2019 Workshop on Gravitational Waves and High-Performance Computing

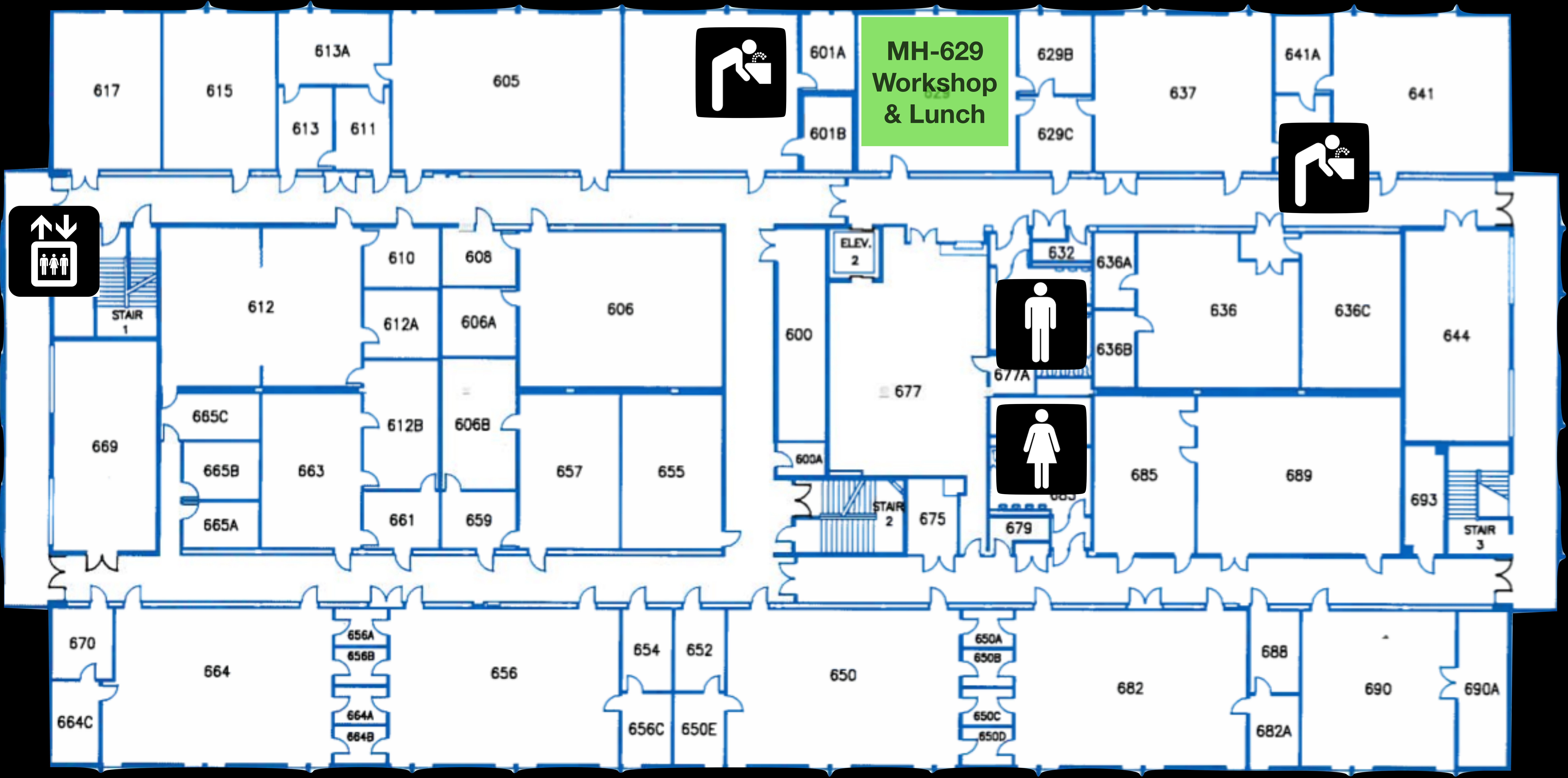
Geoffrey Lovelace

August 19, 2019 – August 23, 2019

Welcome to the workshop!

- Please take an ABCD card
- Donuts today
- Water next store in MH-601 or down the hall
- Workshop supported by the National Science Foundation

Welcome to the workshop!



Photos

- We would like to take photos during the workshop
- The photos would appear on the Cal State Fullerton website, in news stories about the workshop
- If you agree to have your picture taken, please check the box on the sign-in sheet

Daily schedule

- Start: 9:30 AM
- Morning break: 11:00AM - 11:20AM
- Lunch: 12:00PM - 1:30PM
- Afternoon break: 2:30PM - 2:50PM
- End: 4:00PM

Tentative schedule

- **Monday**: Powers of 10 & computing, programming with Python
- **Tuesday**: Programming with Python, Unix Command Line, using a supercomputer
- **Wednesday**: Black holes, gravitational waves, simulating colliding black holes
- **Thursday**: Gravitational-wave research, data center tour
- **Friday**: visualizing colliding black holes, exit survey

GW PAC GRAVITATIONAL WAVE
Physics and Astronomy Center

Which brother am I?

A

B

C Both

D Neither

E Not sure

Icebreaker

- If you had to gain one superpower, which one would you choose?

A	Ability to fly

B	Power to be invisible

Meter

- How many me across is Earth?

Powers of 10

- How many meters across is Earth?

A 10^6

B 10^7

C 10^8

D 10^9


100 million light years
10^{24}
meters

Powers of 10

- How many meters across is Earth?

A	10^6
B	10^7
C	10^8
D	10^9

Powers of 10

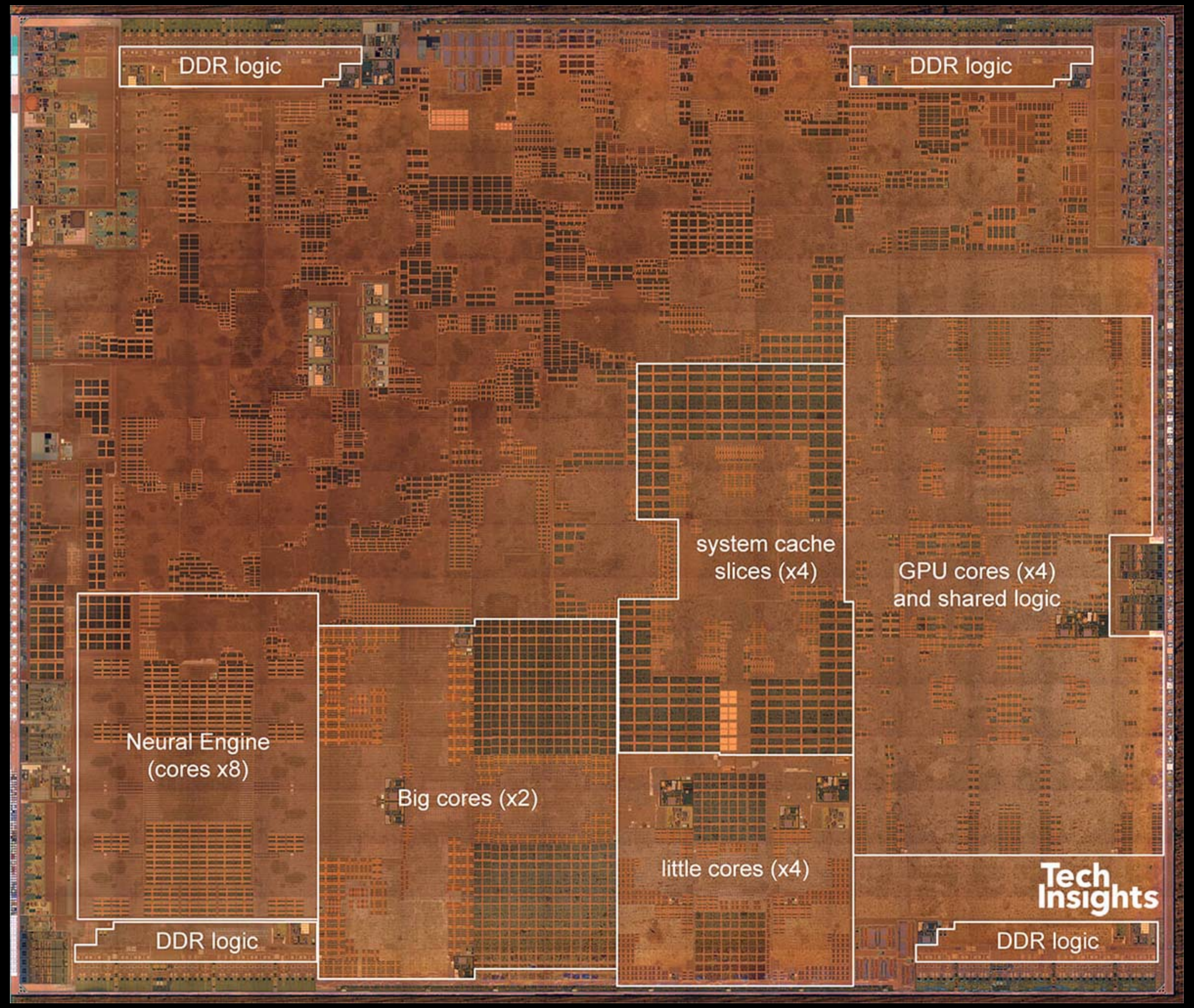
- How many meters is a light year?

