Adventures in Computing

Christopher League

LONG ISLAND
UNIVERSITY

What is computing?

- It is about **information** and **computation**

- **Computational processes** occur everywhere, and existed long before computers were invented

Example: your brain

- Humans did **information processing** long before they had electronic & digital gadgets

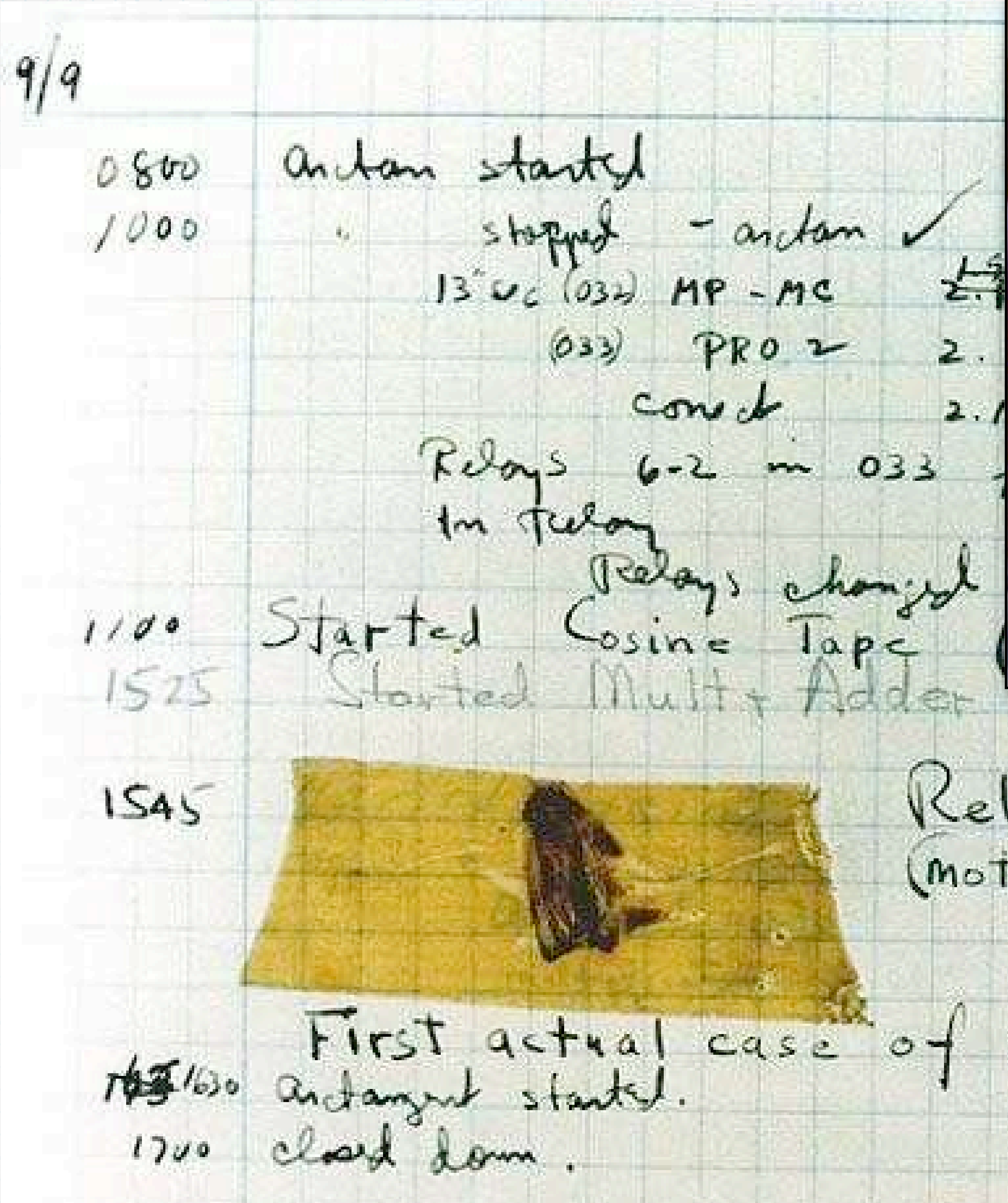
Some things computing professionals think about:

- How to **represent** things in the computer
- How to **instruct** the computer to do things
- How do computers (& people) **communicate?**
- Efficiency, reliability, security, usability, ...

Correctness

Grace Hopper

Correctness



9/9

0800 anctan started

1000 " stopped - anctan ✓

13" UC (032) MP - MC 2.

(033) PRO 2 2.

conrect 2.

Relays 6-2 in 033

in relay

Relays changed

1100 Started Cosine Tape (

1525 Started Mult+ Adder Test.

1545 Relay #70 Panel F (moth) in relay.

First actual case of bug being found.

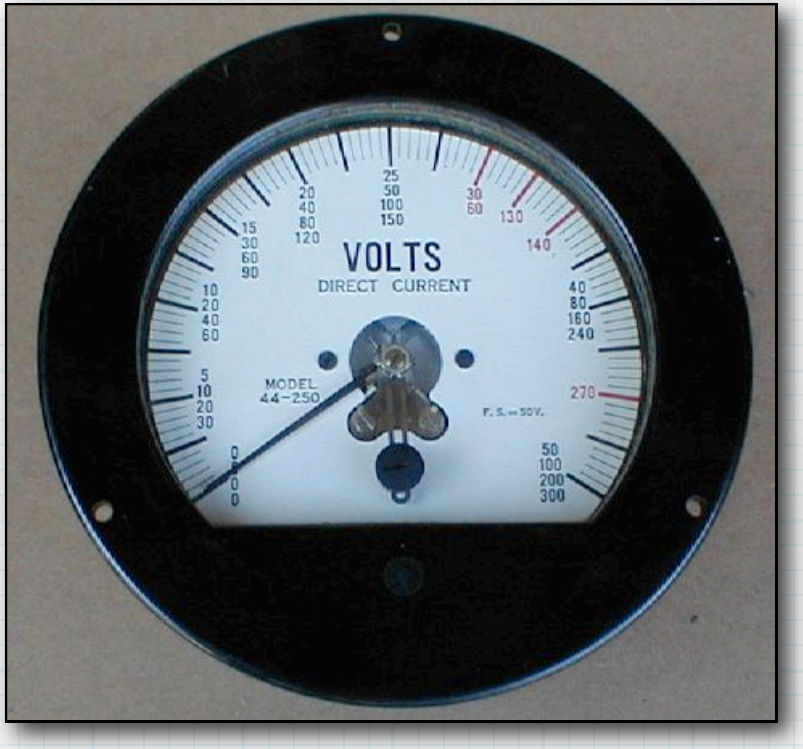
1630 antangent started.

1700 closed down.

Grace Hopper

Representing things

- Ultimately, computers can only hold electrical current in specialized circuits: voltage may be **high** or **low.**

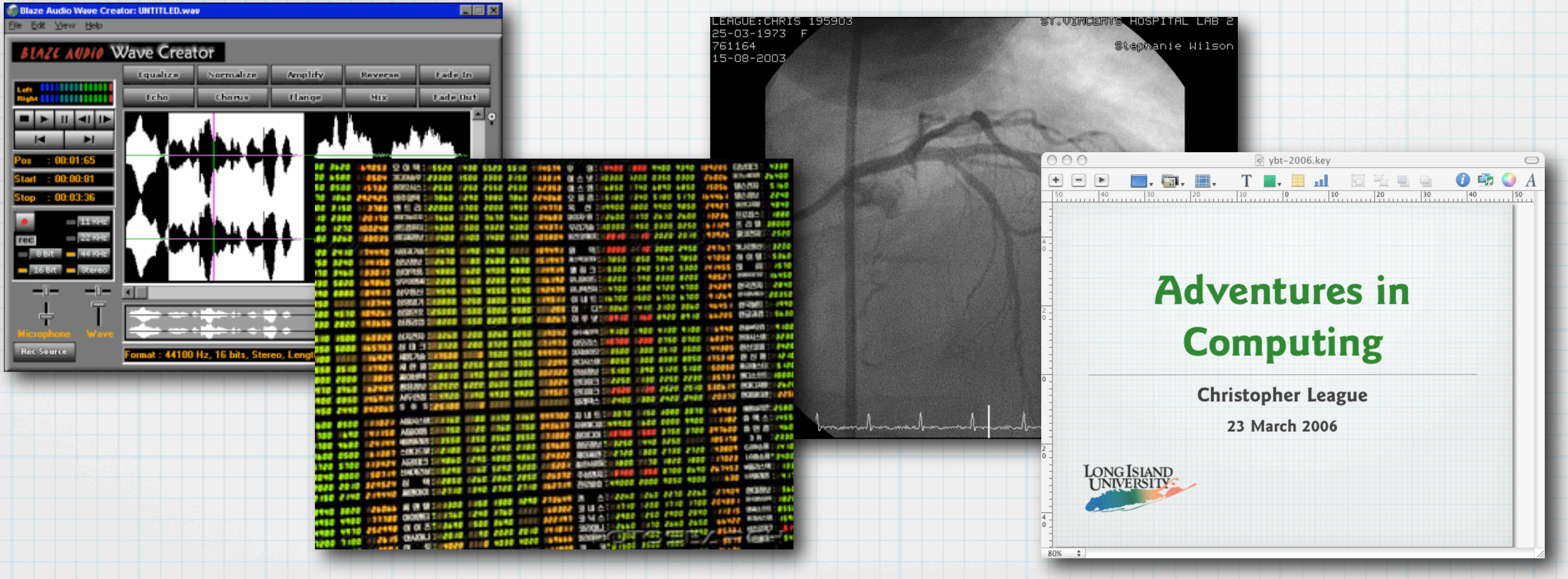


Representing things

- Similarly, storage devices have magnetic surfaces that can be **polarized:** + or –

Representing things

- With just **two** states, we can represent anything:
- Numbers, text, audio/video, financial or medical data, presentations, …

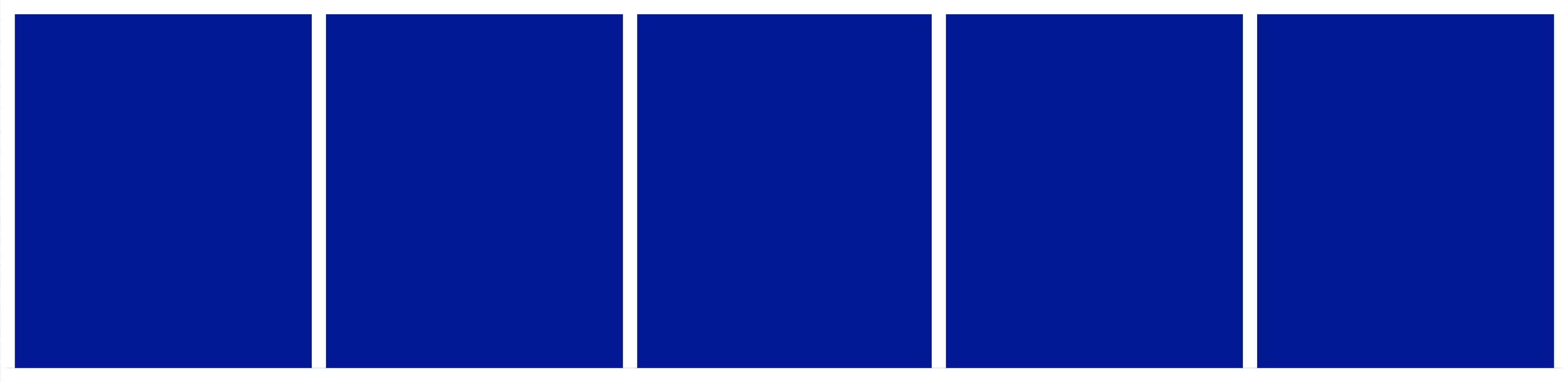


Adventures in Computing

Christopher League

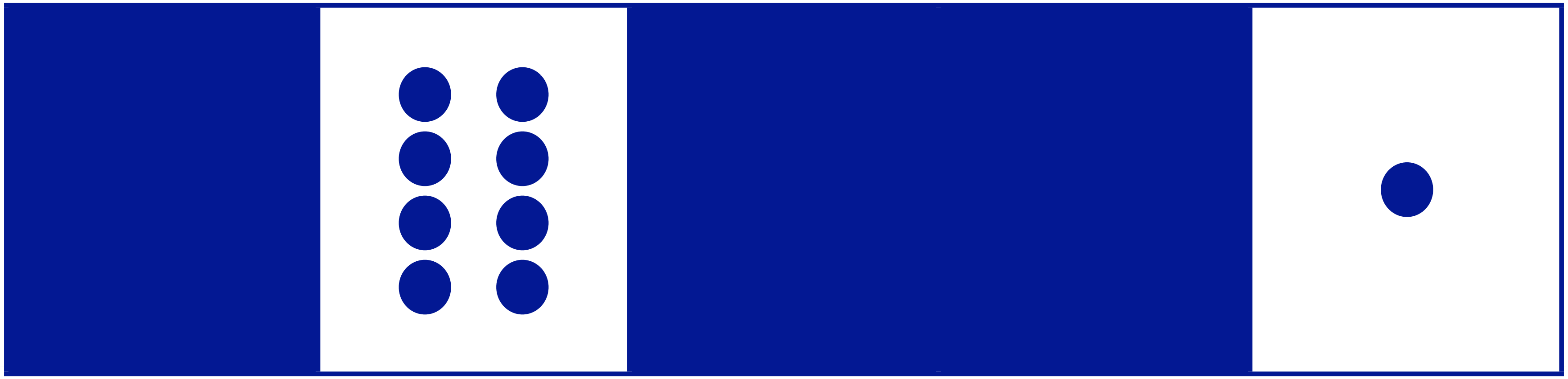
23 March 2006

Representing numbers



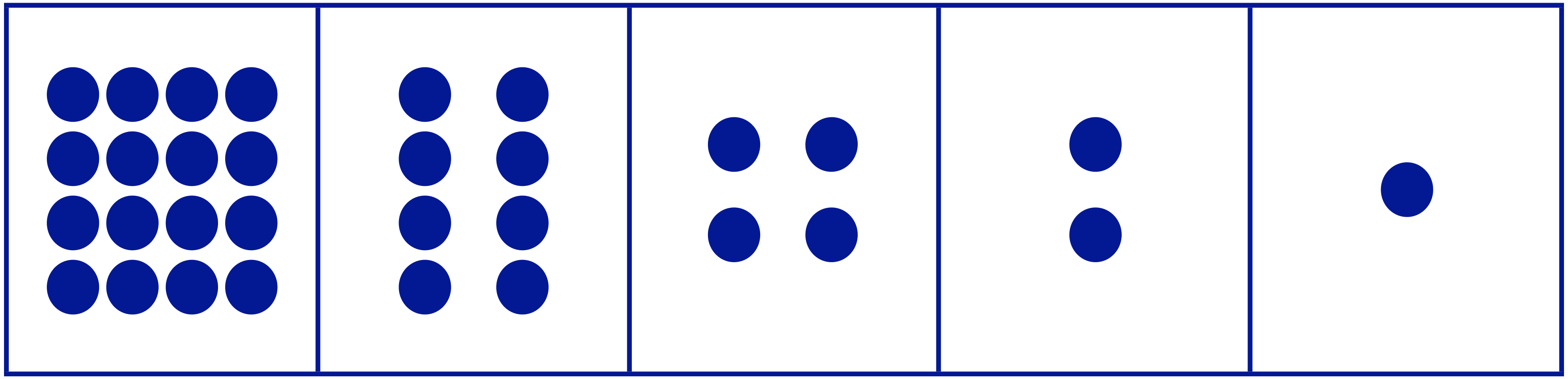
- Each of these cards can be **on** or **off** (face up or face down)
- When **on**, the card represents the number of dots on its face.

