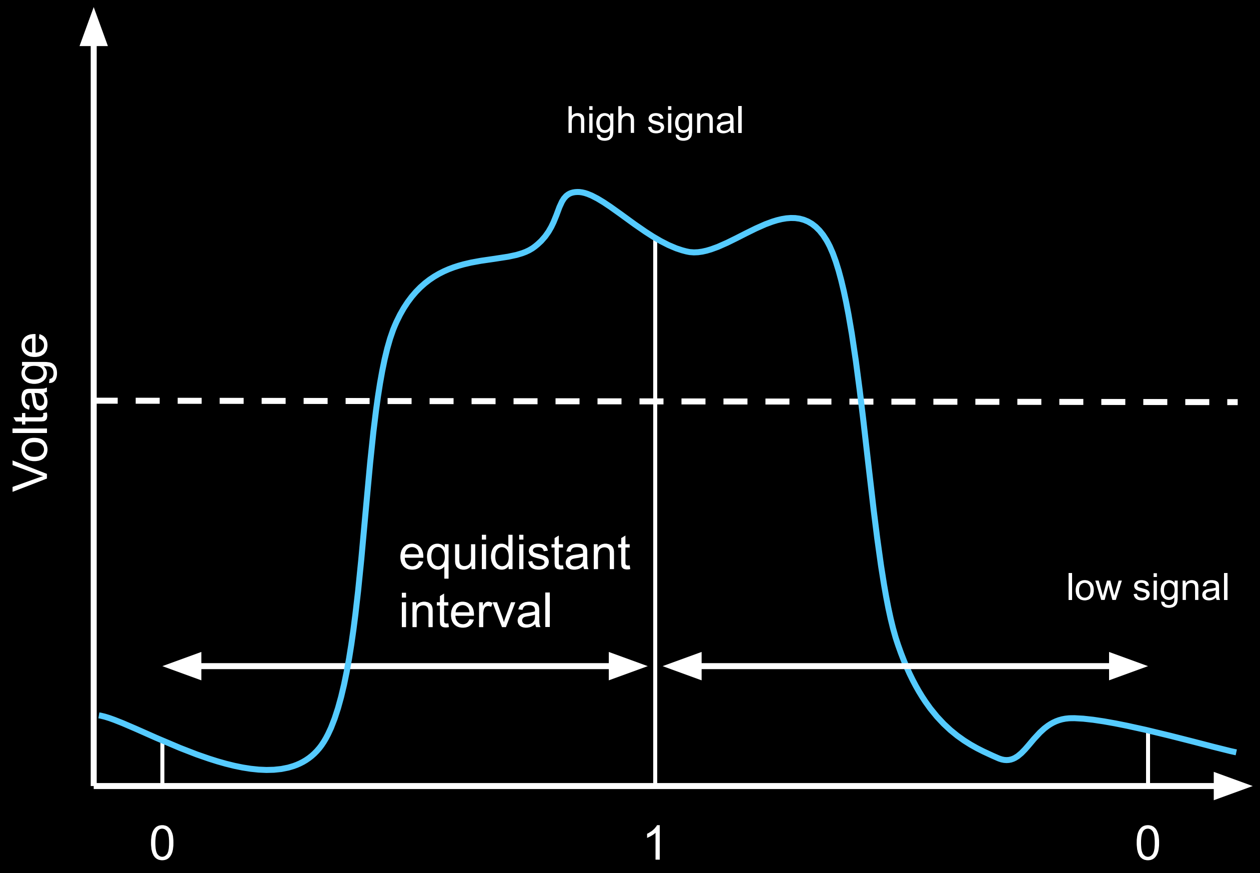

Voltage
Time



1
0
Voltage
Time

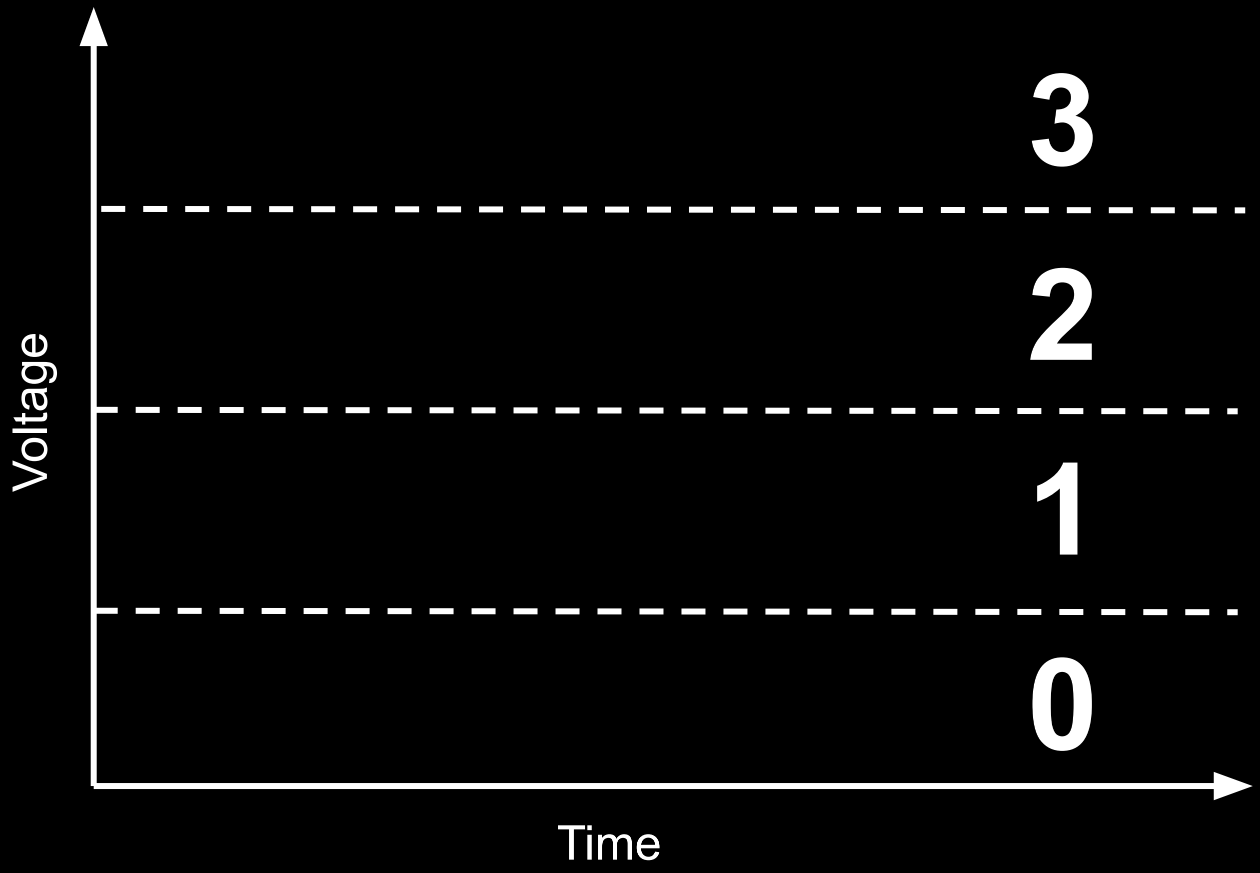