A	10^8
B	10^{12}
C	10^{16}
D	10^{20}

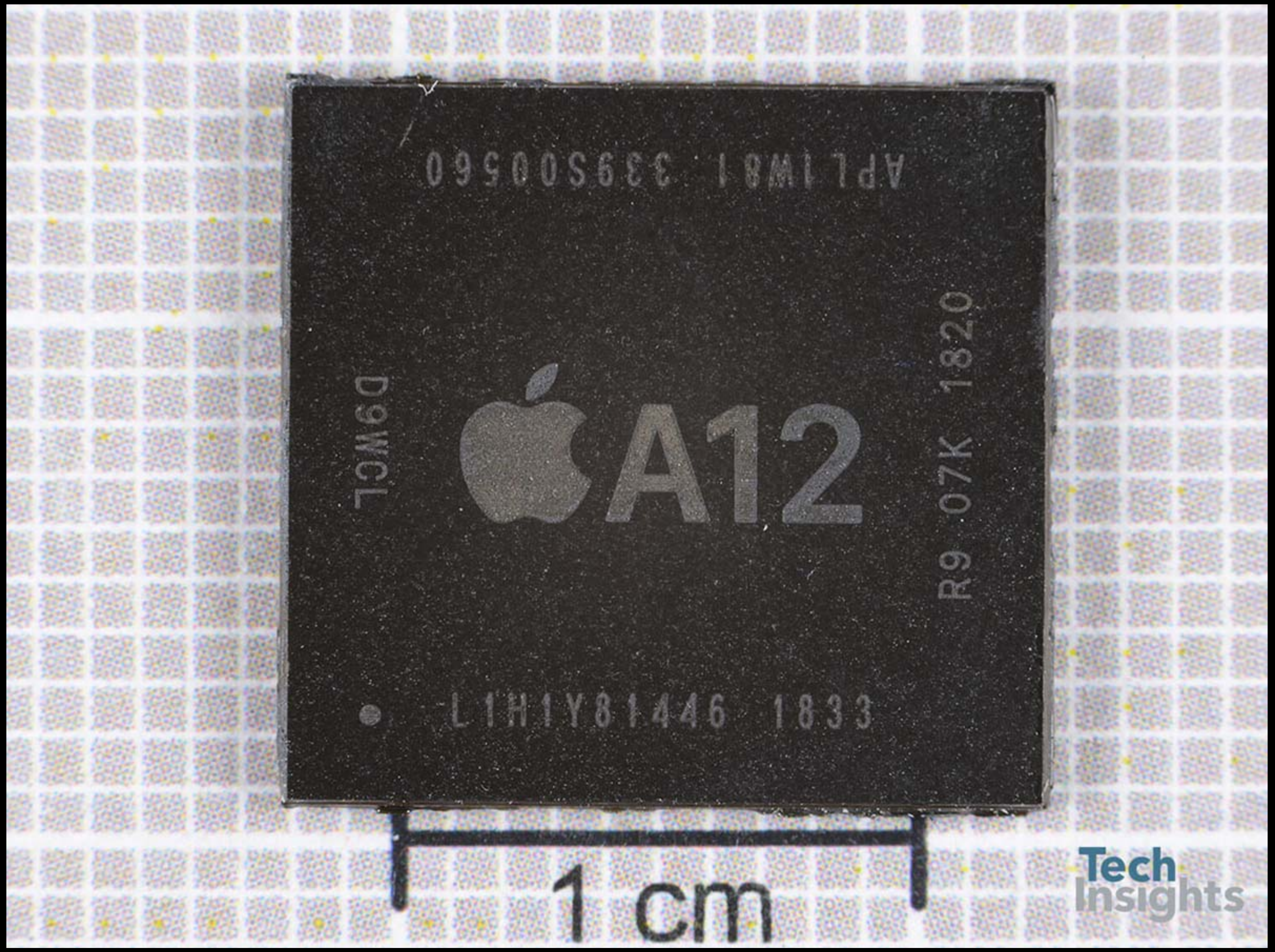
Powers of 10

- Take a guess:
  how many meters across is the "system on a chip" powering my iPhone?