Representing numbers



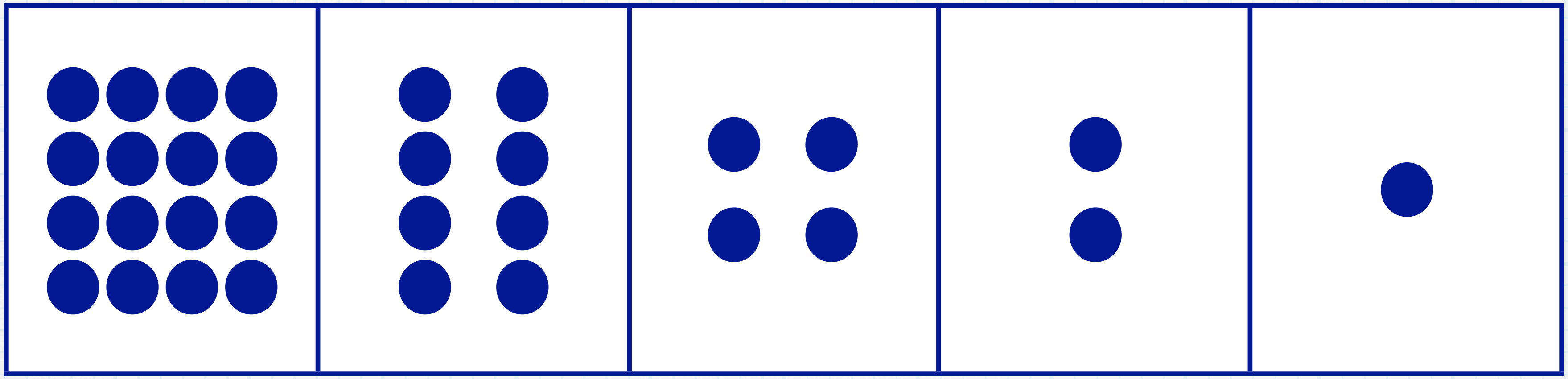
- Each of these cards can be **on** or **off** (face up or face down)
- When **on**, the card represents the number of dots on its face.

Representing numbers



- Each of these cards can be **on** or **off** (face up or face down)
- When **on**, the card represents the number of dots on its face.

Representing numbers



16 + 8 + 4 + 2 + 1 =

31

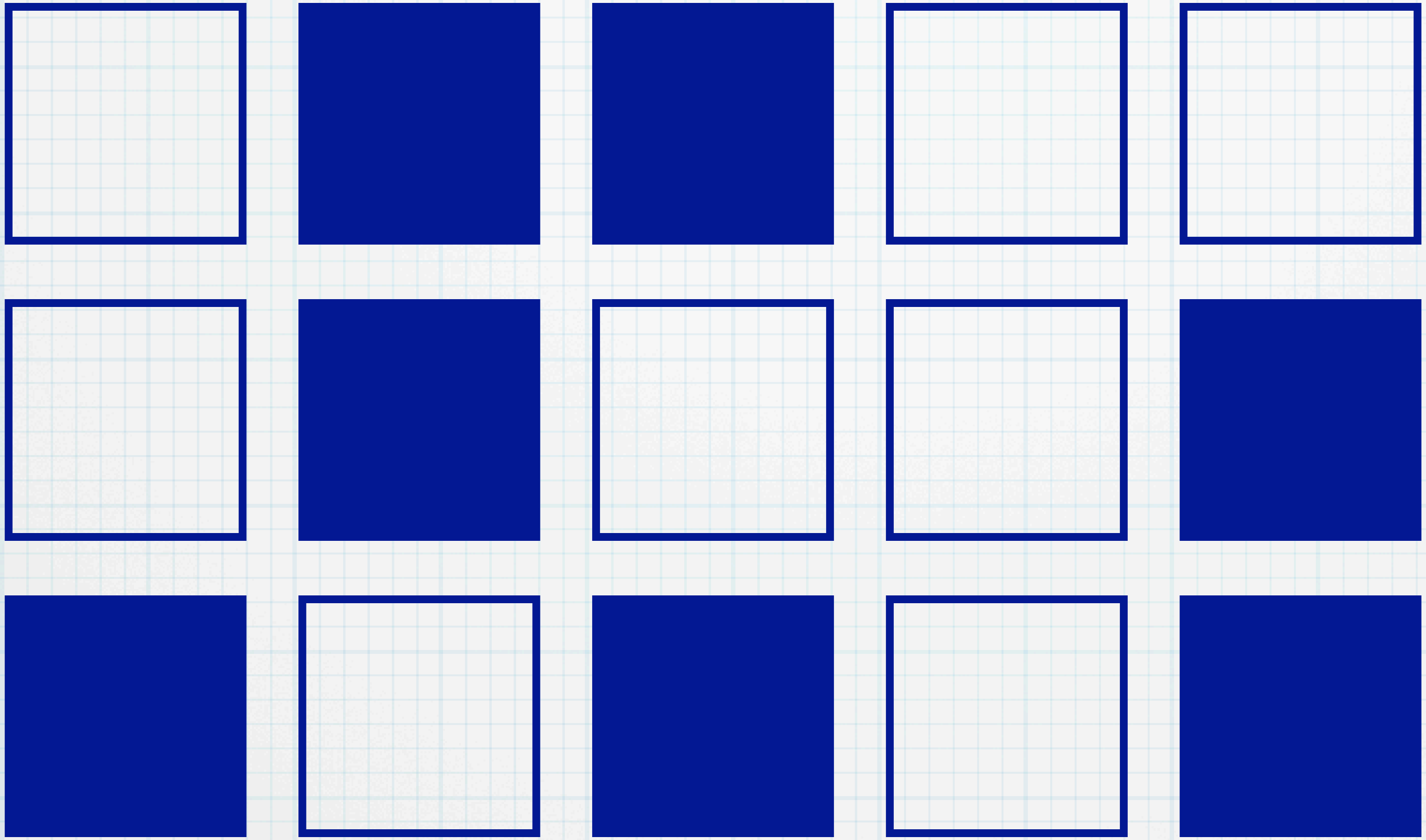
The largest number we can represent with 5 cards!

Representing text

■	■	■	■	■	0	‘A’
■	■	■	■	1	1	‘B’
■	■	■	2	■	2	‘C’
■	■	■	2	1	3	‘D’
■	■	4	■	■	4	‘E’
■	■	4	■	1	5	‘F’
■	■	4	2	■	6	‘G’

etc.