Voltage
equidistant
interval
Time

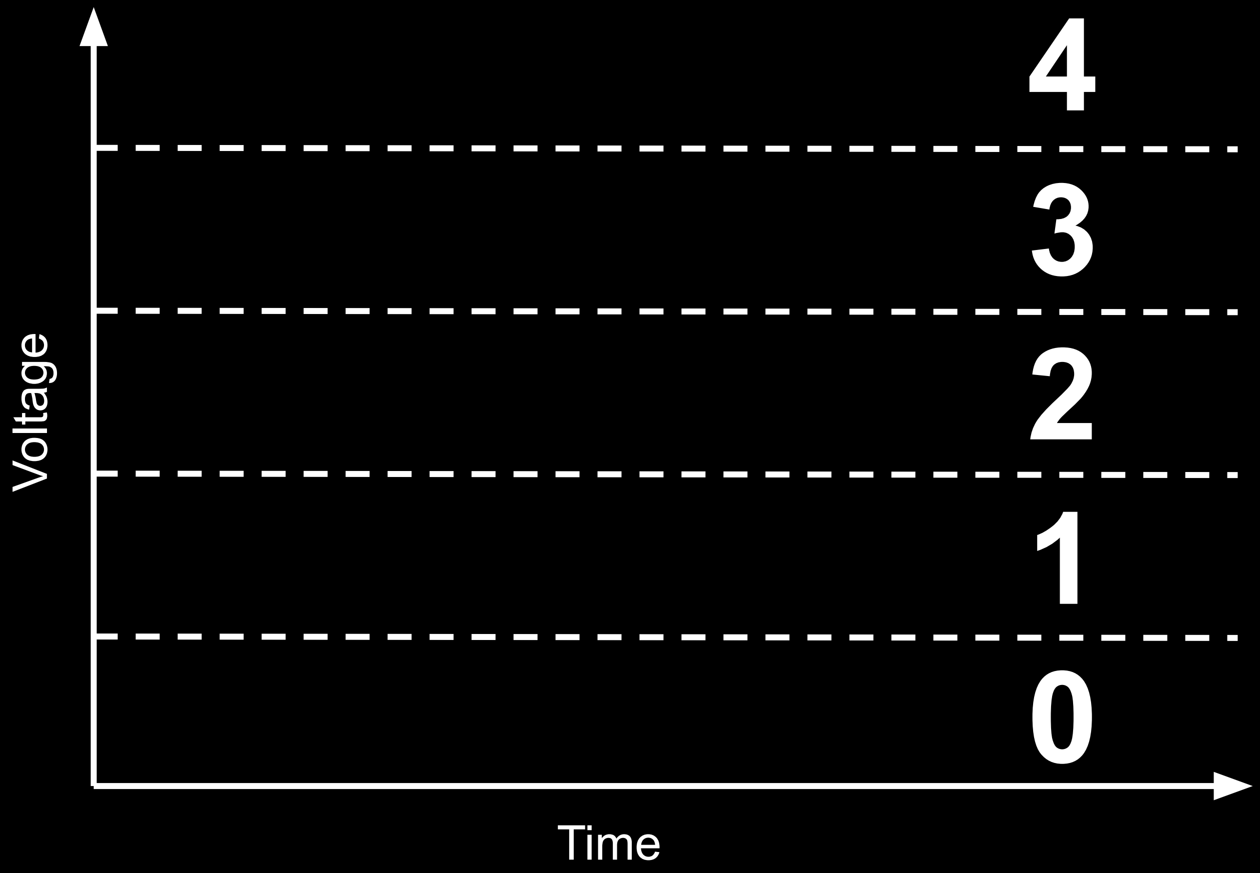