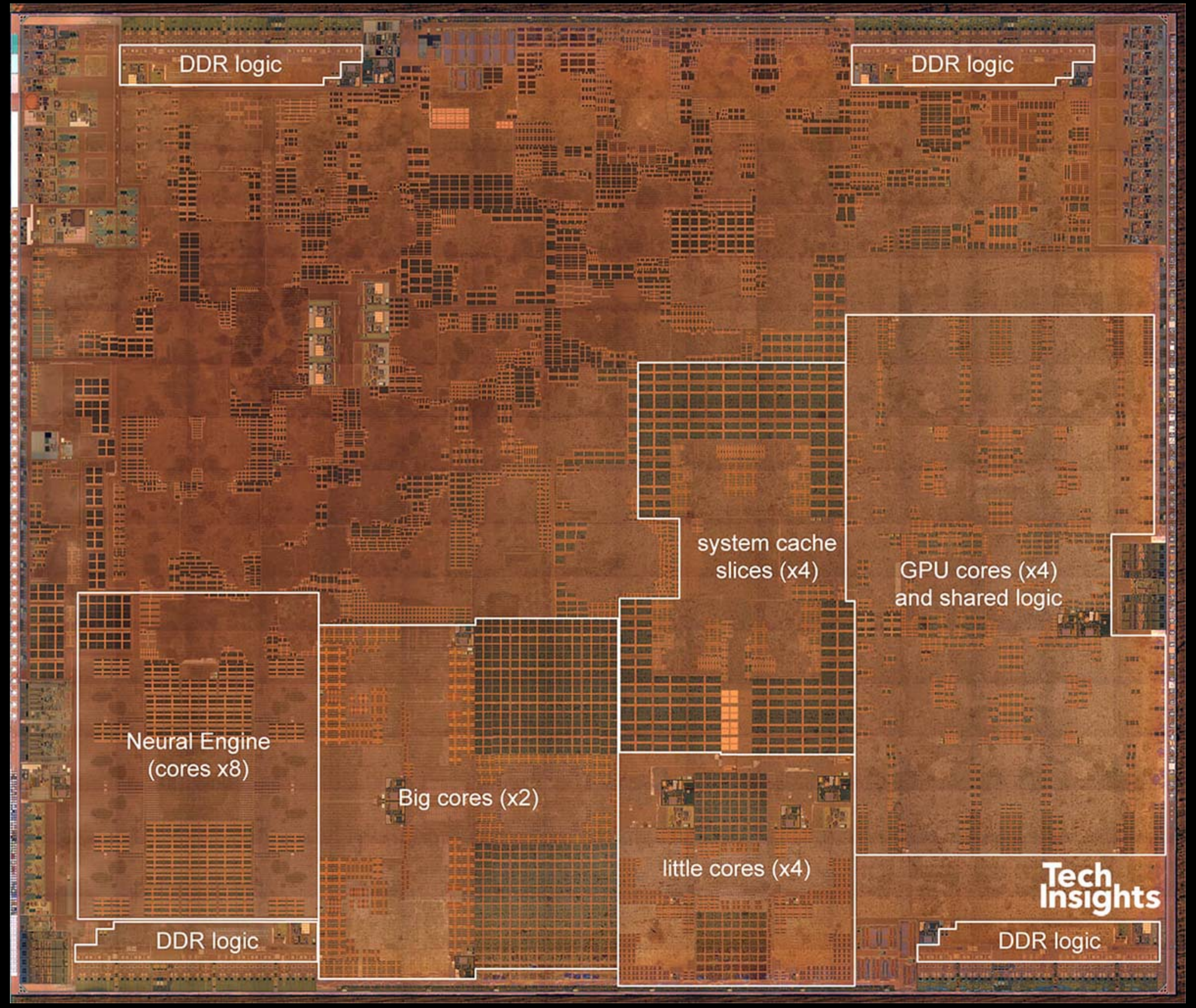
A	10^{-6}
B	10^{-4}
C	10^{-2}
D	10^{0}



Powers of 10

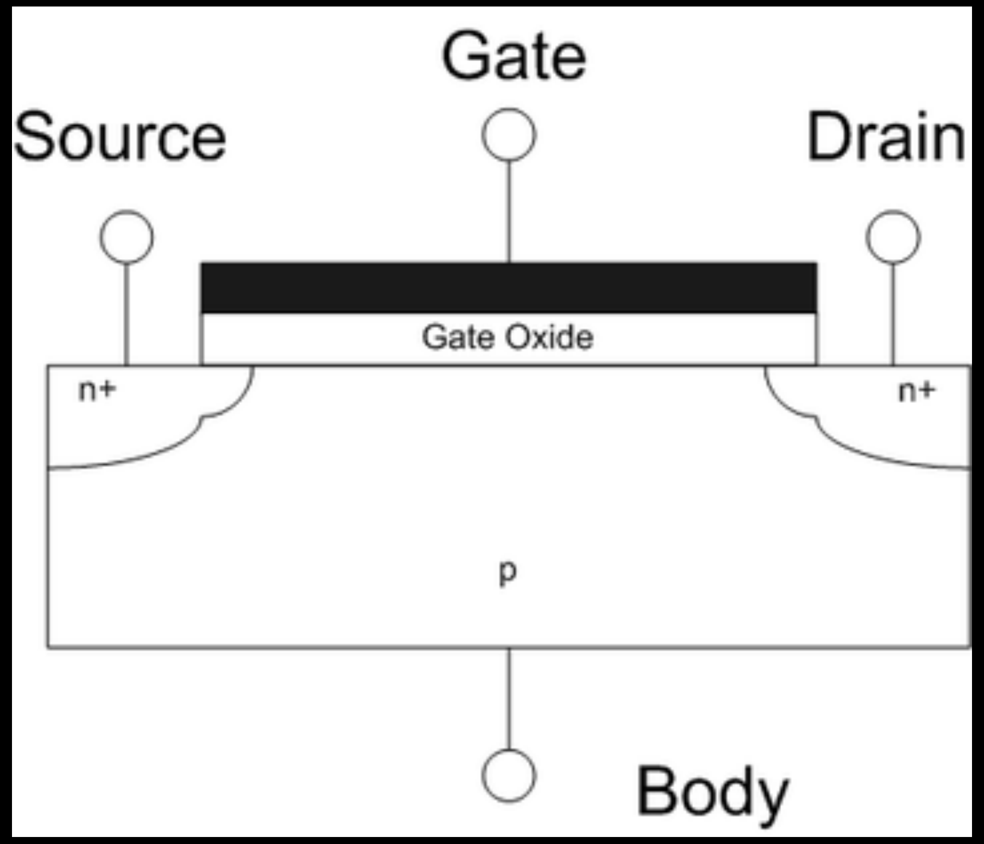


A	10^{-6}
B	10^{-4}
C	10^{-2}
D	10^{0}

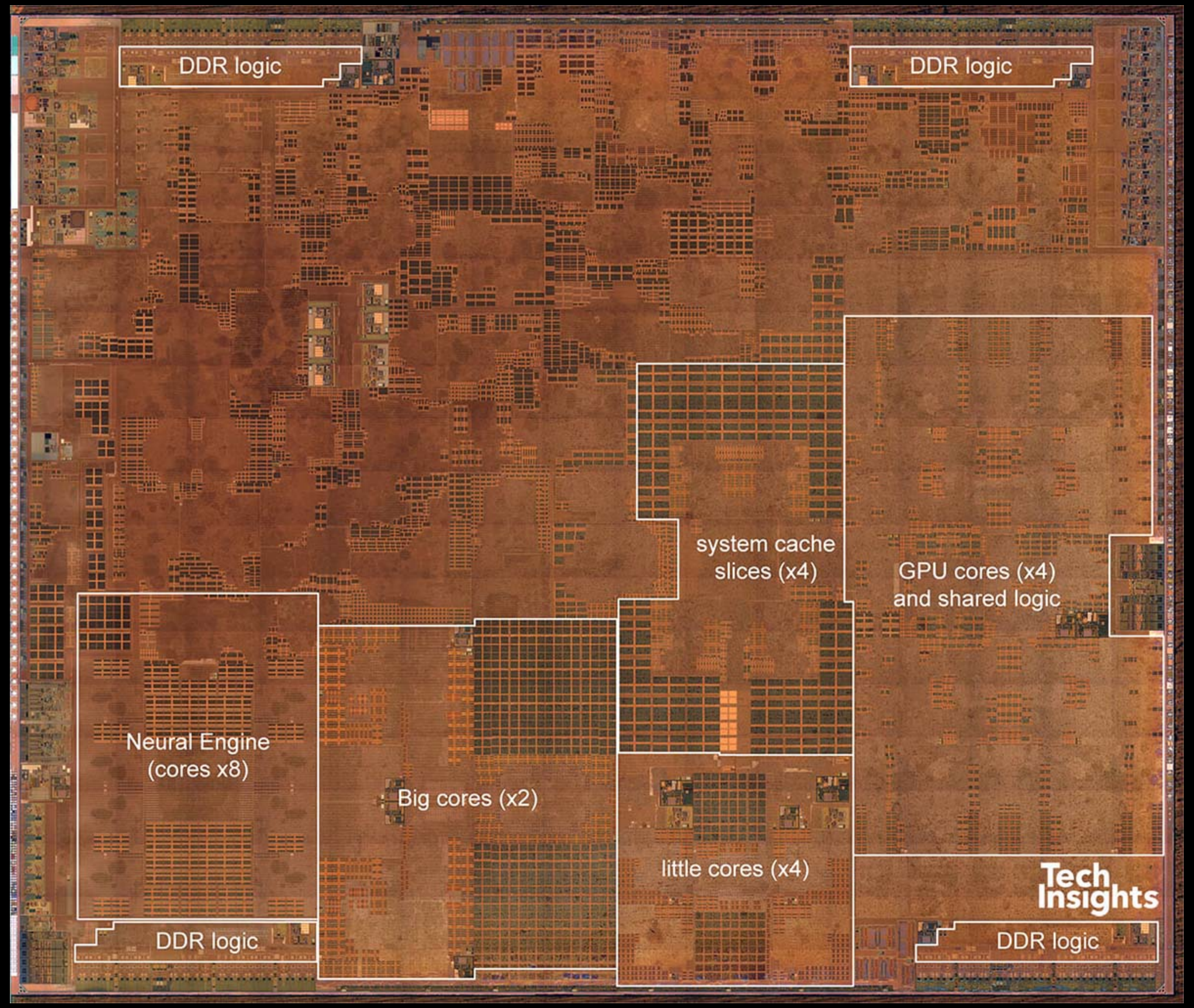


Powers of 10

- Take a guess:
  how big is a single transistor on this chip?
  - Transistors are little circuits that can be combined to make all the circuits
  - Chip ≈ 10^{-4} m^2
  - ≈ 10^{10} transistors on the entire chip