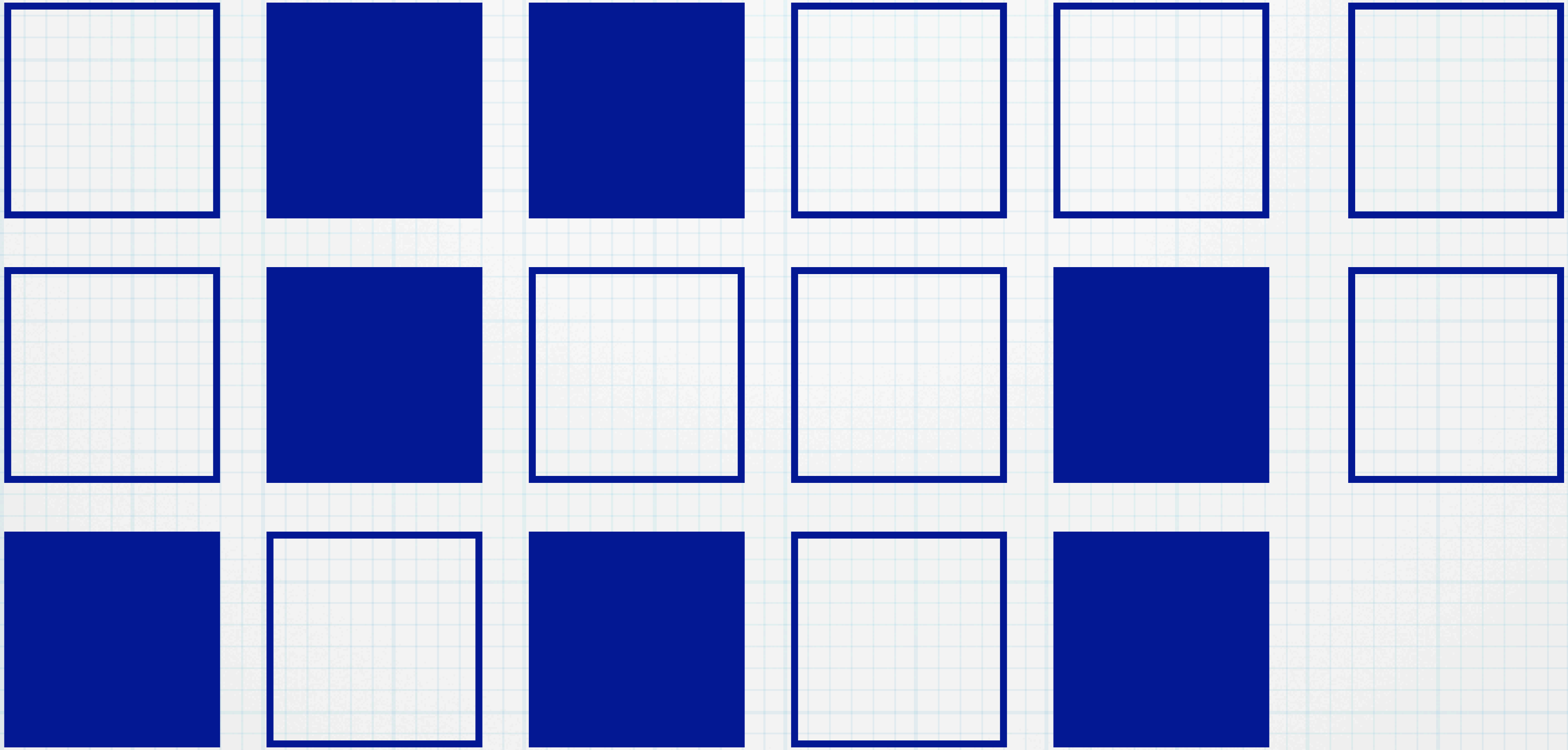
Error correction—parity



Error correction—parity

□	■	■	□	□	□
□	■	□	□	■	
■	□	■	□	■	

Error correction—parity



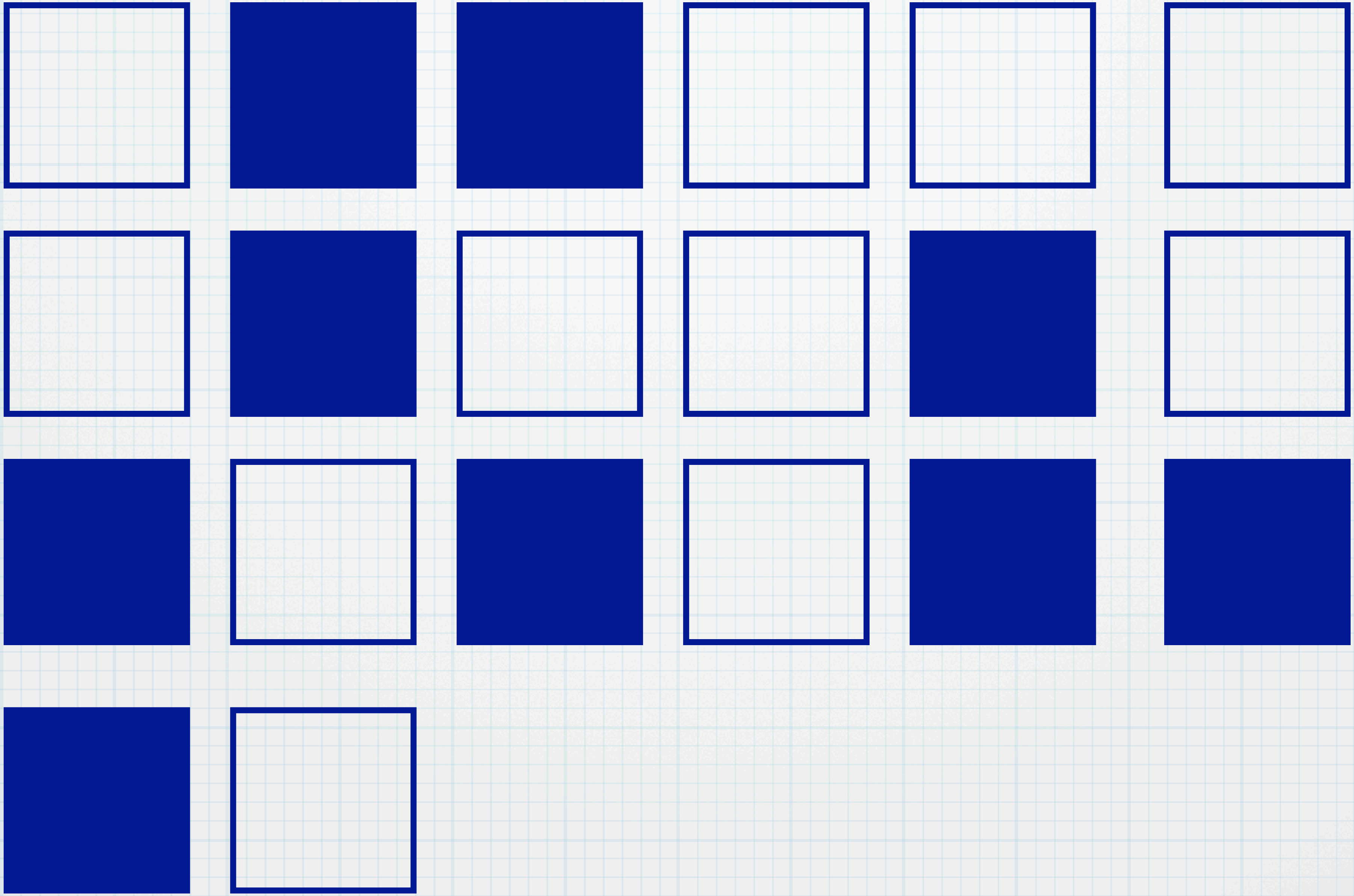
Error correction—parity

□	■	■	□	□	□
□	■	□	□	■	□
■	□	■	□	■	■

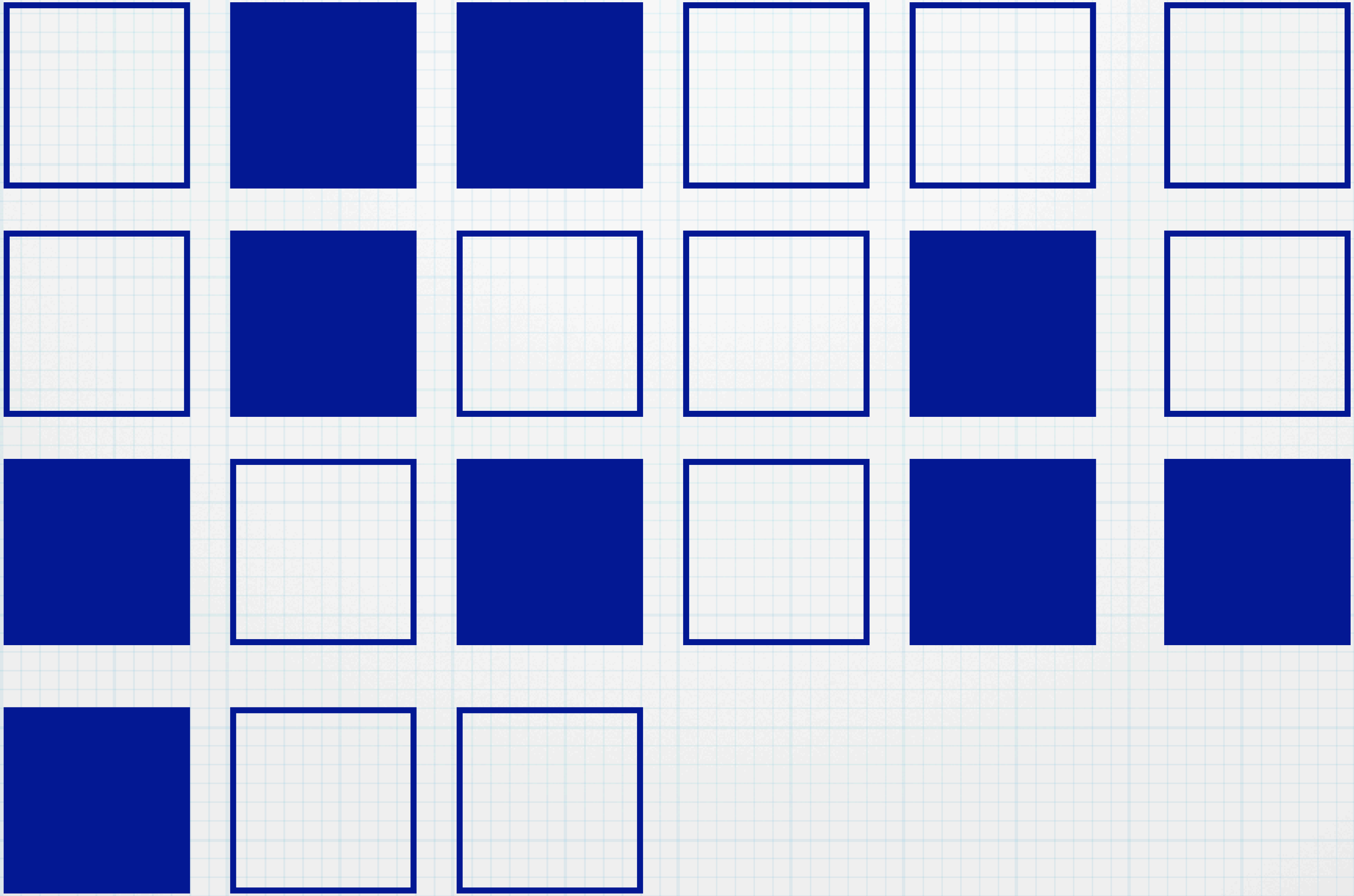
Error correction—parity

□	■	■	□	□	□
□	■	□	□	■	□
■	□	■	□	■	■
■					

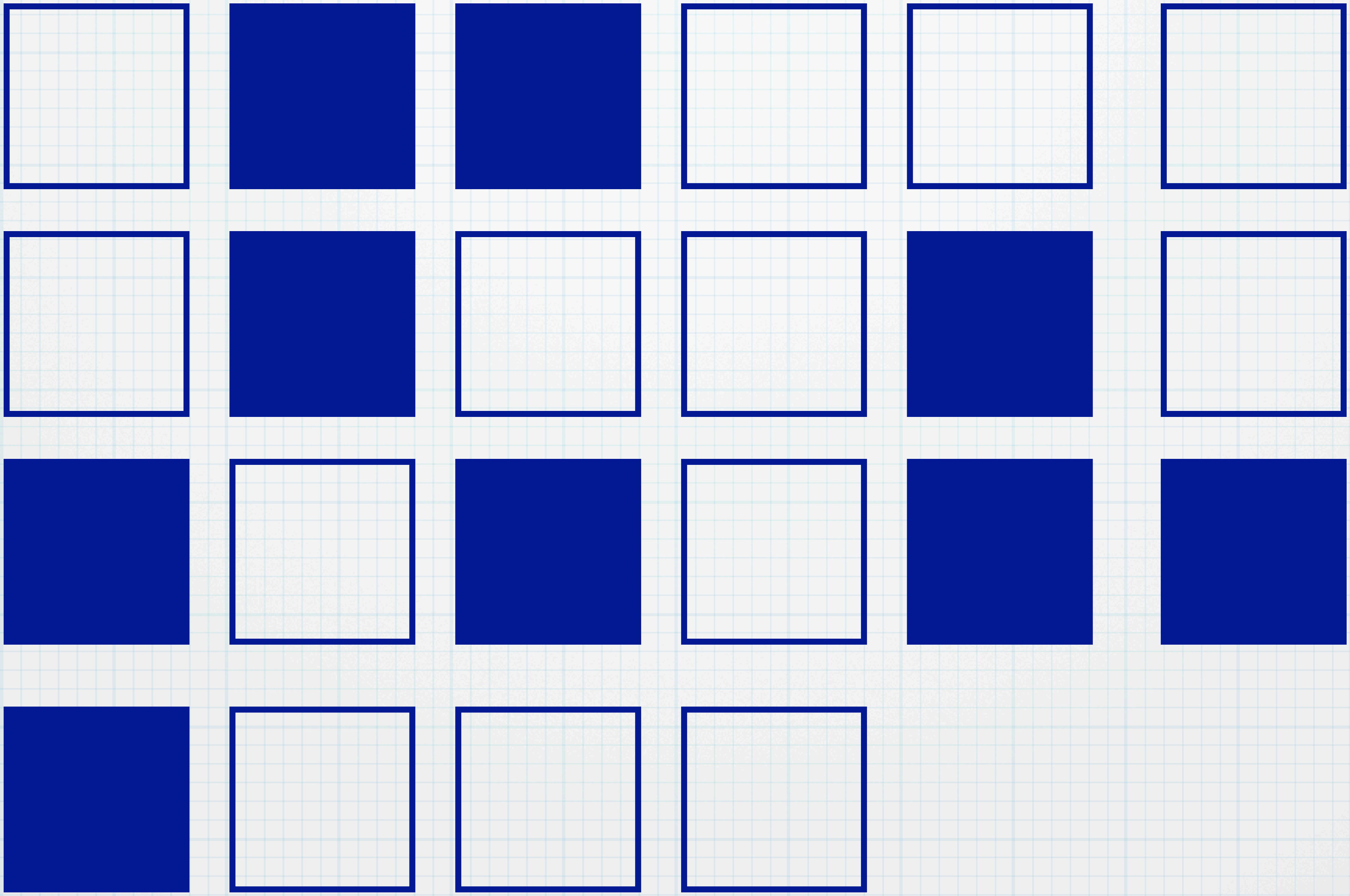
Error correction—parity



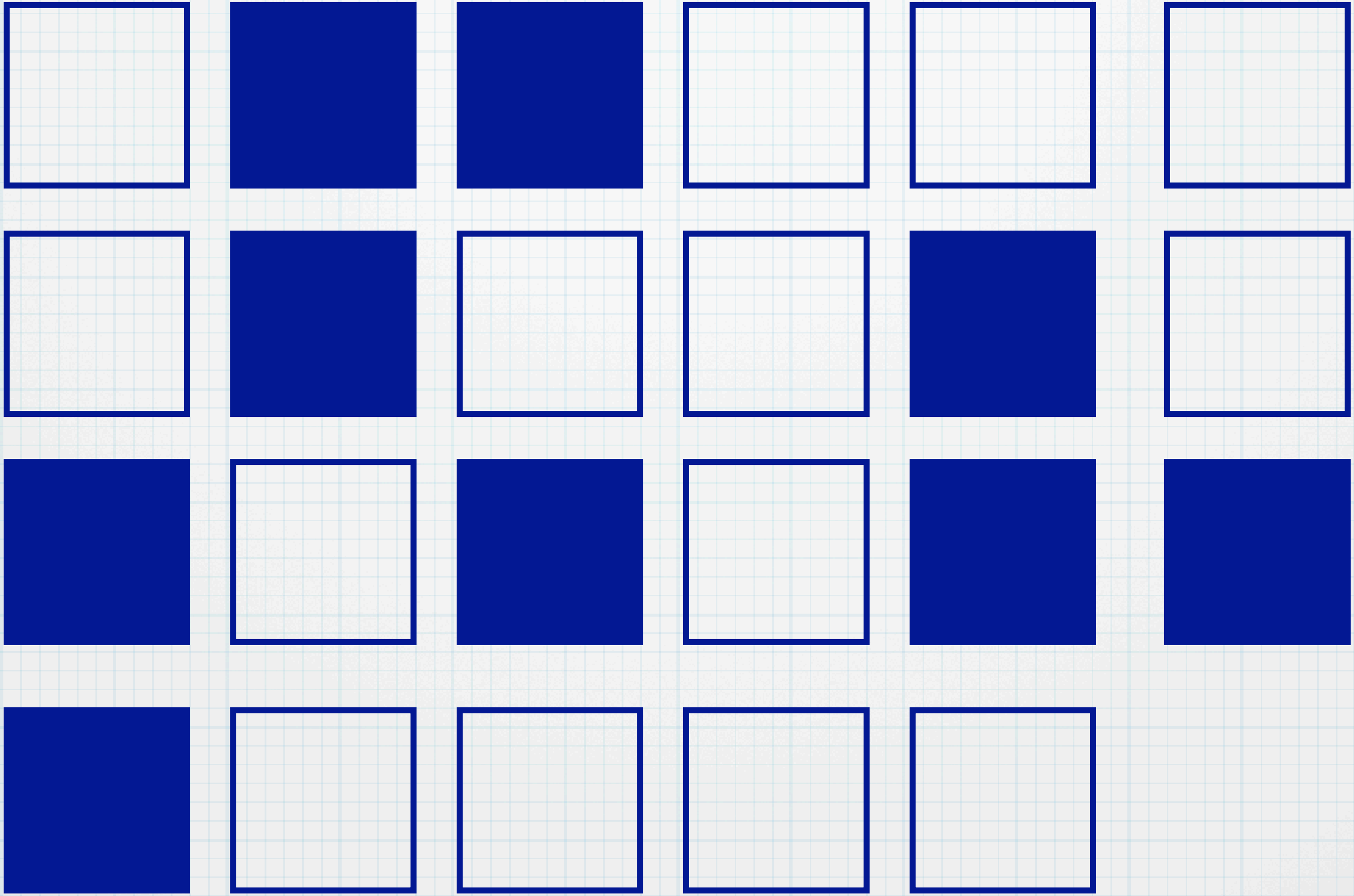
Error correction—parity



Error correction—parity



Error correction—parity



Error correction—parity

□	■	■	□	□	□
□	■	□	□	■	□
■	□	■	□	■	■
■	□	□	□	□	■

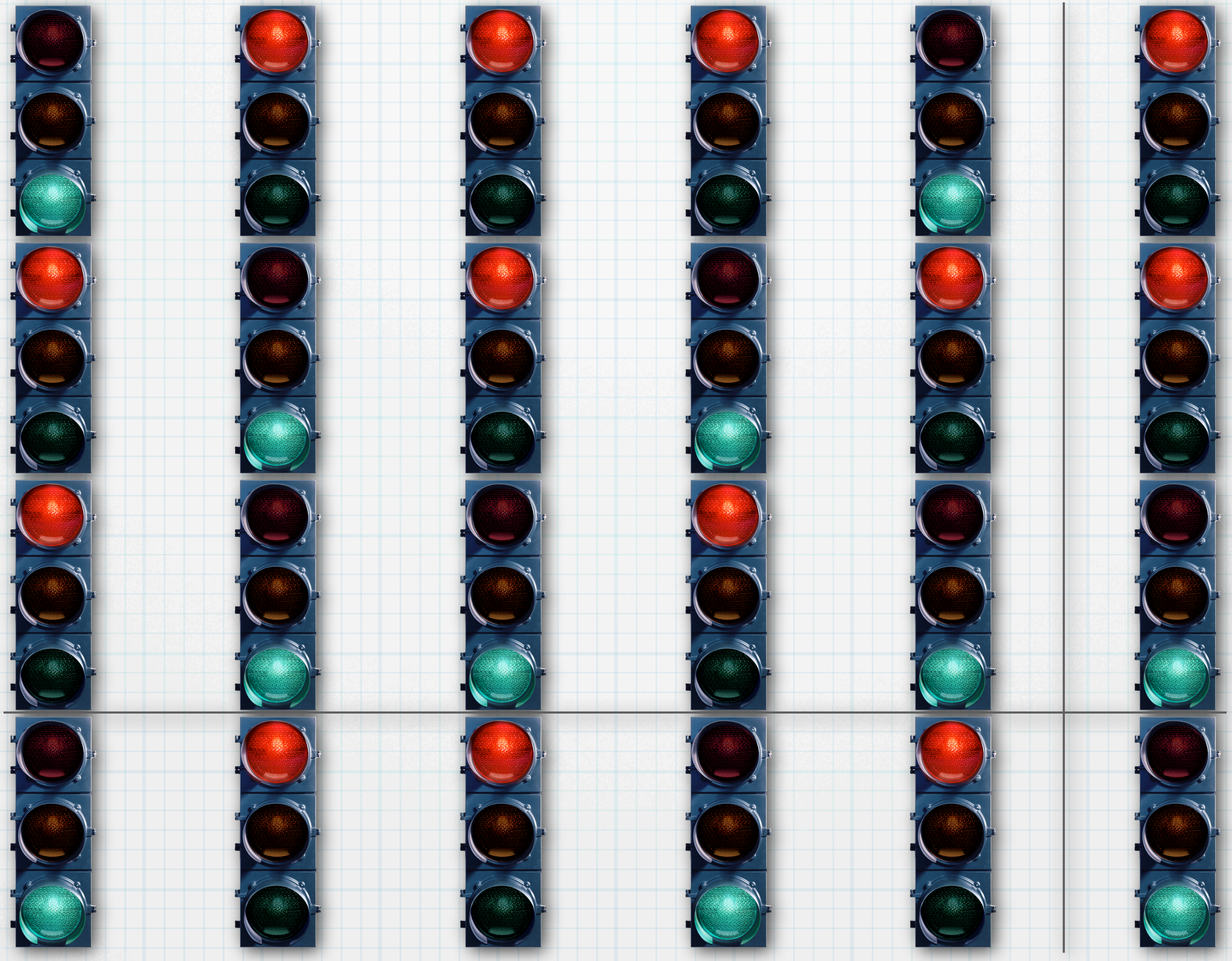
Find the error (#1)