Voltage
equidistant
interval
0
1
0

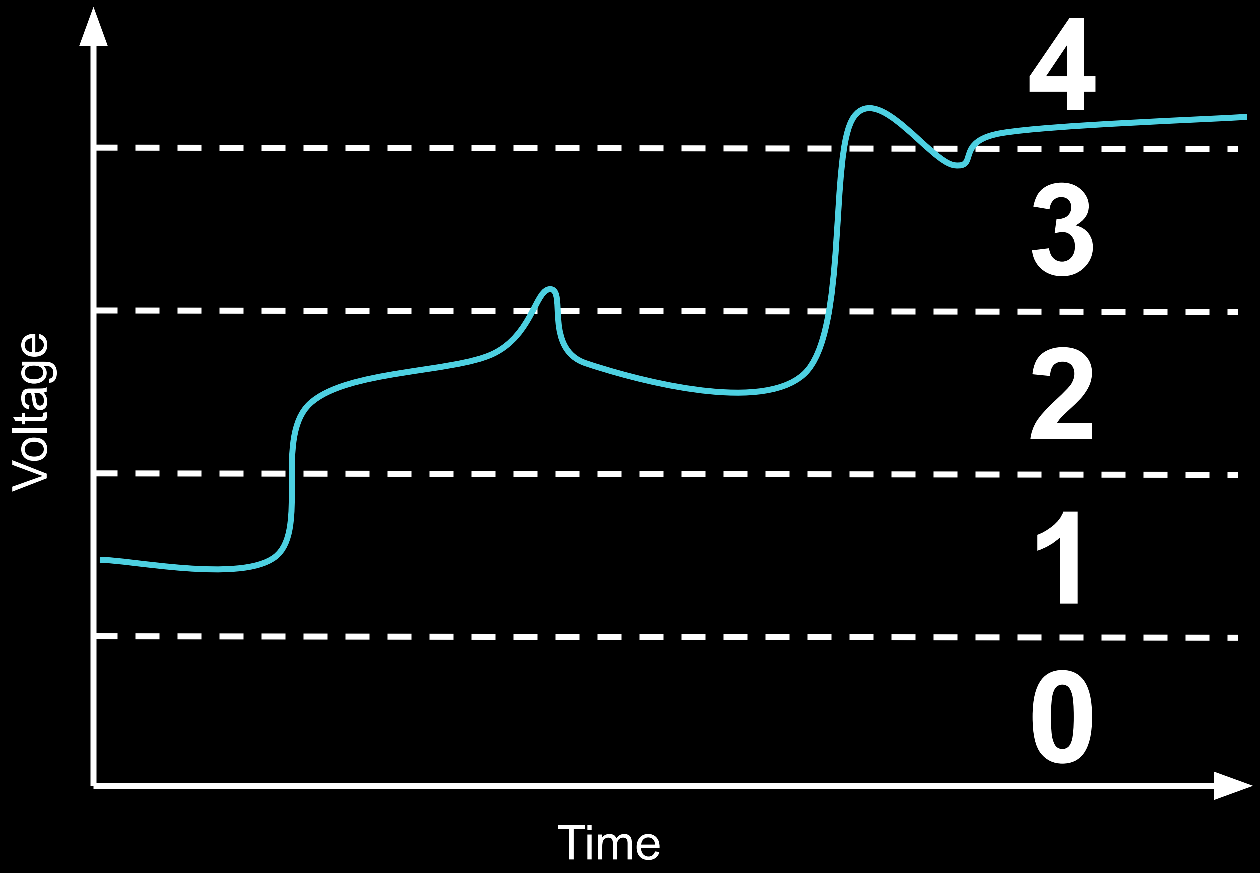

high signal
Voltage