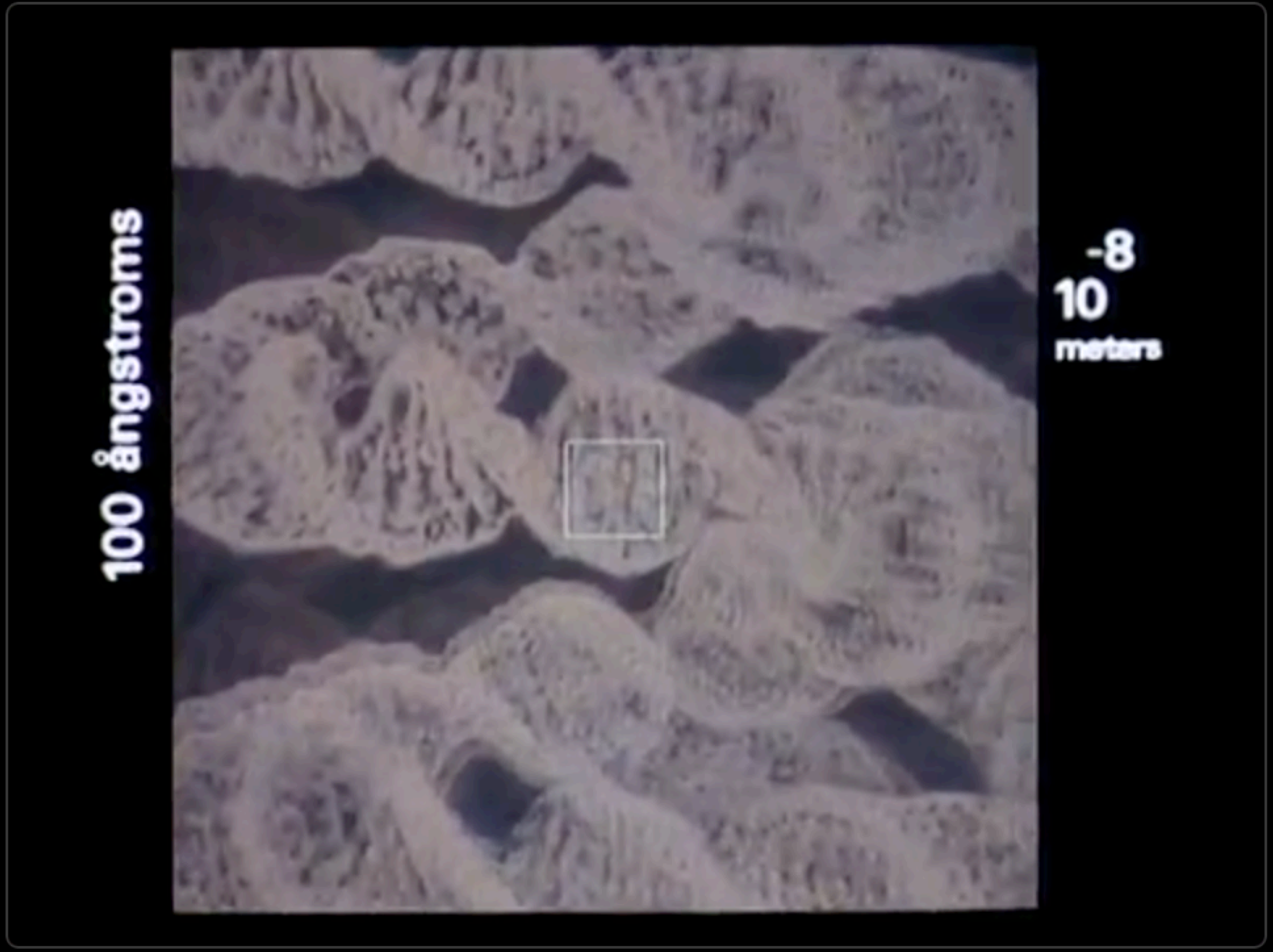


A	10^{-10}
B	10^{-8}
C	10^{-6}
D	10^{-4}



Powers of 10

- Take a guess:
  how big is a single
  transistor on this chip?