■	□	□	□	□	■
■	□	■	□	■	■
□	■	□	□	□	□
□	■	■	□	□	□

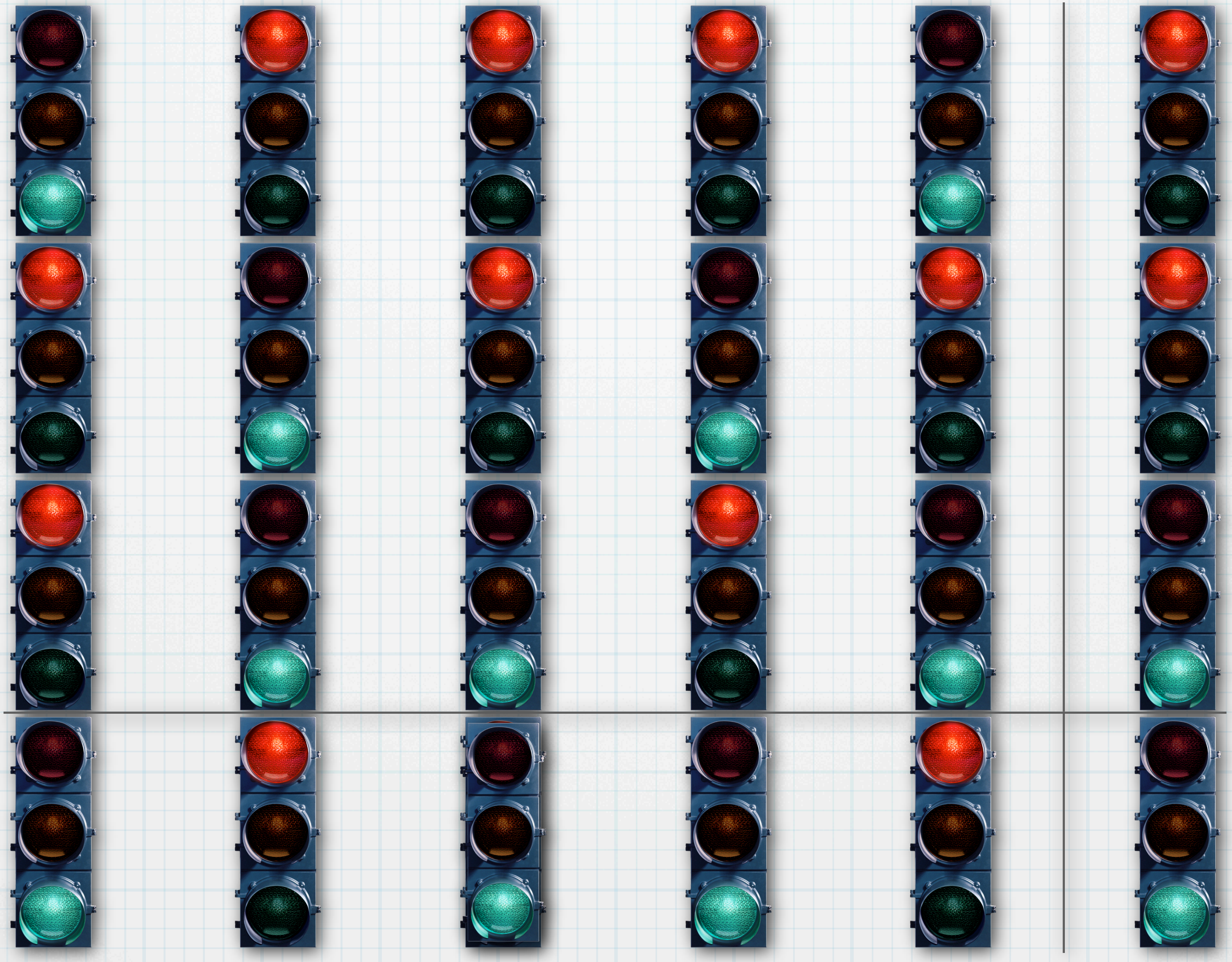
Find the error (#1)

■	□	□	□	□	■
■	□	■	□	■	■
□	■	□	□	■	□
□	■	■	□	□	□

Find the error (#2)



Find the error (#2)



Algorithms

- So far, we explored **representation**, **communication**, and **reliability**.
- Now, consider **instruction** and **efficiency**.

Searching for a name in the phone book

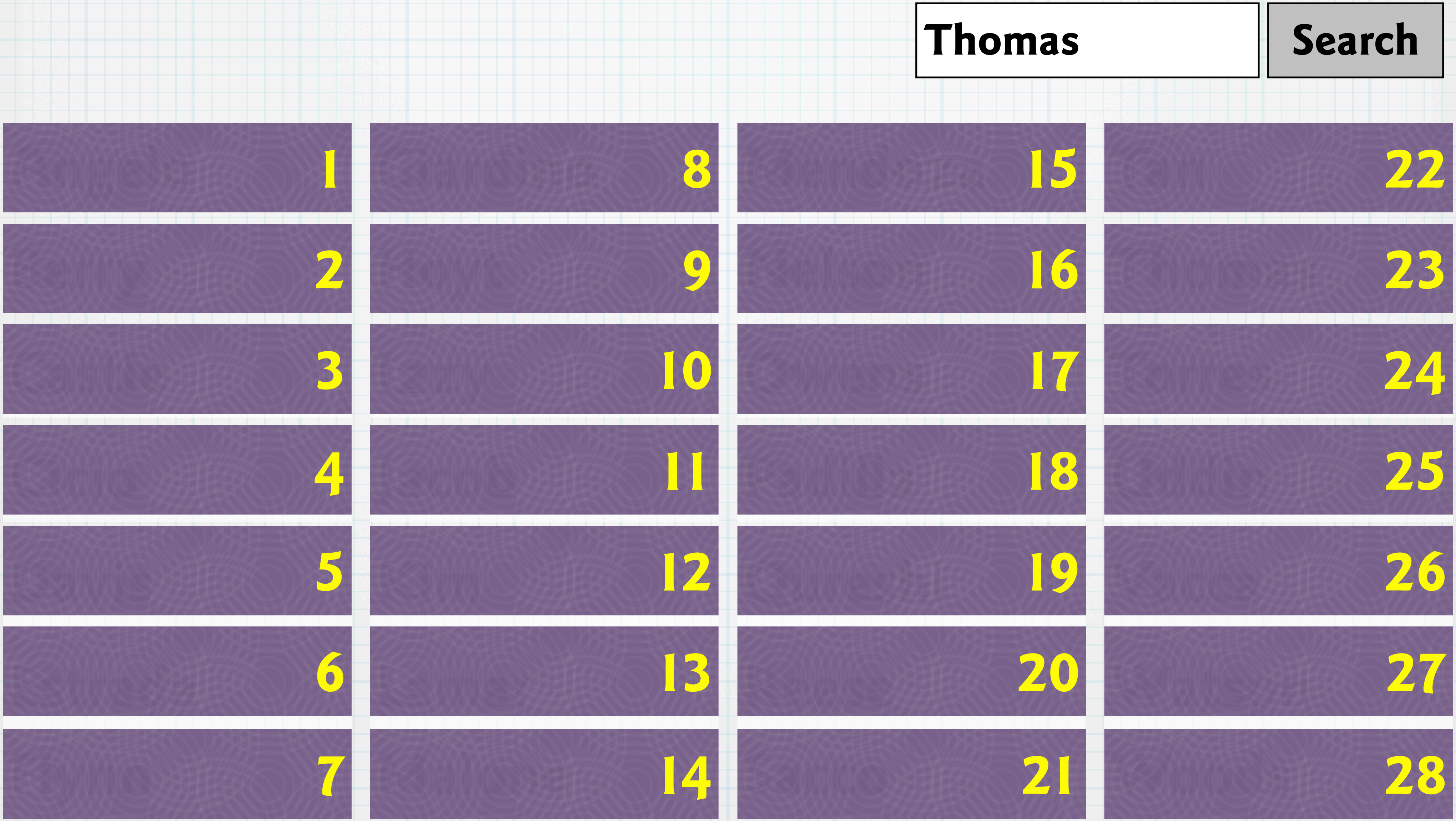
Searching for a name in the phone book

- How long does it take?

Searching for a name in the phone book

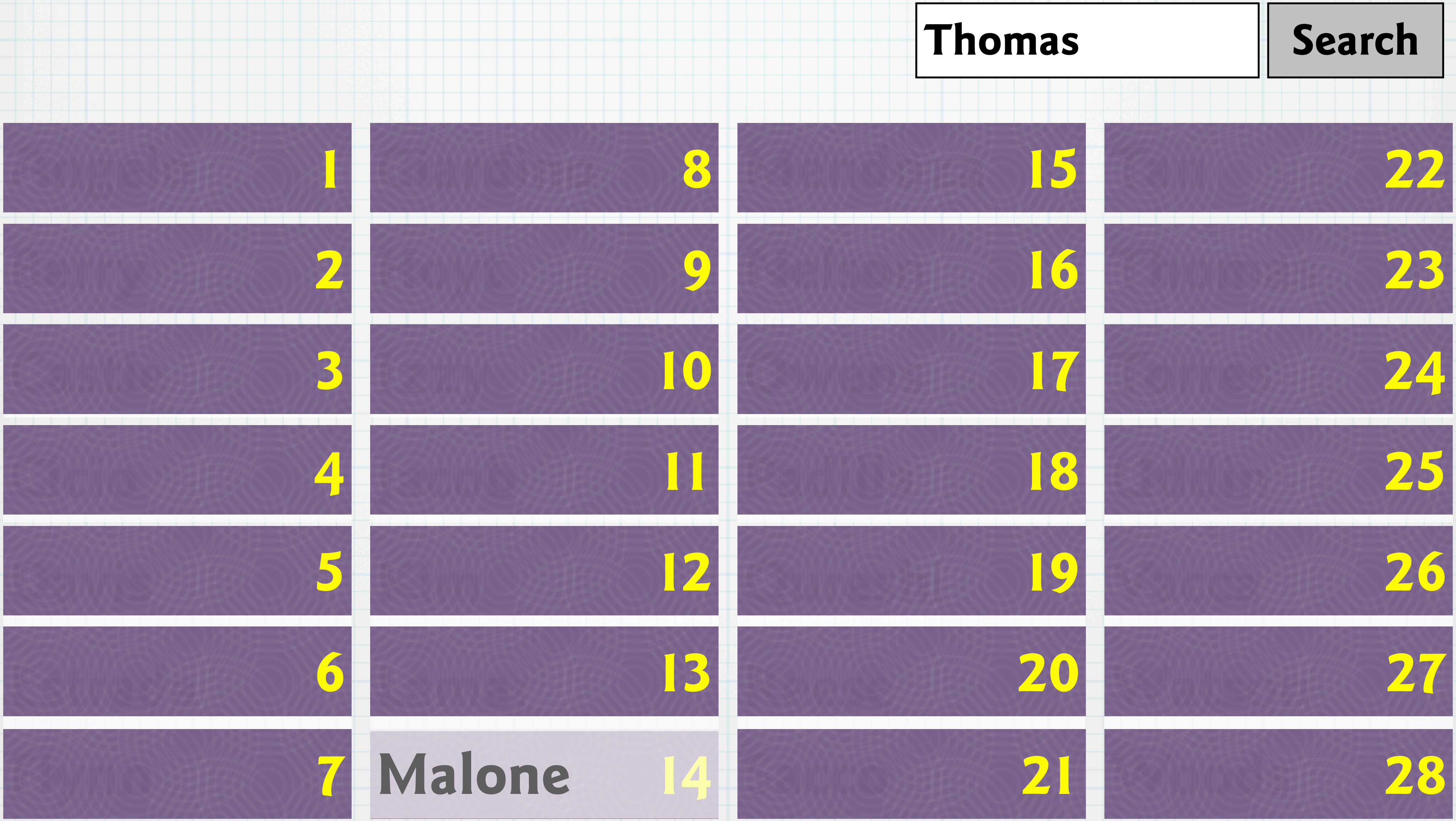
- How long does it take?
  - Depends on the size of the book!