equidistant
interval
low signal
0
1
0

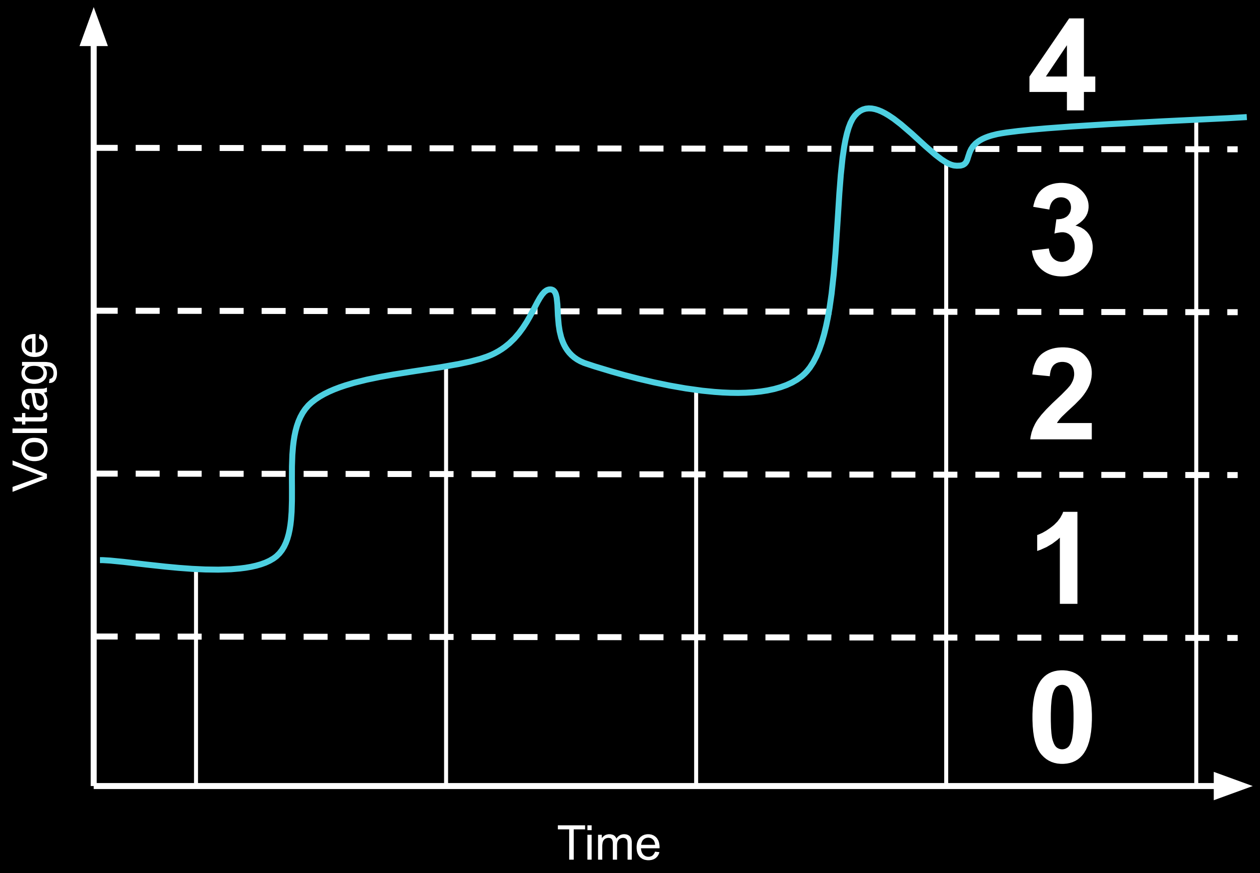
what about $R > 2$?