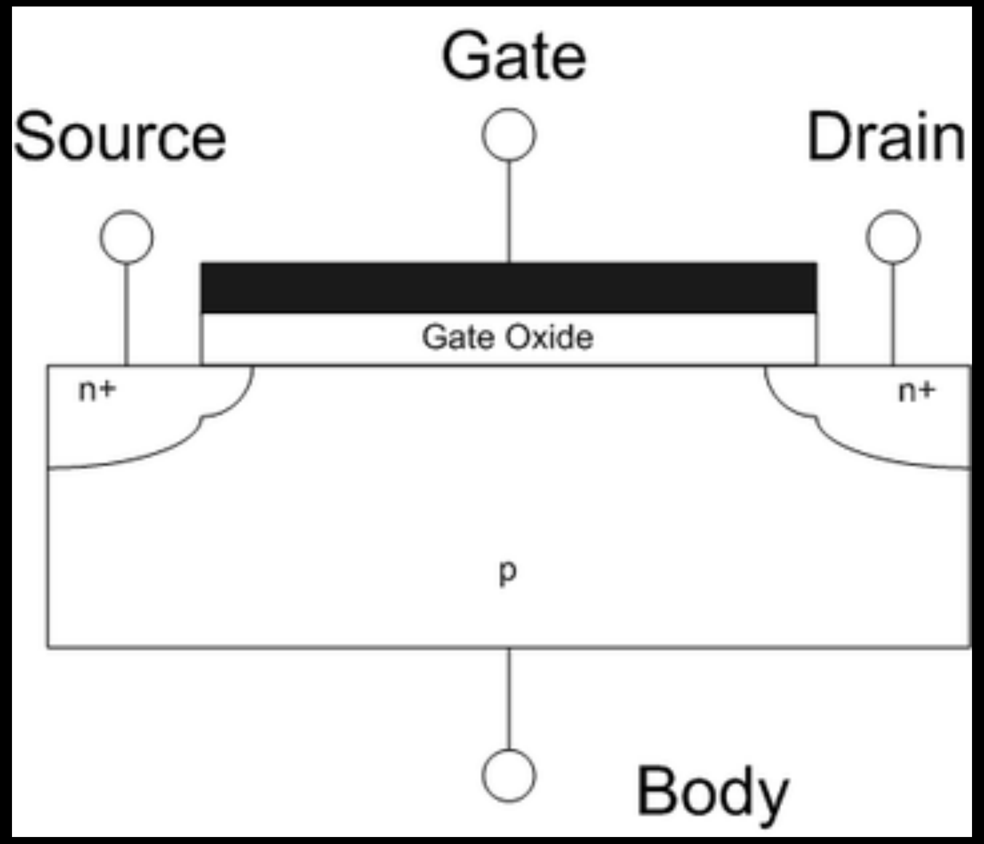


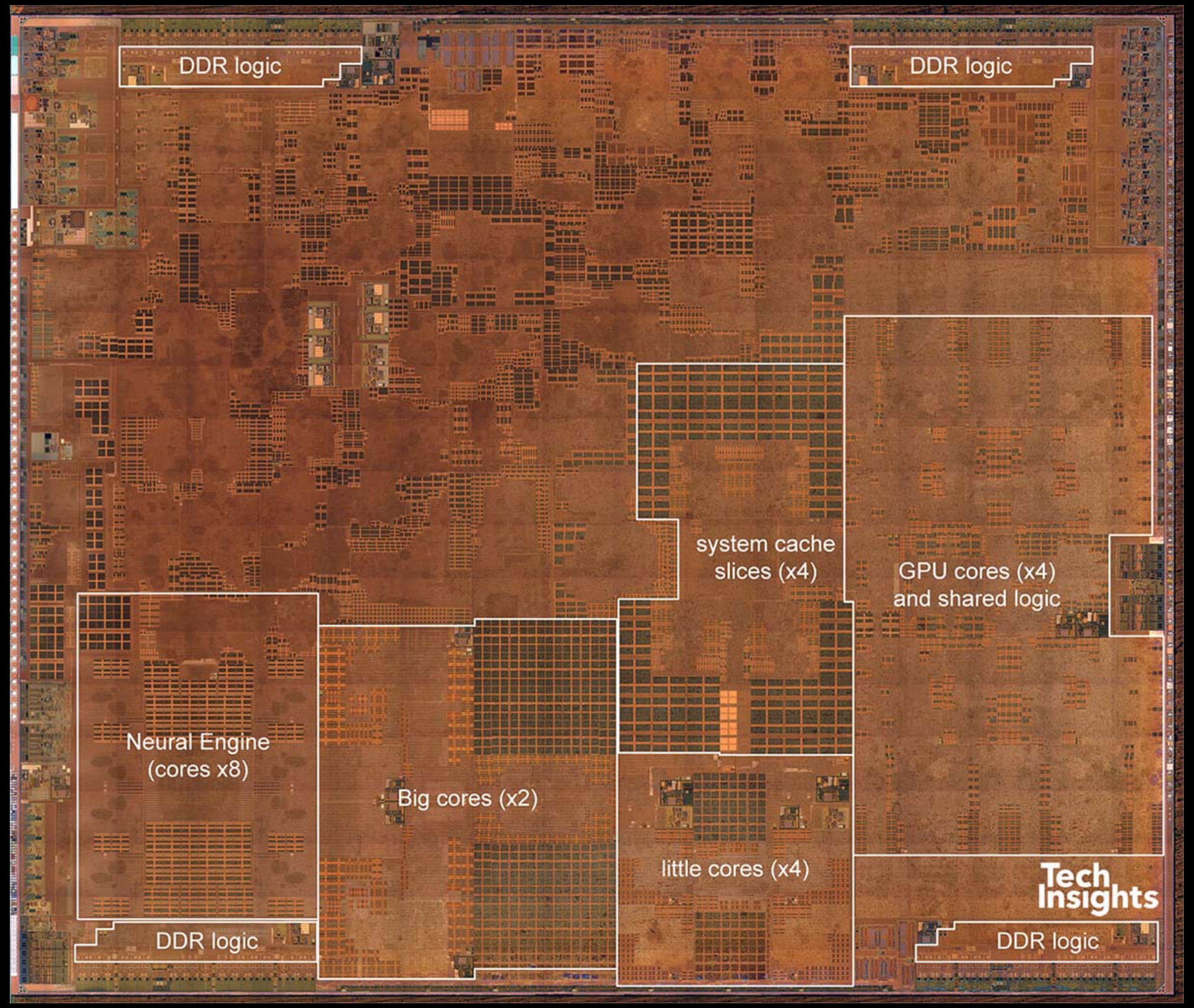
A 10^{-10}

B 10^{-8}

C 10^{-6}

D 10^{-4}





Powers of 10
& computers

Humans

- First entities called “computers” were teams of people
- Divide up the work into operations done in parallel, by hand (perhaps with mechanical aid)
- Redundant calculations to check accuracy
- Since 1700s
- 10^{-1} to 1 FLOPS / human (decimal operations / second / human)

1949 NACA High Speed Flight Station “Computer Room”)

Image courtesy wikipedia

Colossus (1942)

- First programmable, digital, electronic computer
- Break codes in World War II Britain
- 5 x 10^5 FLOPS

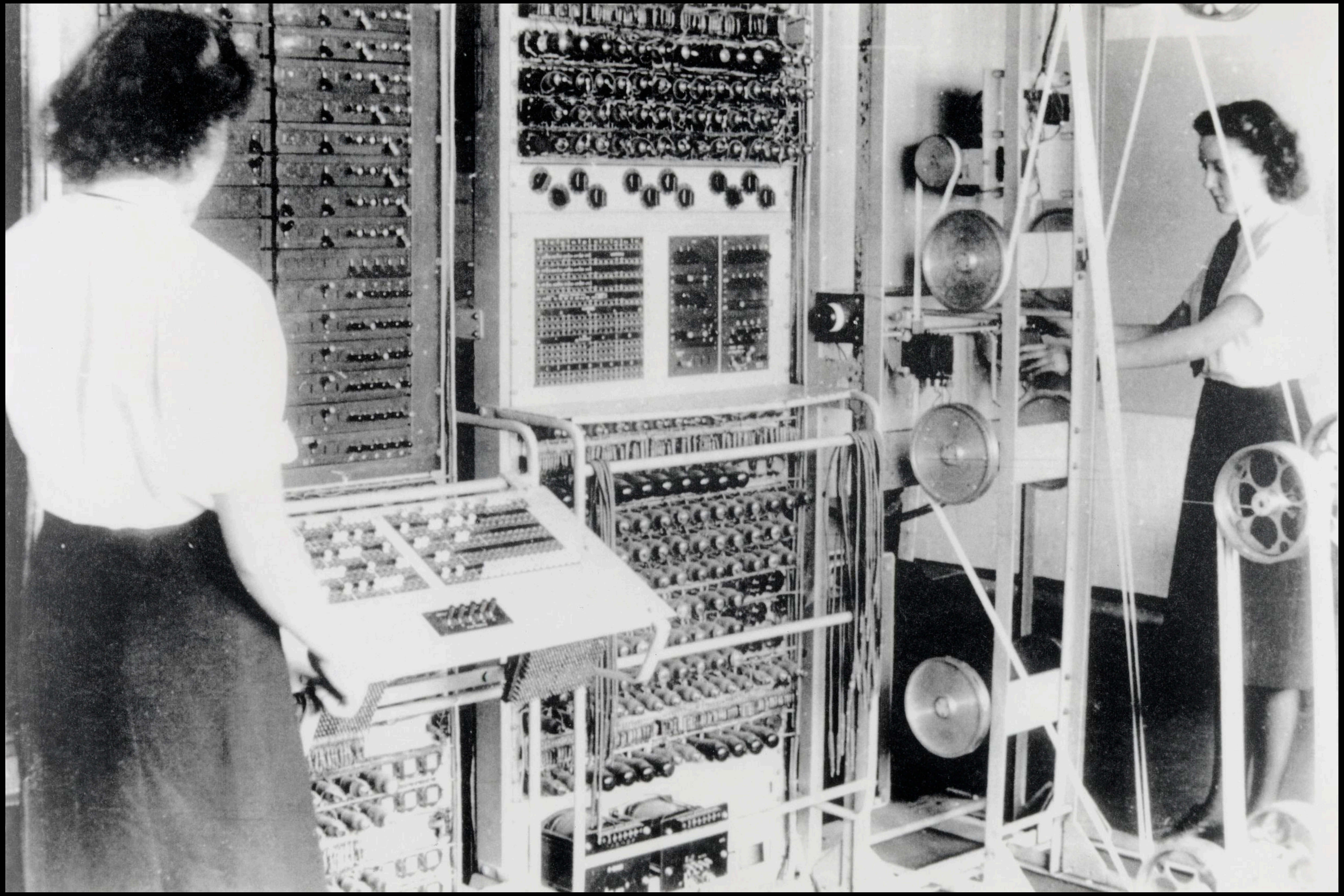


Image courtesy wikipedia

My first Mac (1984)

- First Macintosh
- 1 x 10^6 FLOPS

Image courtesy wikipedia

My Mac in 2003

- 2 cores
- 2×10^9 FLOPS

Image courtesy Apple

My current Mac

- 4 cores
- 2×10^{11} FLOPS

Image courtesy Apple

My current iPhone

- 6 cores
- 8×10^9 FLOPS

Image courtesy Apple

Clicker question #1.4

- In 1 second, today's high-end smart phones can perform ________ calculations per second (FLOPS).

A 10^4 (10 thousand)

B 10^7 (10 million)

C 10^{10} (10 billion)

D 10^{12} (1 trillion)

Images courtesy wikipedia, NASA

Clicker question #1.5

- In 1 second, today's high-end smart phones can perform as many calculations as _______ humans?

A 10^4 (10 thousand)

B 10^7 (10 million)

C 10^{10} (10 billion)

D 10^{12} (1 trillion)

For comparison:

Humans alive in 2018: 7.6×10^9
Total humans who ever lived: 10^{11}
Sources: google.com, pro.org

Clicker question #1.3

- Today's most powerful computers are __________ times more powerful than *today's high-end personal computers*.

A	10 (ten)
B	10^3 (a thousand)
C	10^6 (a million)
D	10^9 (a billion)

Images courtesy wikipedia, NASA

Ocean supercomputer at Cal State Fullerton

- Supercomputer for Cal State Fullerton Gravitational-Wave Physics and Astronomy Center
- 824 cores
- $\approx 10^{12}$ - 10^{13} FLOPS

Blue Waters

- Most powerful computer I have used
- 70,000 cores
- 1×10^{16} FLOPS

Image courtesy Blue Waters

Summit

- Most powerful computer in the world (June 2019)
- 200,000 cores
- 2 x 10^{17} FLOPS
- Record with graphics cards: 2 x 10^{18} FLOPS

Image courtesy Oak Ridge National Laboratory

High performance computing

- Computing beyond what personal devices can do
  - Many cores work together in parallel

FLOPS	Example	Computing Type
10^{0}	*Addition by human with pen & paper*	*Early*
10^{3}	*Room-sized computer in 1940s*	
10^{6}	1980s personal computers (1984)	Personal
10^{9}	Personal computers around year 2000	
10^{10}	High-end smartphone today	
10^{11}	High-end PC today	
10^{12}	**Small supercomputer today**	**High-Performance**
10^{16}	**Most powerful supercomputer I've**	
10^{17}	**Most powerful supercomputers today**	

Clicker question #1.3

- Today's most powerful computers are __________ times more powerful than *today's high-end personal computers*.