Searching for a name



1	8	15	22
2	9	16	23
3	10	17	24
4	11	18	25
5	12	19	26
6	13	20	27
7	14	21	28

Searching for a name



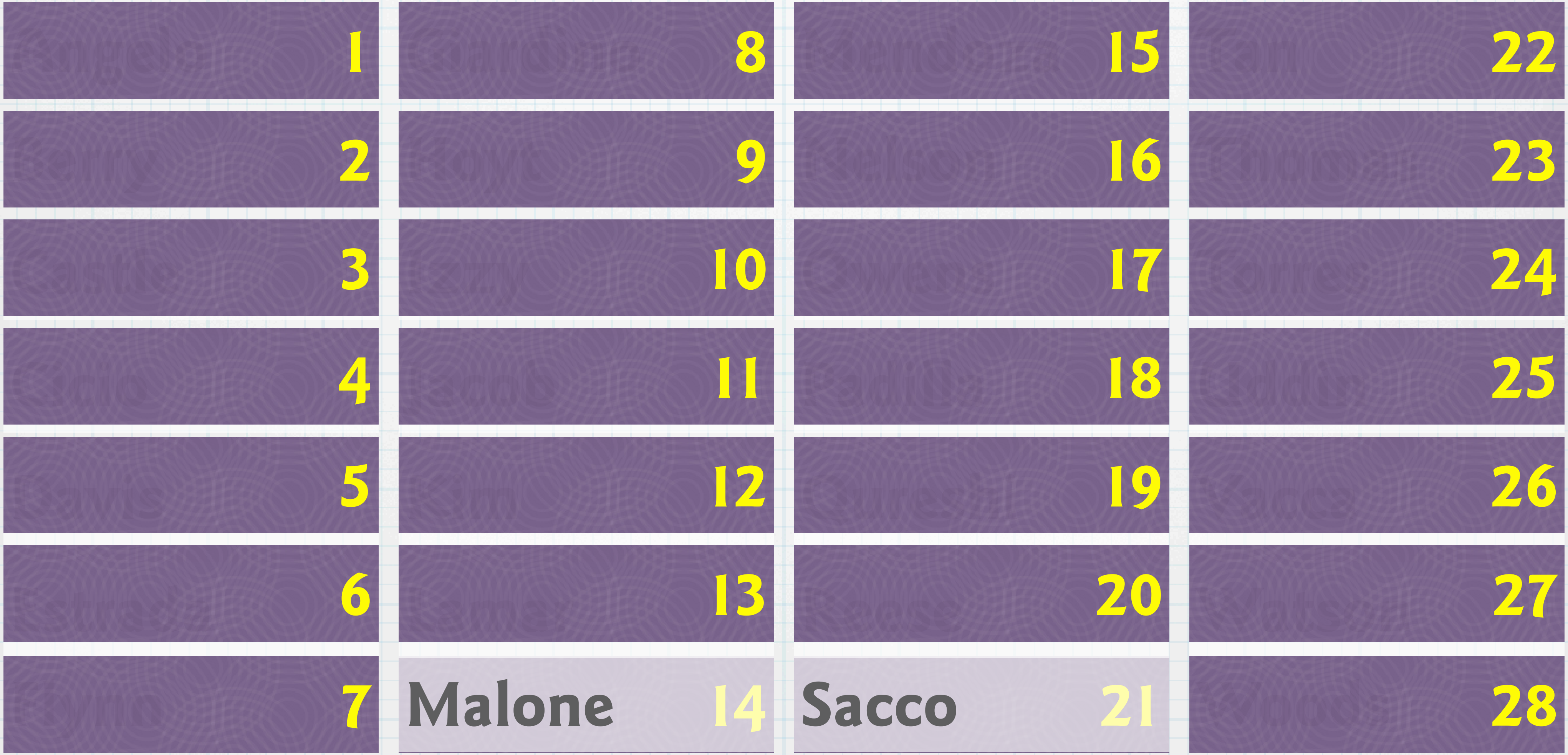
Thomas

1	8	15	22
2	9	16	23
3	10	17	24
4	11	18	25
5	12	19	26
6	13	20	27
7	Malone 14	21	28

Searching for a name

Thomas

Search



1	8	15	22
2	9	16	23
3	10	17	24
4	11	18	25
5	12	19	26
6	13	20	27
7	Malone 14	Sacco 21	28

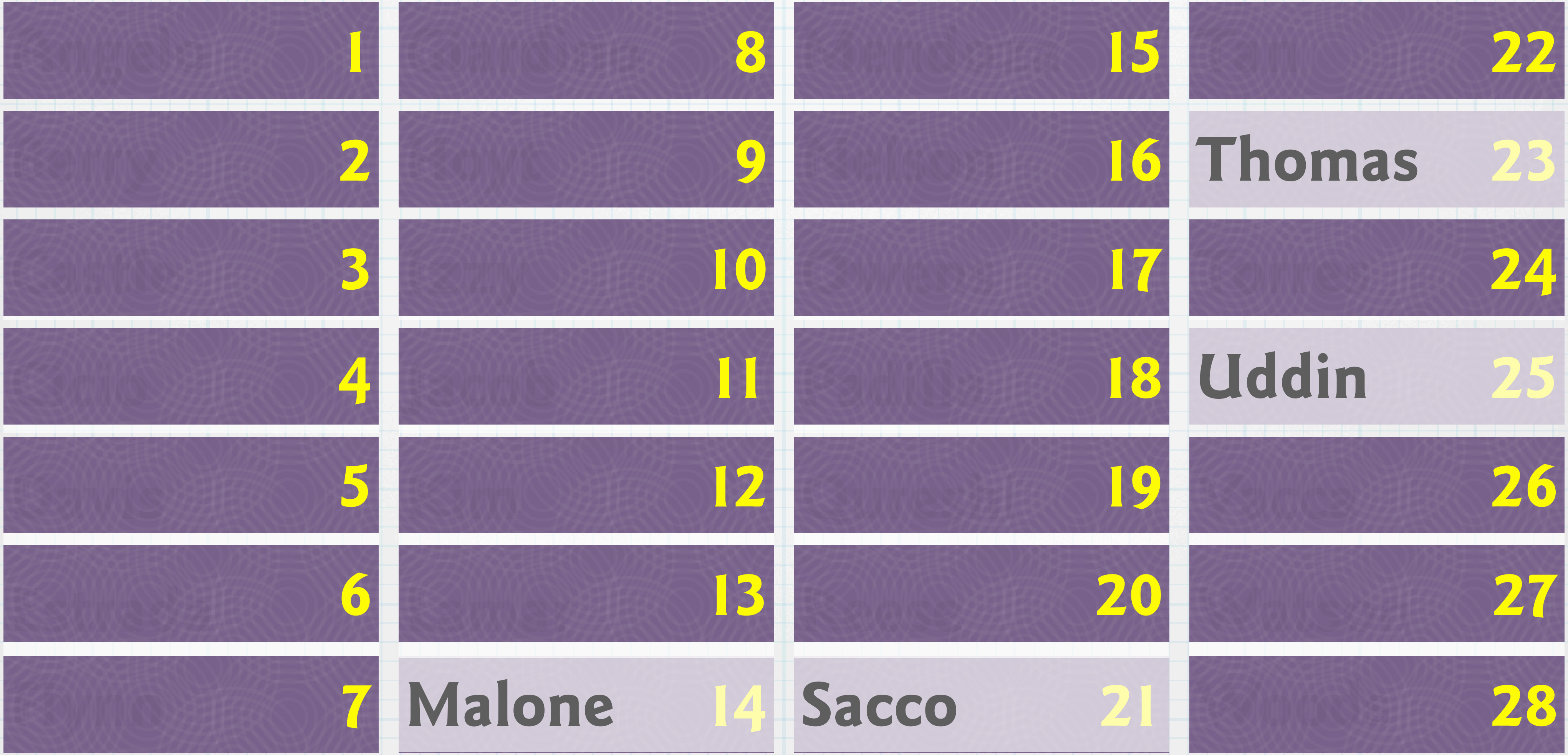
Searching for a name

Thomas | Search

1	8	15	22
2	9	16	23
3	10	17	24
4	11	18	Uddin 25
5	12	19	26
6	13	20	27
7	Malone 14	Sacco 21	28

Searching for a name

Thomas | Search



1	8	15	22
2	9	16	Thomas 23
3	10	17	24
4	11	18	Uddin 25
5	12	19	26
6	13	20	27
7	Malone 14	Sacco 21	28

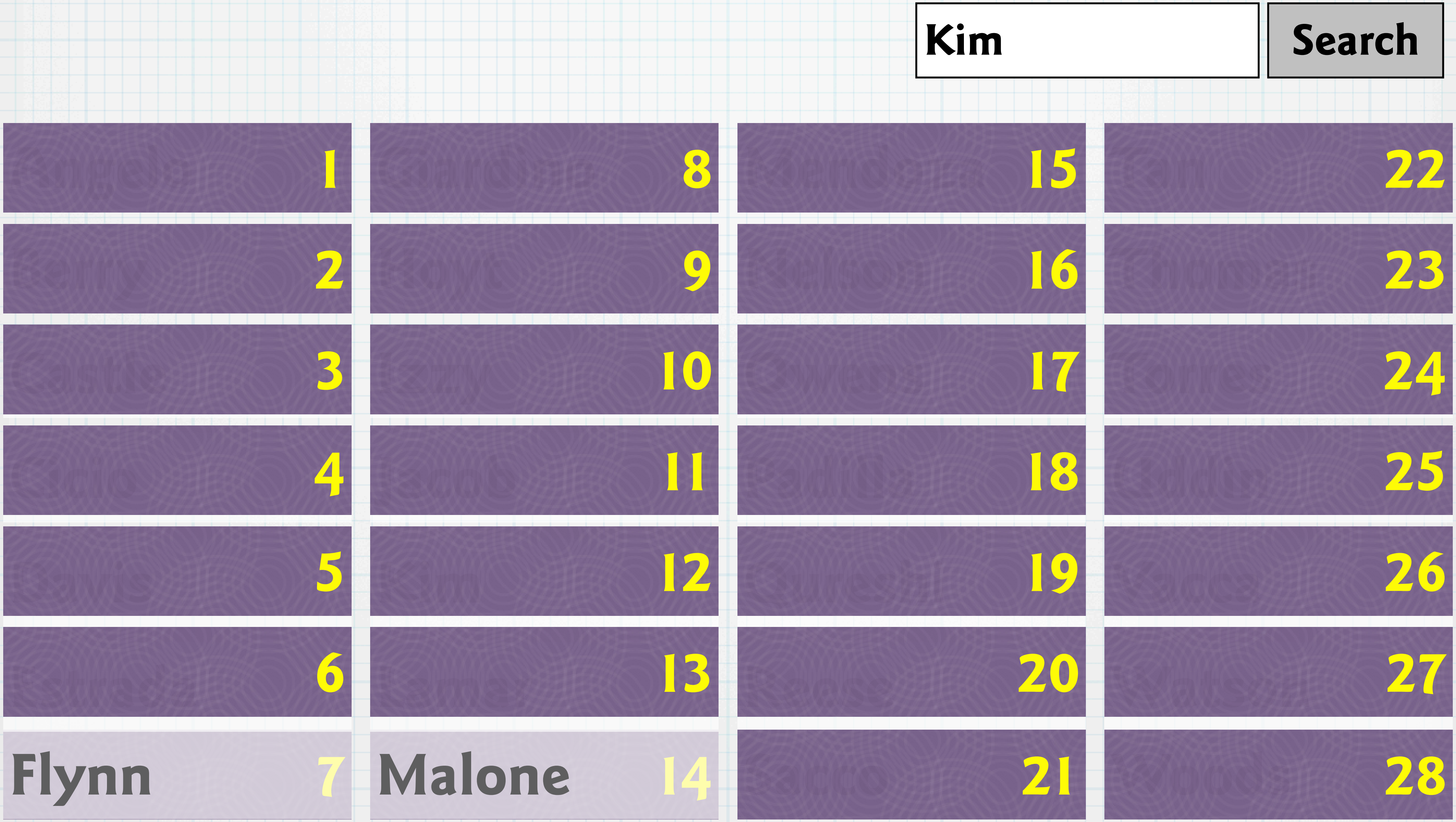
Searching for a name

Kim

Search

1	8	15	22
2	9	16	23
3	10	17	24
4	11	18	25
5	12	19	26
6	13	20	27
7	Malone 14	21	28

Searching for a name



1	8	15	22
2	9	16	23
3	10	17	24
4	11	18	25
5	12	19	26
6	13	20	27
Flynn 7	Malone 14	21	28

Searching for a name

Kim | Search

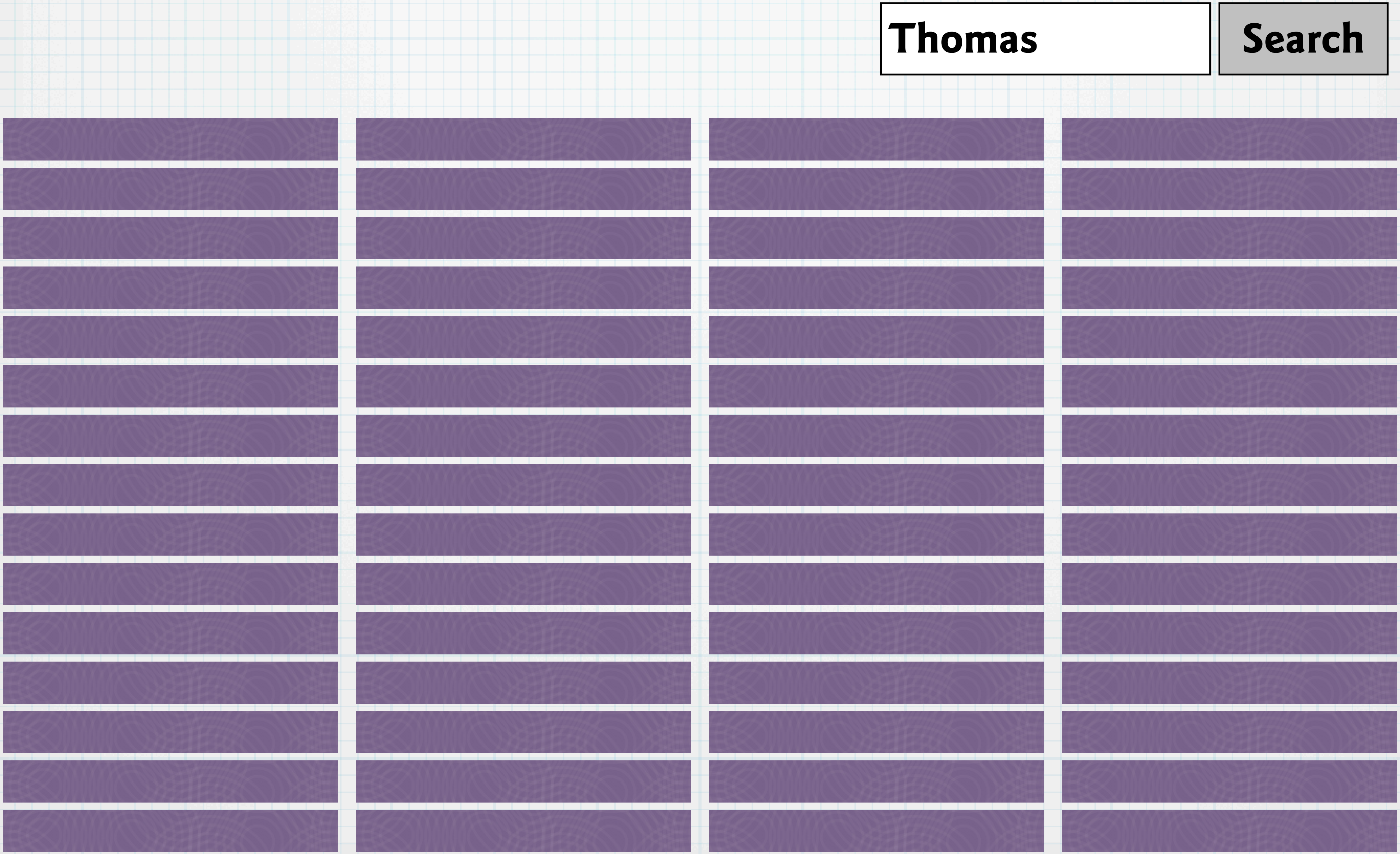
1	8	15	22
2	9	16	23
3	Izzy 10	17	24
4	11	18	25
5	12	19	26
6	13	20	27
Flynn 7	Malone 14	21	28

Searching for a name

Kim Search

1	8	15	22
2	9	16	23
3	Izzy 10	17	24
4	11	18	25
5	Kim 12	19	26
6	13	20	27
Flynn 7	Malone 14	21	28