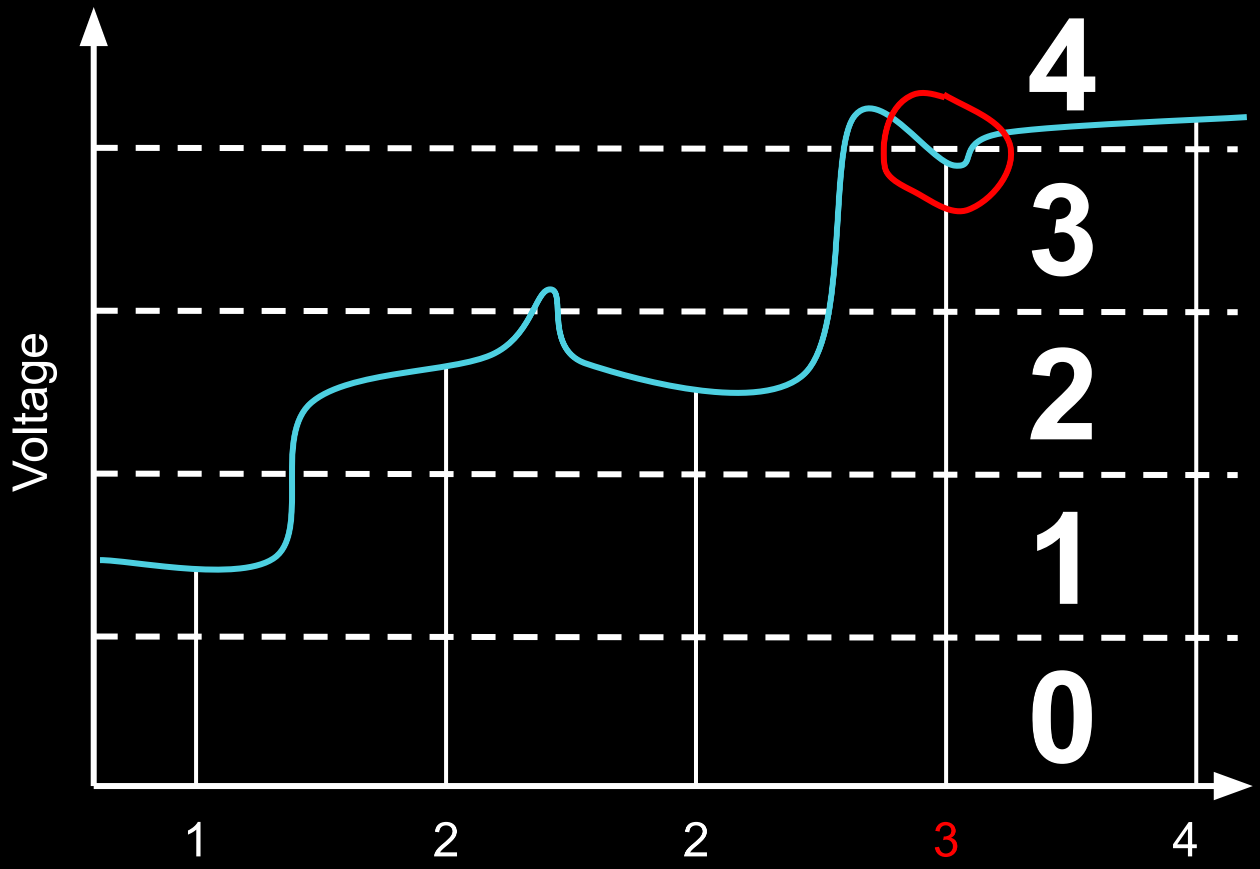

2
1
0
Voltage
Time


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

a higher base means less hardware

a higher base means less hardware

but more complex devices

a higher base means less hardware

but more complex devices

and more errors