A. 10 (ten)

B. 10^3 (a thousand)

C. 10^6 (a million)

D. 10^9 (a billion)

Images courtesy wikipedia, NASA

Clicker question #1.6

- In 1 second, the most powerful computer in the world can perform as many calculations as _______ humans?

A 10^8 (100 million)

B 10^{11} (100 billion)

C 10^{14} (100 trillion)

D 10^{17} (100 quadrilliion)

For comparison:

Humans alive in 2018: 7.6 x 10^9

Total humans who ever lived: 10^{11}

Sources: google.com, pro.org

Clicker question #1.7

- In 1 second, a small supercomputer like ORCA can perform as many calculations as _______ humans?

A 10^6 (1 million)

B 10^9 (1 billion)

C 10^{12} (1 trillion)

D 10^{15} (1 quadrilliion)

For comparison:

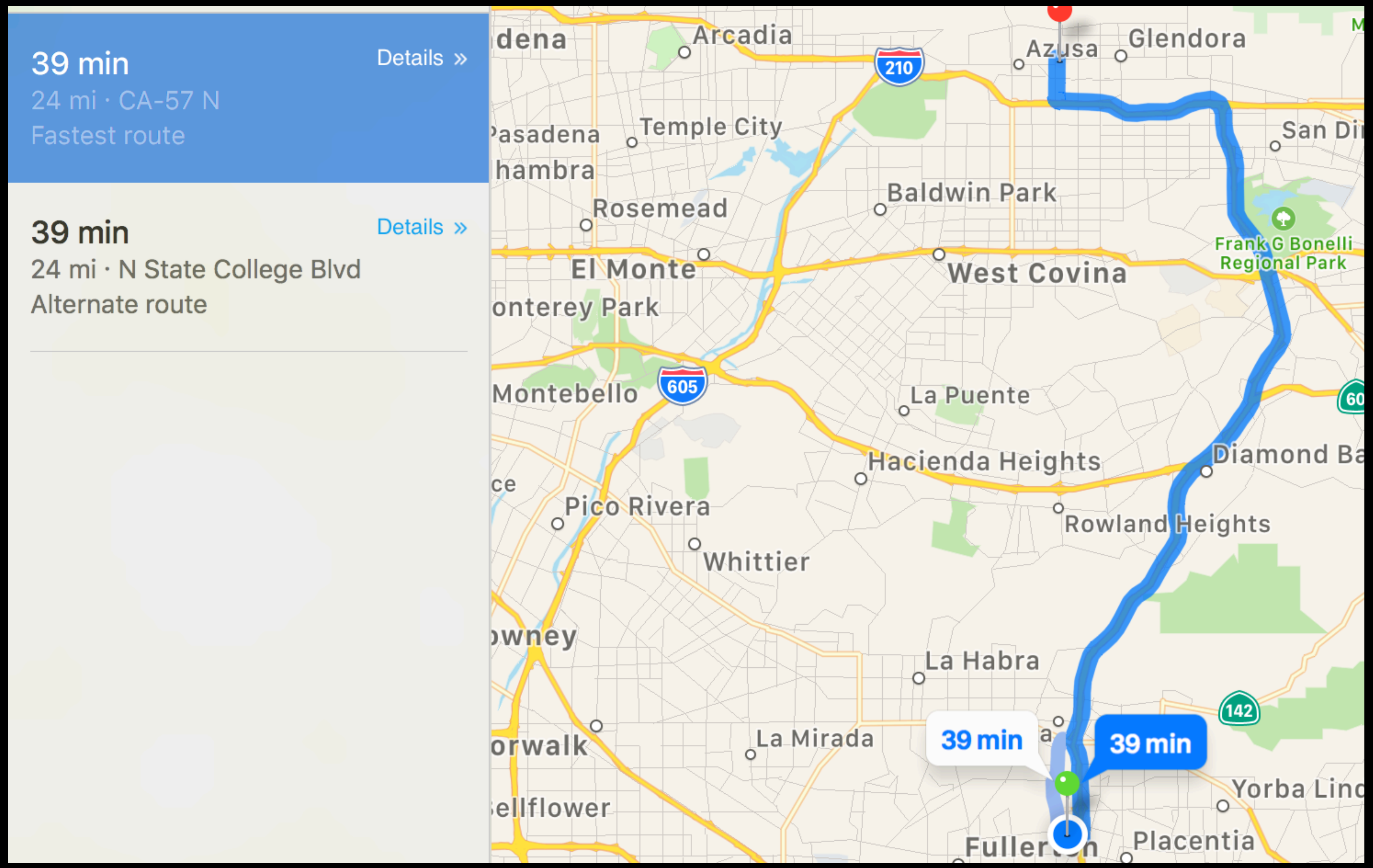
Humans alive in 2018: 7.6 x 10^9

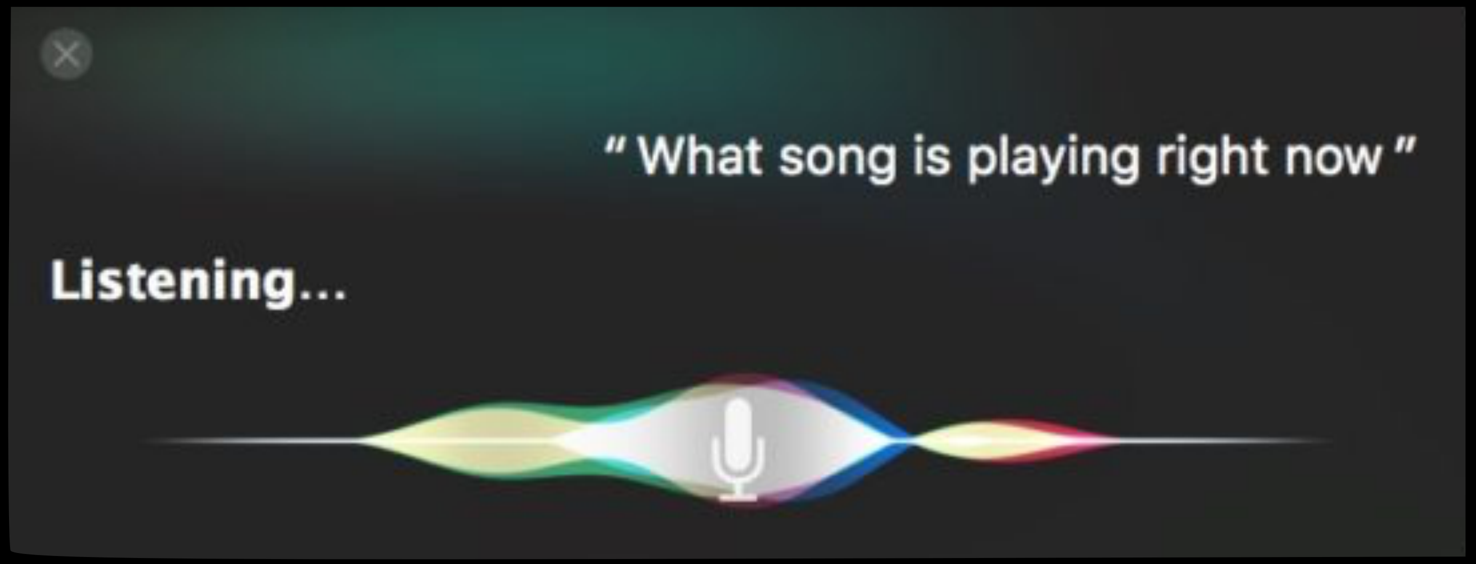
Total humans who ever lived: 10^{11}

Sources: google.com, pro.org