Size of search space



Thomas

Searching for a name in the phone book

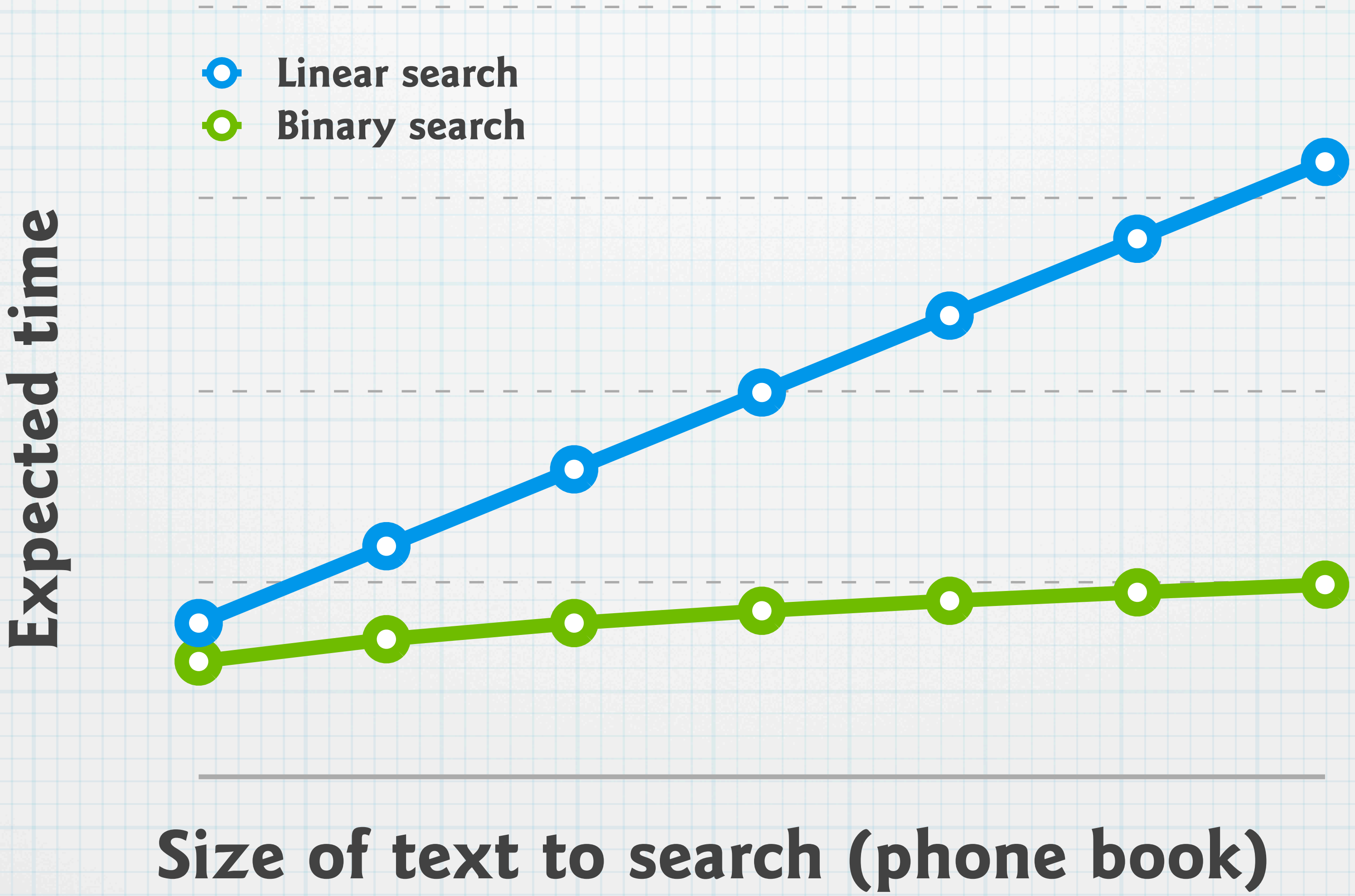
Searching for a name in the phone book

- How does representation affect efficiency?

Searching for a name in the phone book

- How does representation affect efficiency?
  - Imagine if names were not alphabetized!

Speed of search algorithms



A related problem: sorting

- How do you alphabetize a list of names?
- Are there different techniques?
- Are some faster than others?
- How do we measure the speed?

How many comparisons to find the smallest?

bat
hay
egg
ant
foe
cop
gum
duo

# comparisons: 1

How many comparisons to find the smallest?

bat ←
hay
egg ←
ant
foe
cop
gum
duo

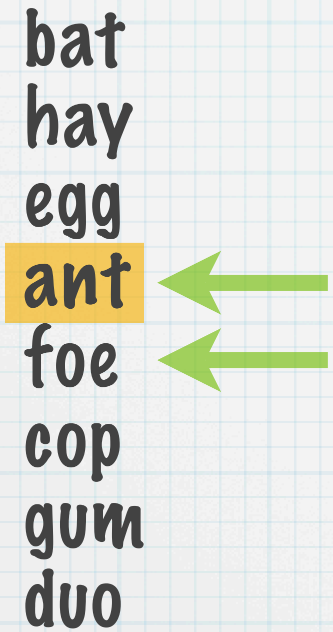
# comparisons: 2

How many comparisons to find the smallest?

bat ←
hay
egg
ant ←
foe
cop
gum
duo

# comparisons: 3

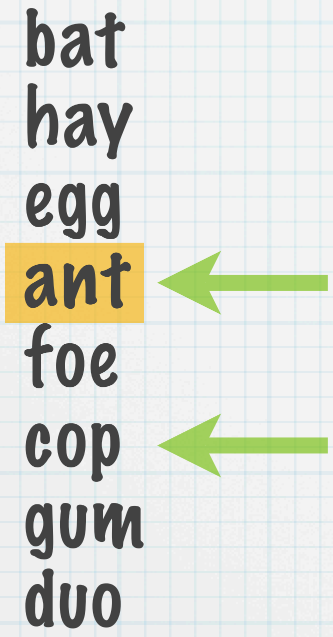
How many comparisons to find the smallest?



bat
hay
egg
ant
foe
cop
gum
duo

# comparisons: 4

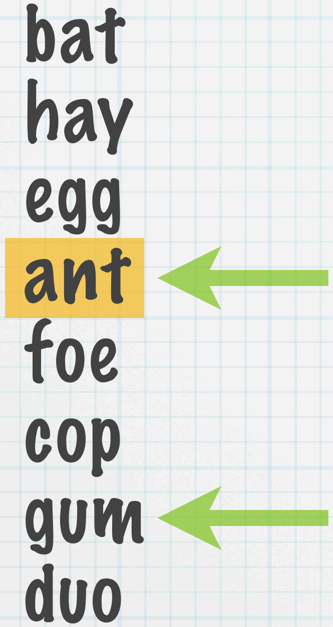
How many comparisons to find the smallest?



bat
hay
egg
ant
foe
cop
gum
duo

# comparisons: 5

How many comparisons to find the smallest?



bat
hay
egg
ant
foe
cop
gum
duo

# comparisons: 6

How many comparisons to find the smallest?

bat
hay
egg
ant ←
foe
cop
gum
duo ←

# comparisons: 7

Move smallest to top

bat	ant	
hay	bat	
egg	hay	◆
ant	egg	◆
foe	foe	◆
cop	cop	◆
gum	gum	◆
duo	duo	◆

# comparisons: 7+6

Move smallest to top

bat	ant	ant	
hay	bat	bat	
egg	hay	hay	
ant	egg	egg	◆
foe	foe	foe	◆
cop	cop	cop	◆
gum	gum	gum	◆
duo	duo	duo	◆

# comparisons: 7+6+5

Move smallest to top

bat	ant	ant	ant	
hay	bat	bat	bat	
egg	hay	hay	cop	
ant	egg	egg	hay	
foe	foe	foe	egg	◆
cop	cop	cop	foe	◆
gum	gum	gum	gum	◆
duo	duo	duo	duo	◆

# comparisons: 7+6+5+4

Move smallest to top

bat	ant	ant	ant	ant	
hay	bat	bat	bat	bat	
egg	hay	hay	cop	cop	
ant	egg	egg	hay	duo	
foe	foe	foe	egg	hay	
cop	cop	cop	foe	egg	◆
gum	gum	gum	gum	foe	◆
duo	duo	duo	duo	gum	◆

# comparisons: 7+6+5+4+3

Move smallest to top

ant	ant	ant	ant	ant	
bat	bat	bat	bat	bat	
hay	hay	cop	cop	cop	
egg	egg	hay	duo	duo	
foe	foe	egg	hay	egg	
cop	cop	foe	egg	hay	
gum	gum	gum	foe	foe	◆
duo	duo	duo	gum	gum	◆

# comparisons: 7+6+5+4+3+2

Move smallest to top

ant	ant	ant	ant	ant
bat	bat	bat	bat	bat
hay	cop	cop	cop	cop
egg	hay	duo	duo	duo
foe	egg	hay	egg	egg
cop	foe	egg	hay	foe
gum	gum	foe	foe	hay
duo	duo	gum	gum	gum

# comparisons: 7+6+5+4+3+2+1

Move smallest to top

ant	ant	ant	ant	ant
bat	bat	bat	bat	bat
cop	cop	cop	cop	cop
hay	duo	duo	duo	duo
egg	hay	egg	egg	egg
foe	egg	hay	foe	foe
gum	foe	foe	hay	gum
duo	gum	gum	gum	hay

# comparisons: 7+6+5+4+3+2+1 = 28

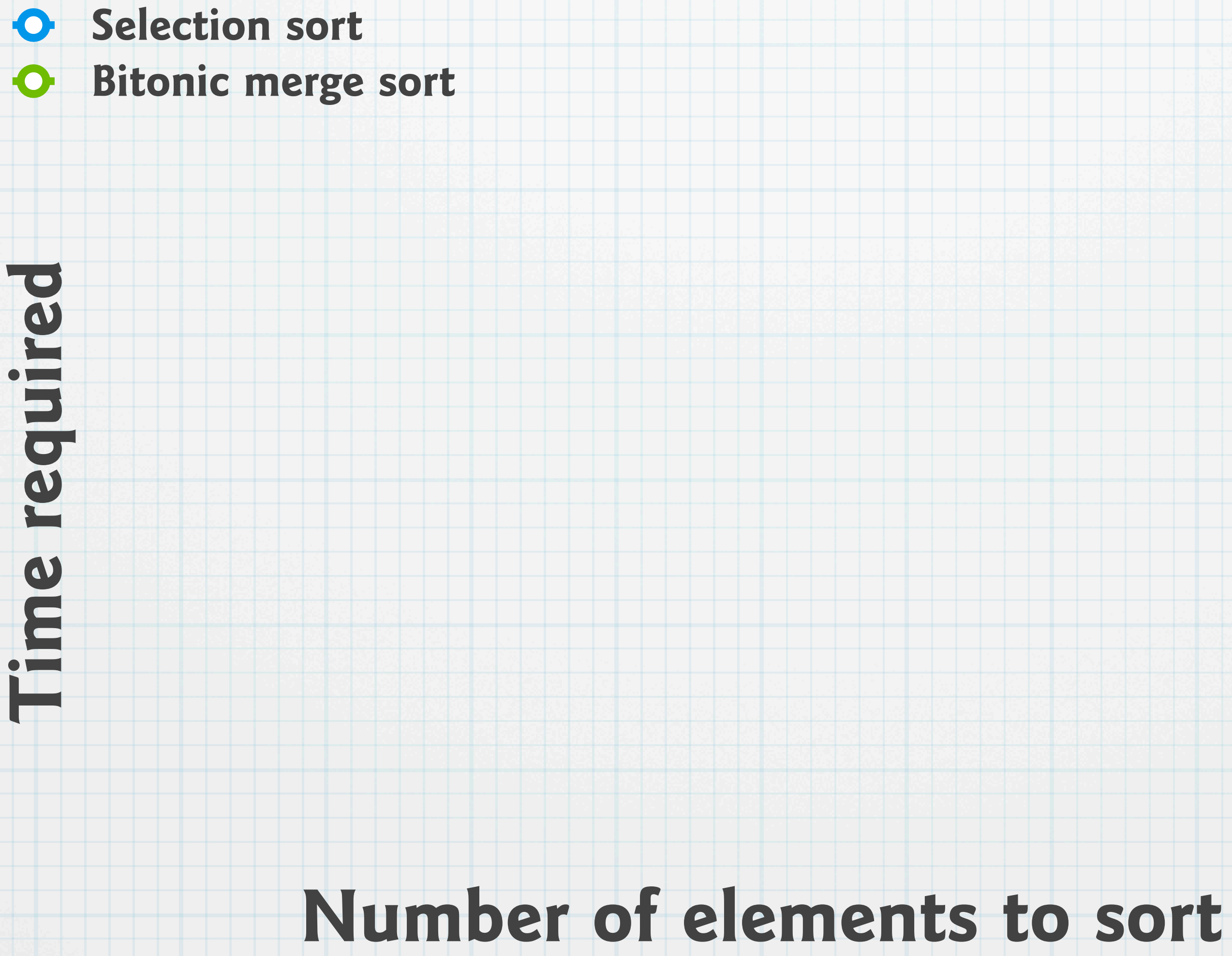
A faster technique

- What we did is called **selection sort**
  - One of the most obvious, but slowest
- Many faster sorting algorithms have been developed:
  - Quick sort, merge sort, ...
  - **Bitonic sorting network**