High-performance computing in everyday life

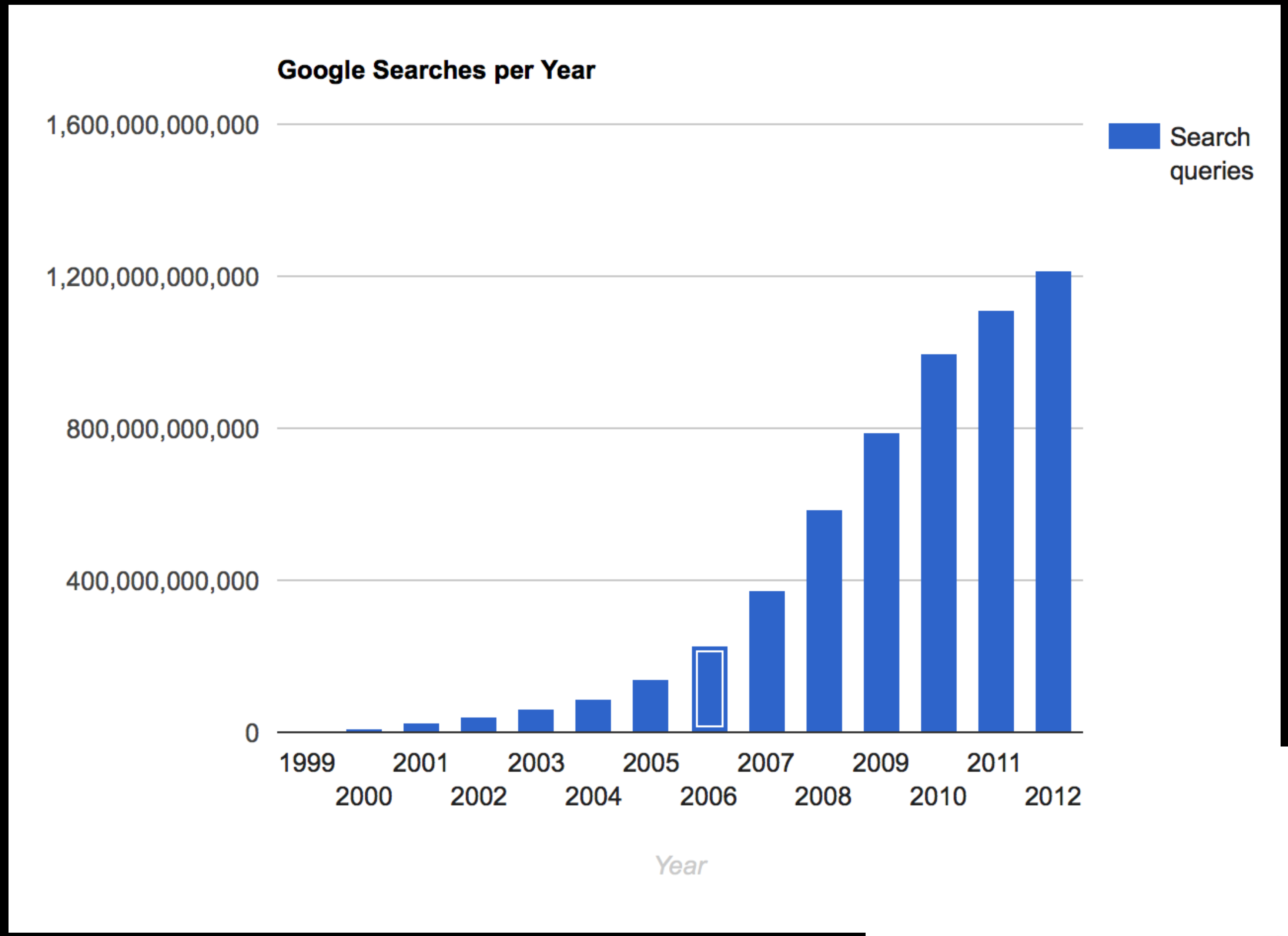
- Cloud computing
  - Search the web
  - Identify a song
  - Get directions
  - Voice assistants
  - Speech recognition





Example: Google search

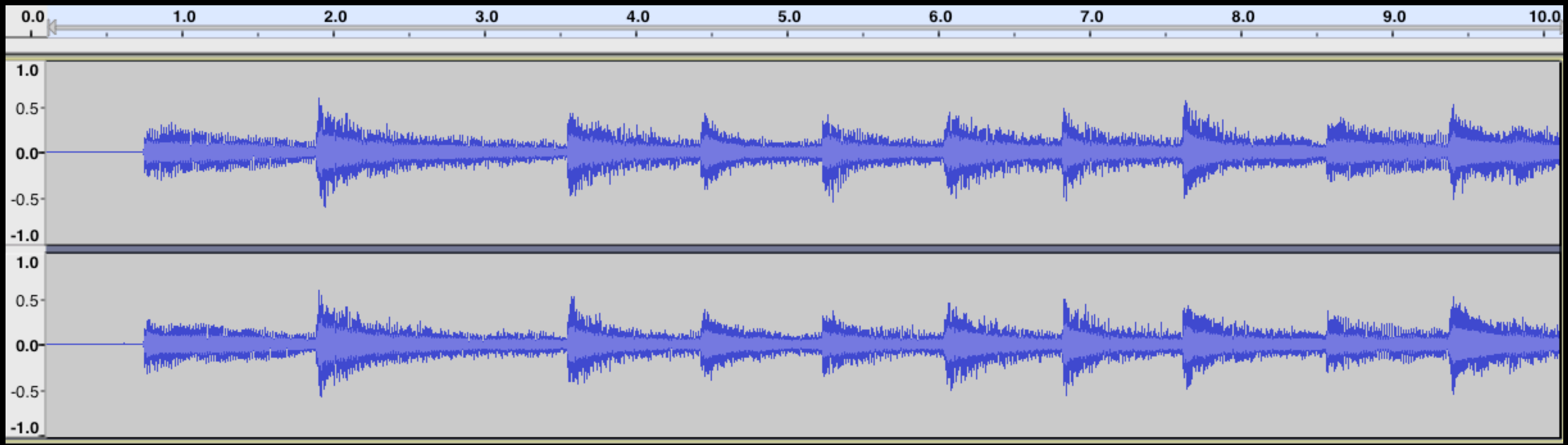
- Search ~ 10^{13} web pages
- 10^3 "servers" per query
- Each query takes about 0.2 seconds
- 4×10^4 queries on average every second of every day
  - If each server is "only" 10^9 FLOPS, Google search requires about 10^{16} FLOPS

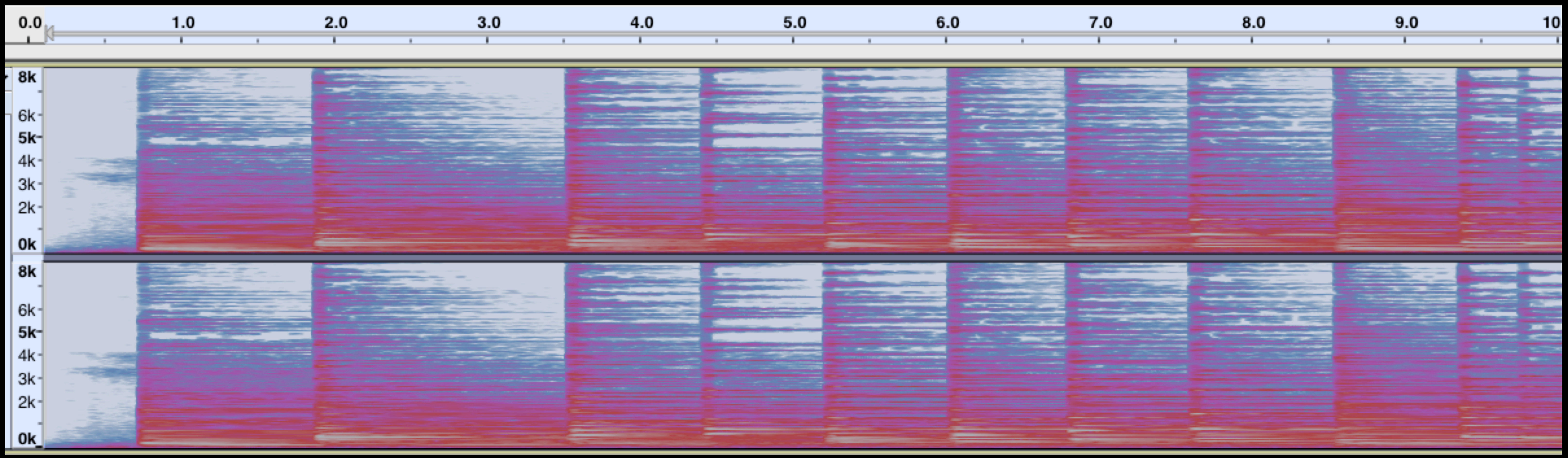


Images courtesy Google, internetlivestats.com

Example: Shazam

- 200 queries on average every second of every day
  - Convert sound into time-frequency plots, filter to keep only the loudest notes
  - Compare to a large library
  - Similar to how LIGO searches data for gravitational waves!
  - One query is a PC-sized calculation, roughly