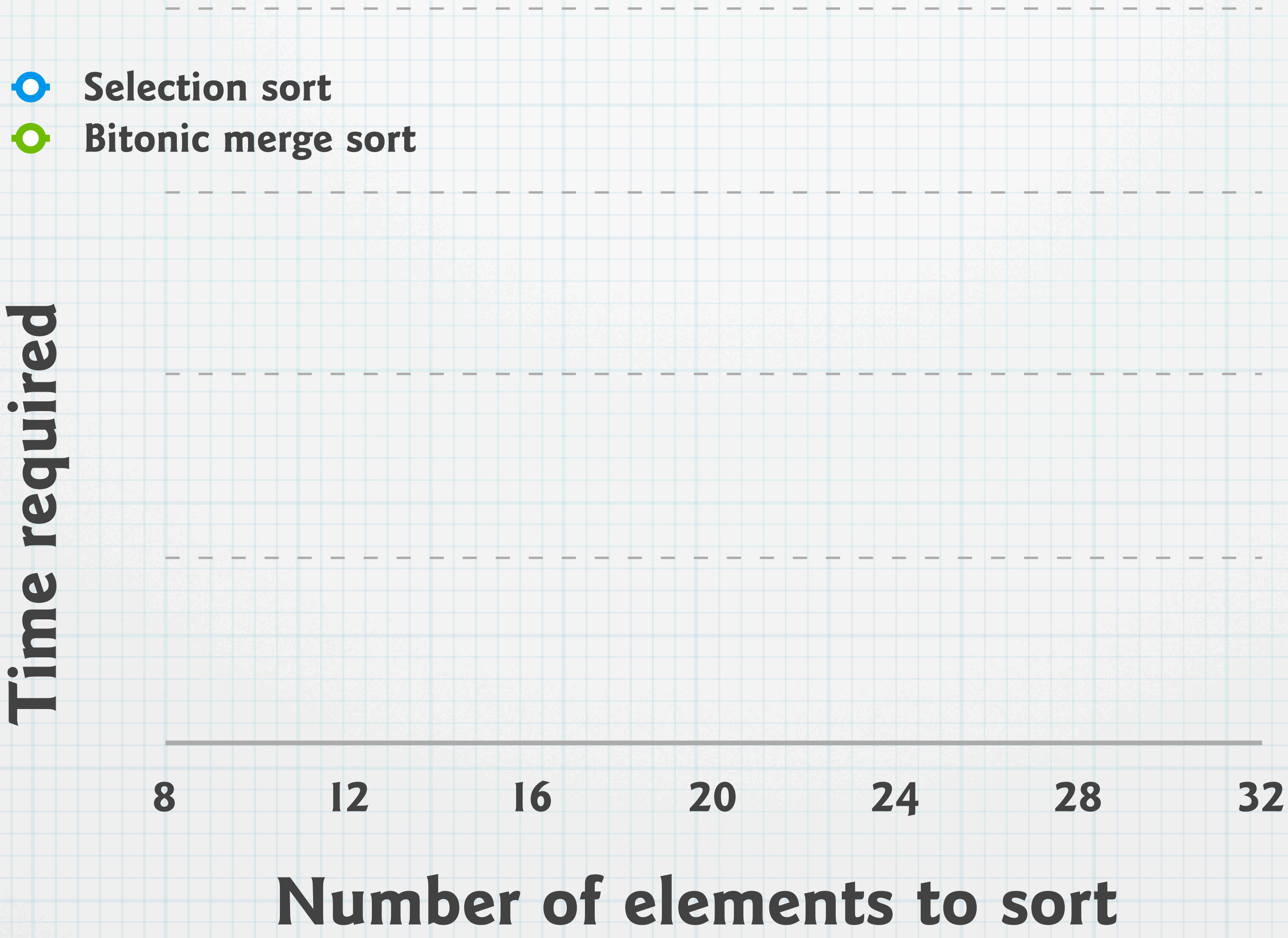
Speed of sort algorithms

- We sorted 8 elements:
  - Selection sort made 28 comparisons.
  - Bitonic merge sort made 24.
- BIG DEAL, you say.
  - But remember: the size of the problem matters!

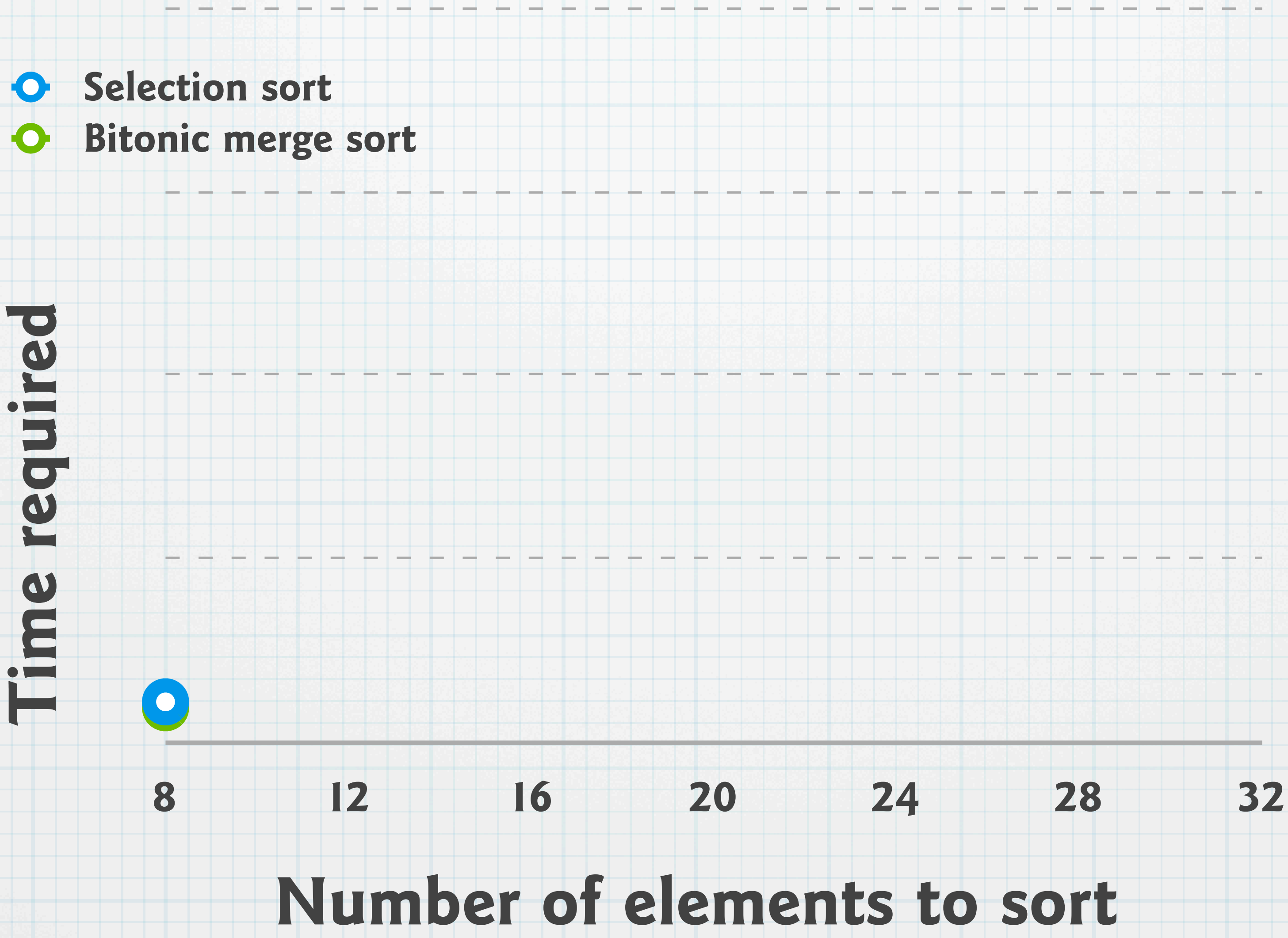
Speed of sort algorithms



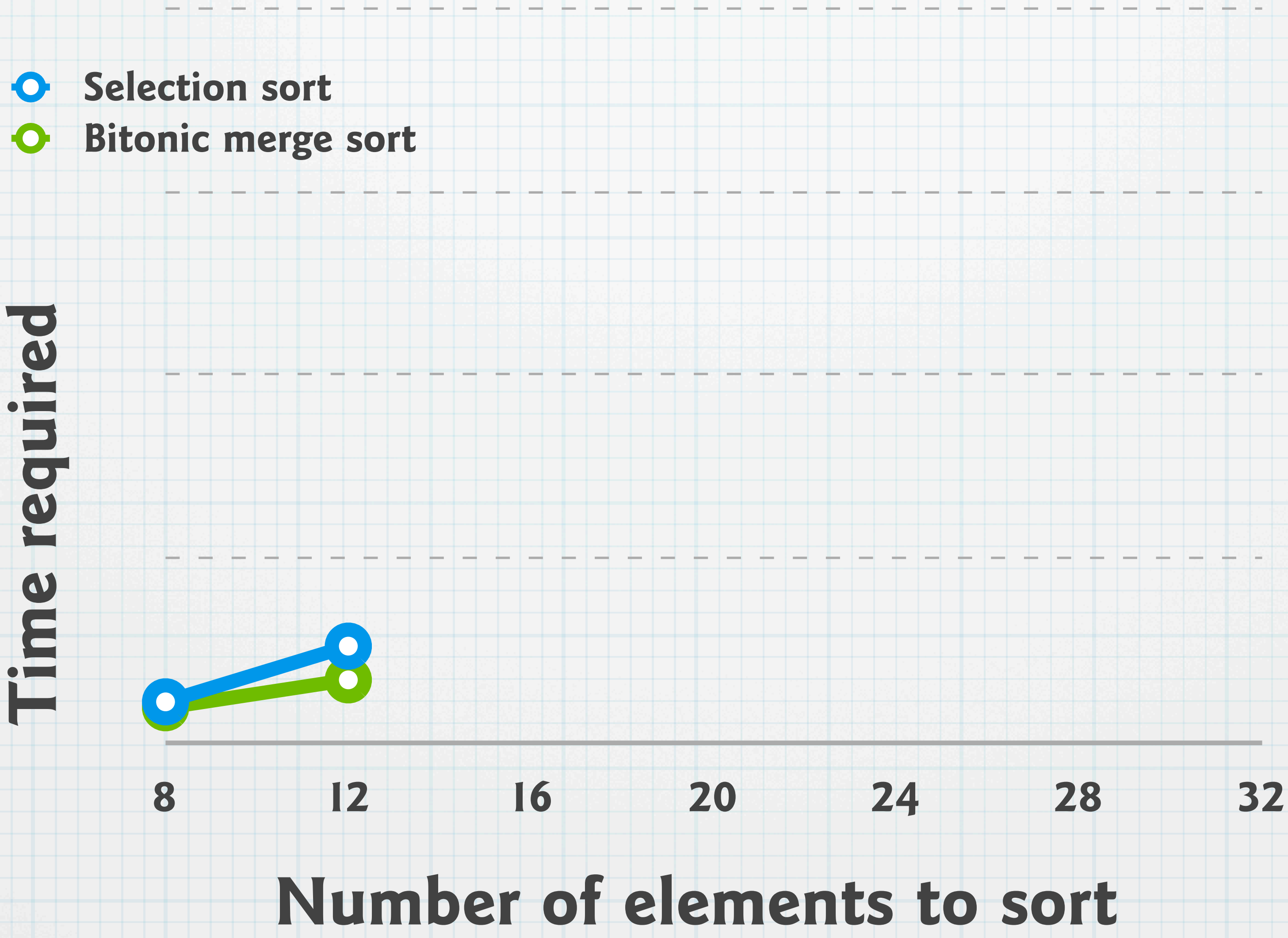
Speed of sort algorithms



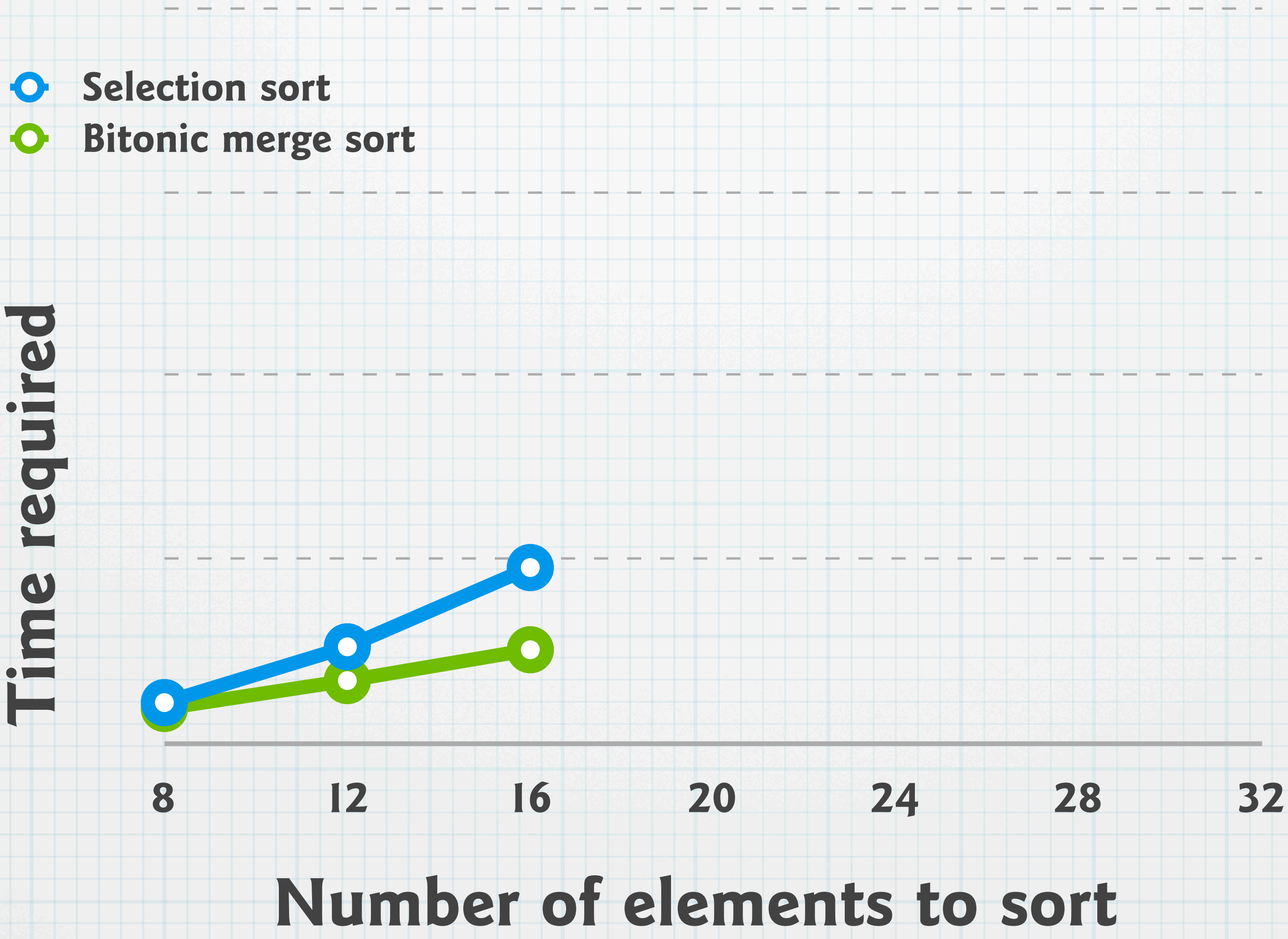
Speed of sort algorithms



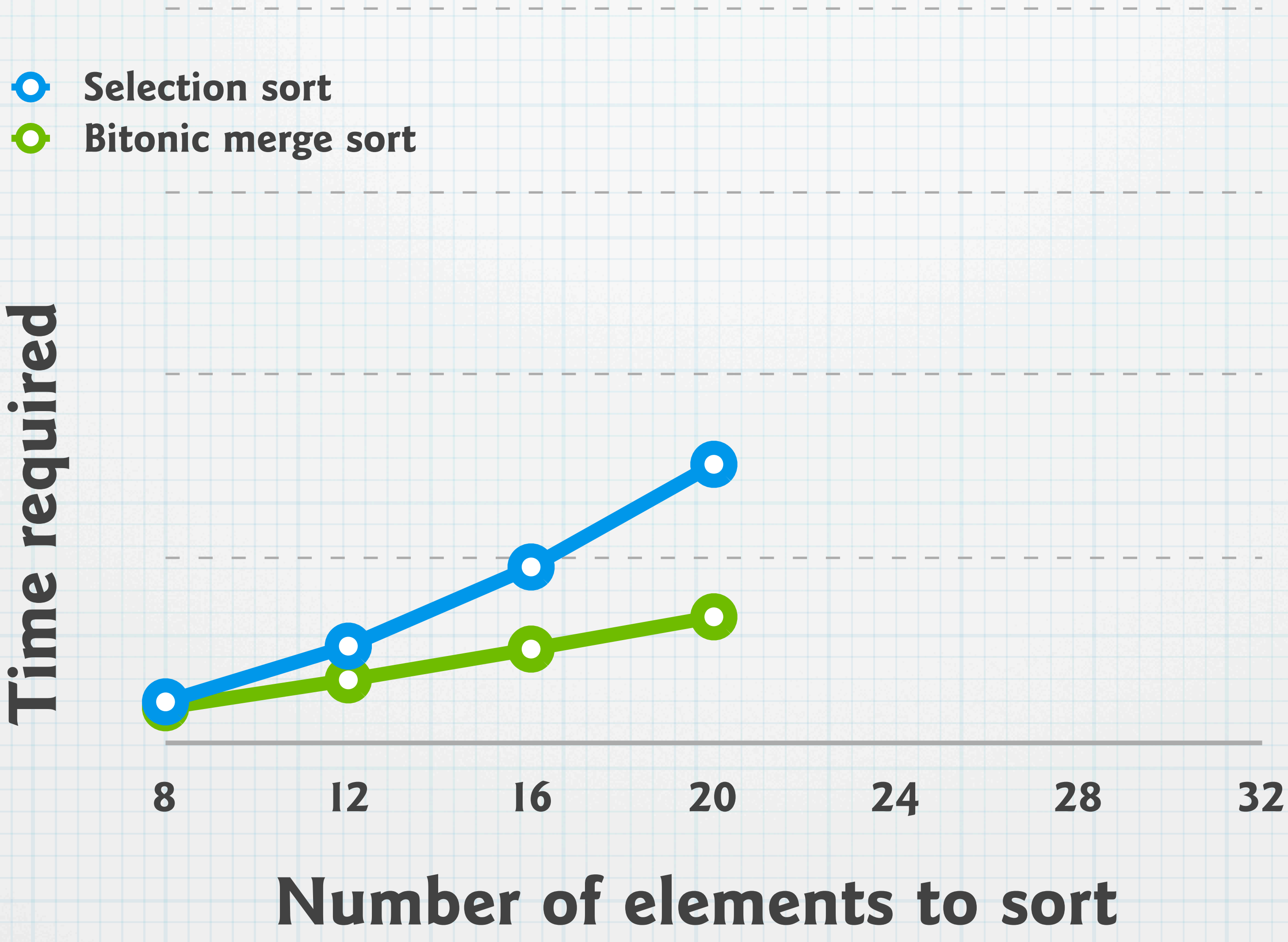
Speed of sort algorithms



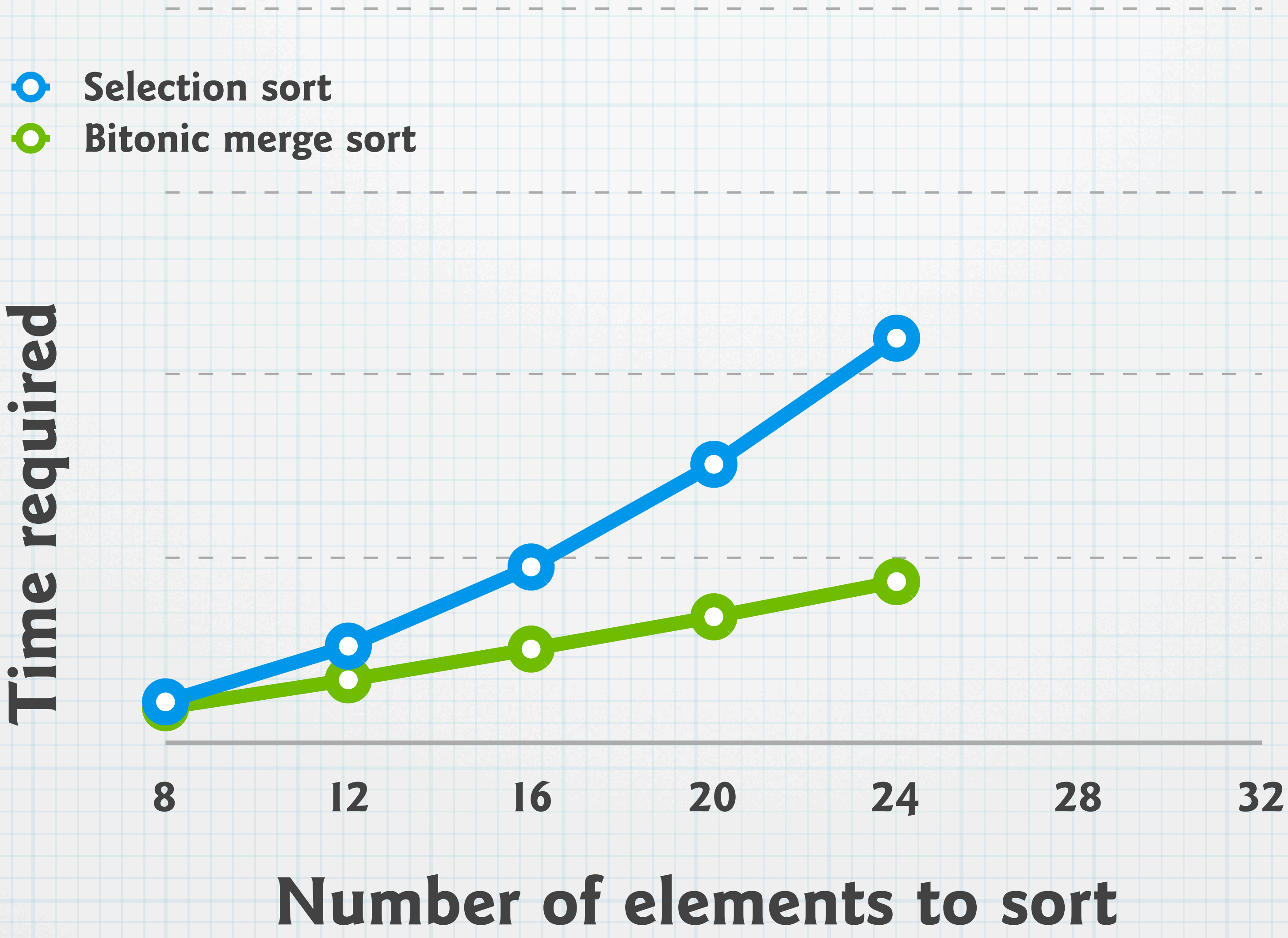
Speed of sort algorithms



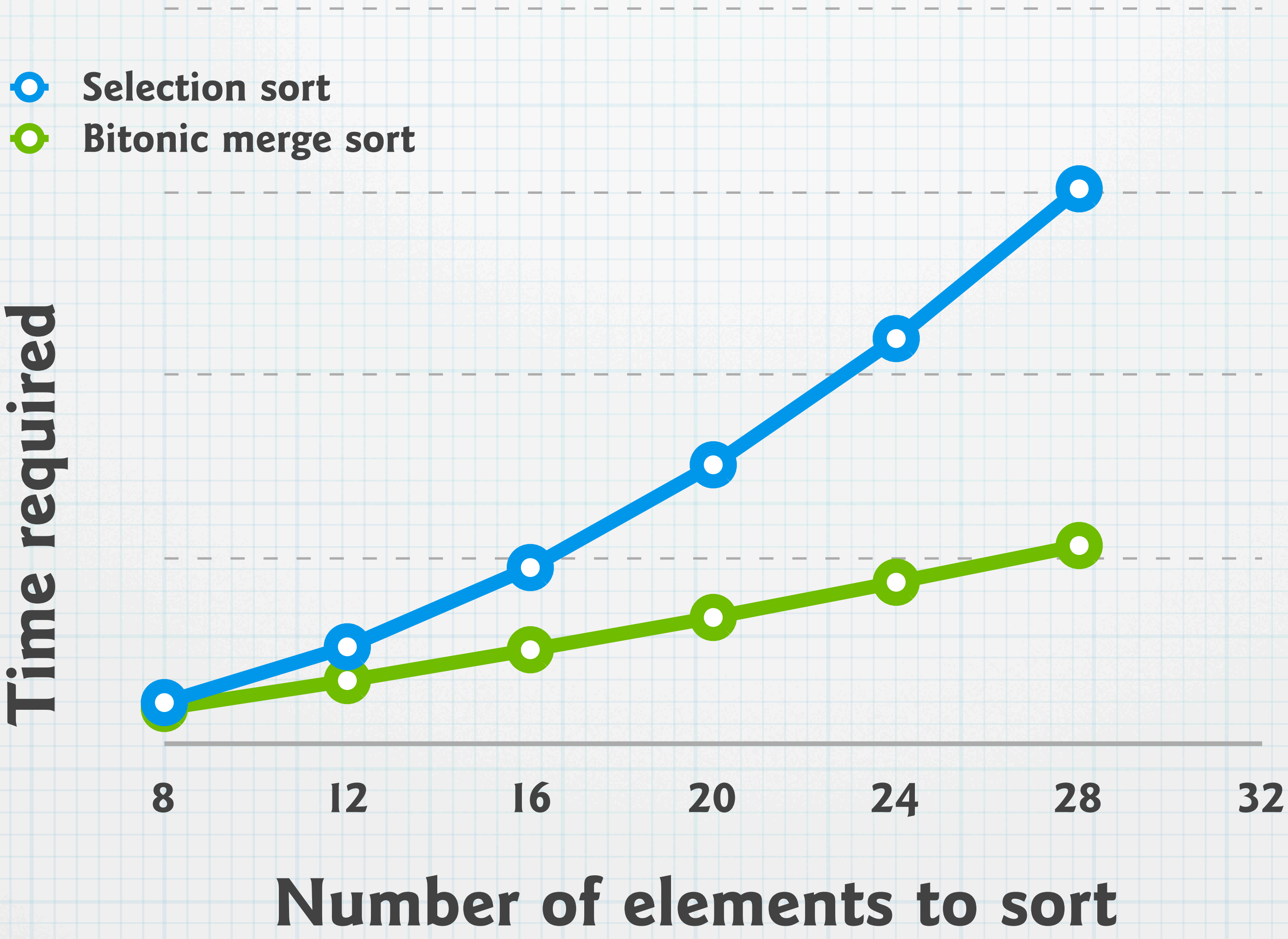
Speed of sort algorithms



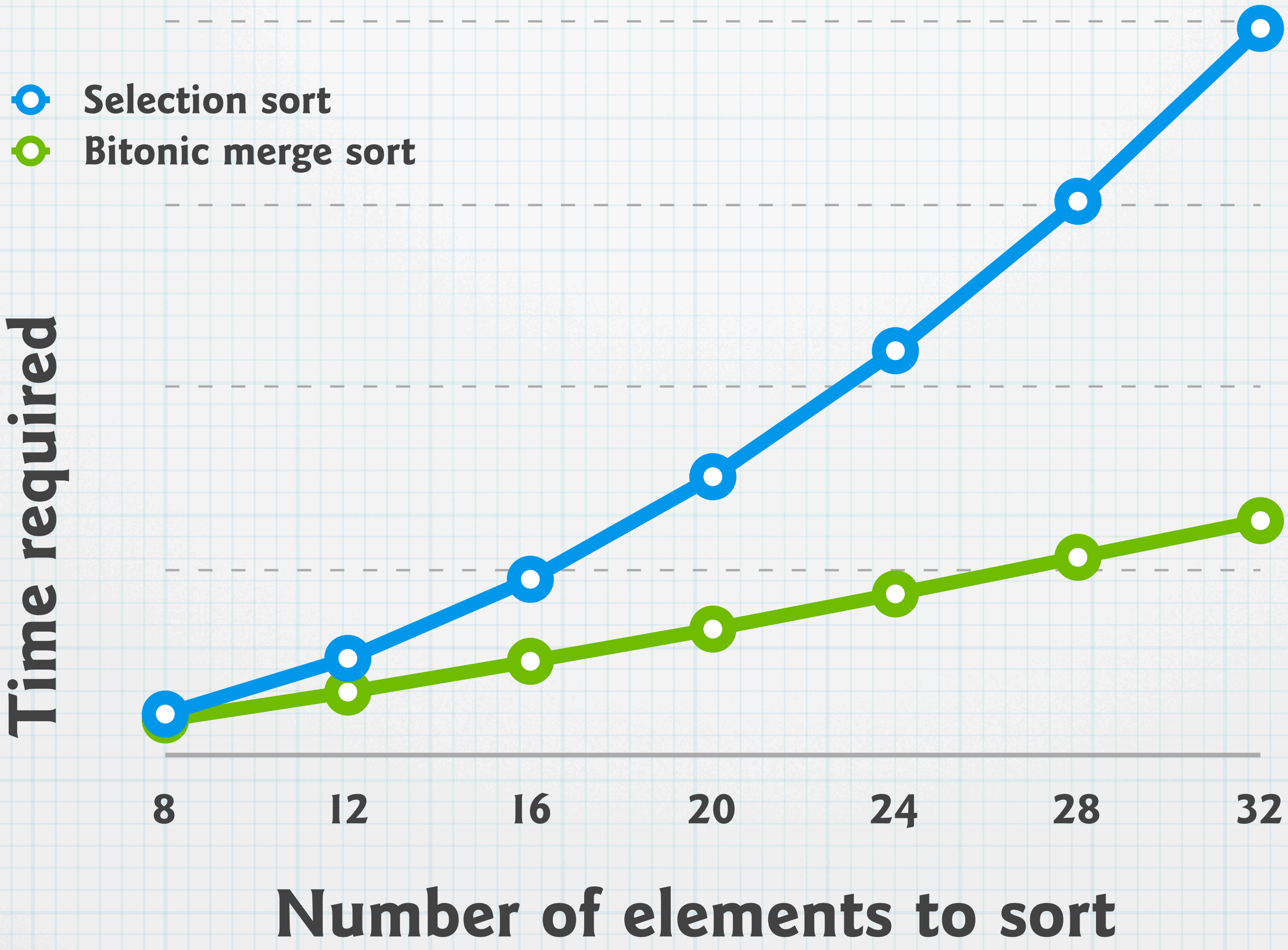
Speed of sort algorithms



Speed of sort algorithms



Speed of sort algorithms



Conclusions

- ‘Computing’ is about much more than the computer.
- It is about problem solving, computational processes, and information.