Amazon web services data center
Courtesy amazon.com

Microsoft Azure data center
(courtesy sensorslab.co)

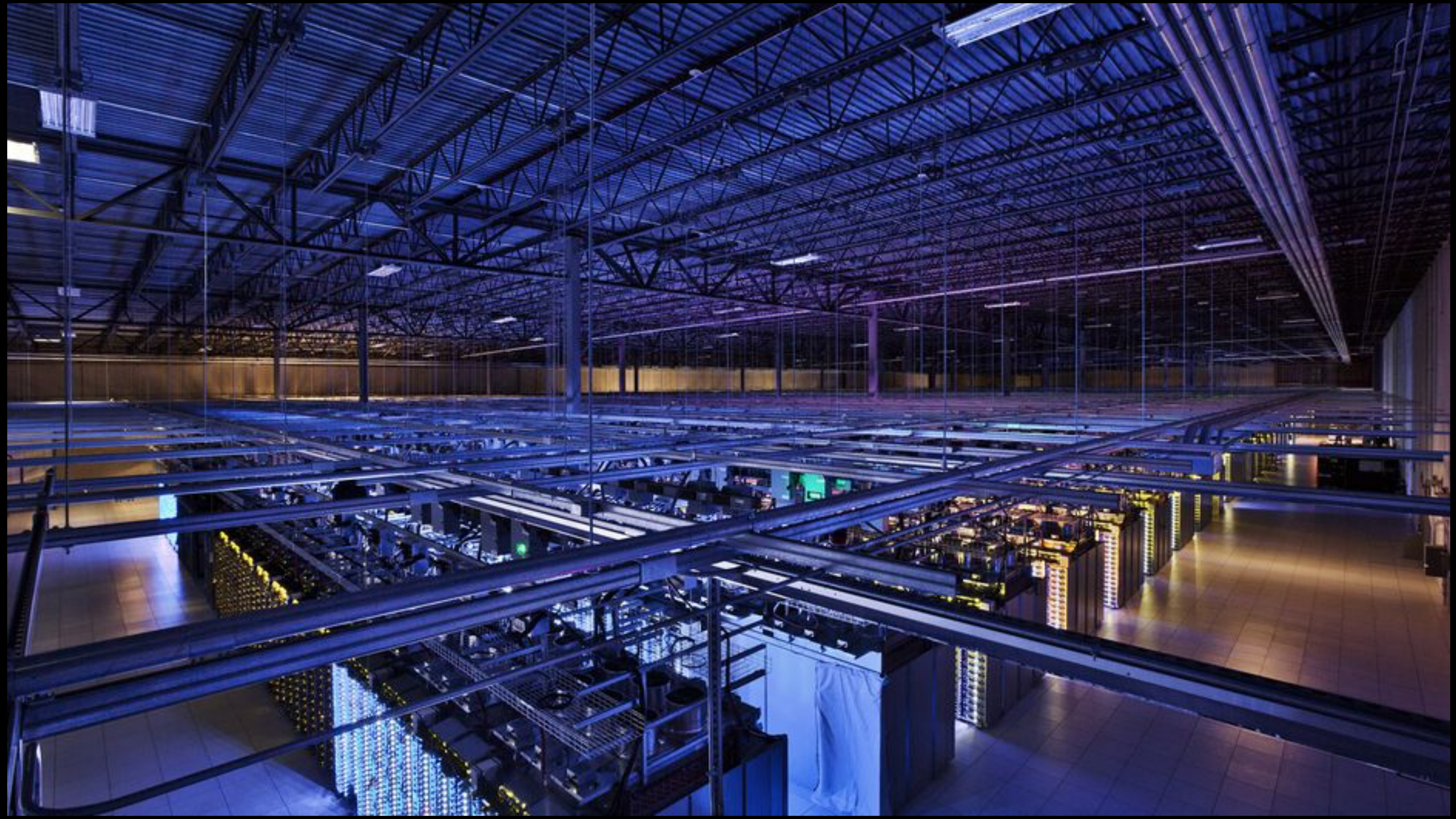


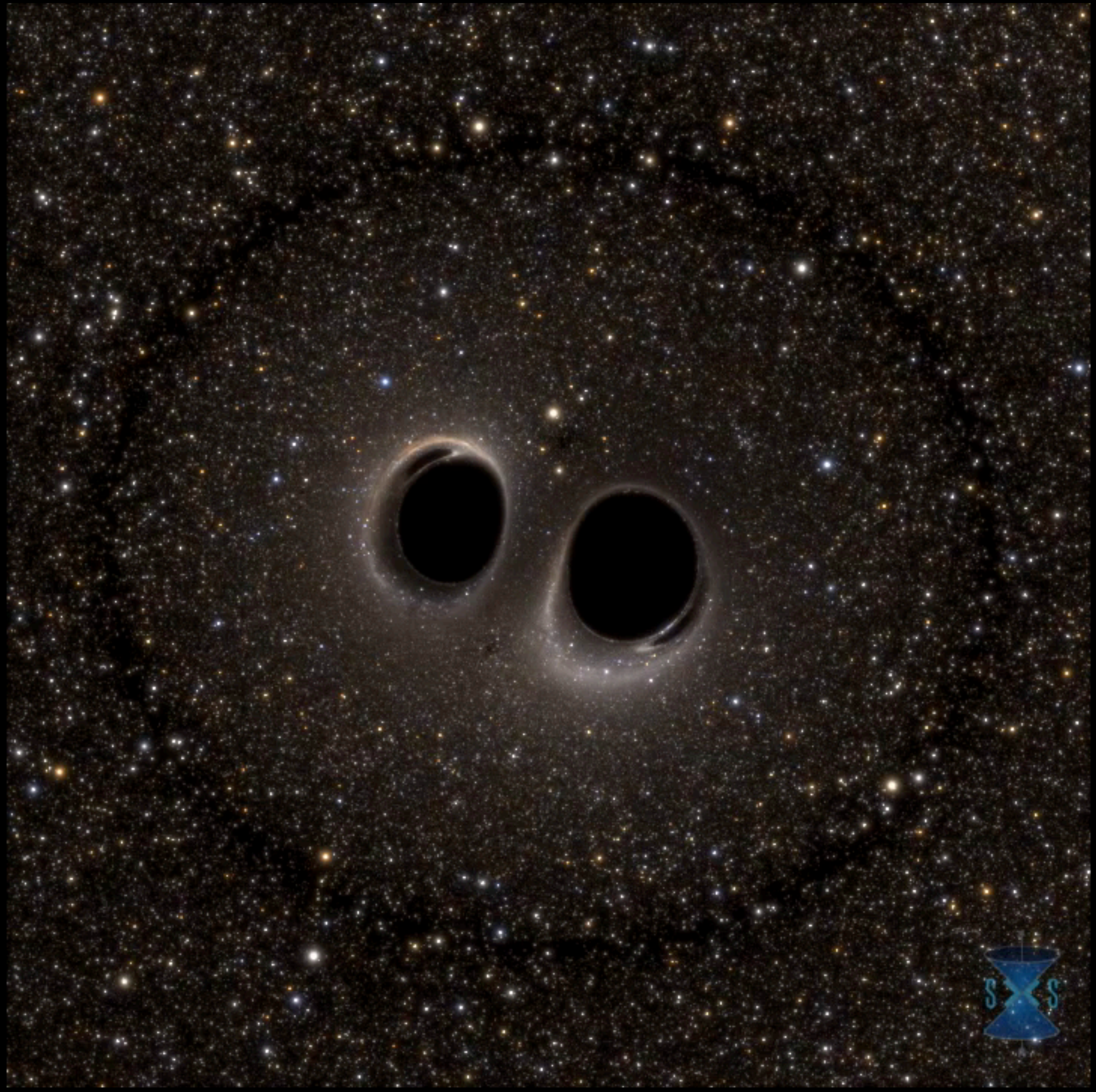
Image courtesy cnet: Google data center, Council Bluffs, Iowa

Google: 60,000 searches/second

Provide many 10^{15} FLOPS
of performance to customers

High-performance computing for science

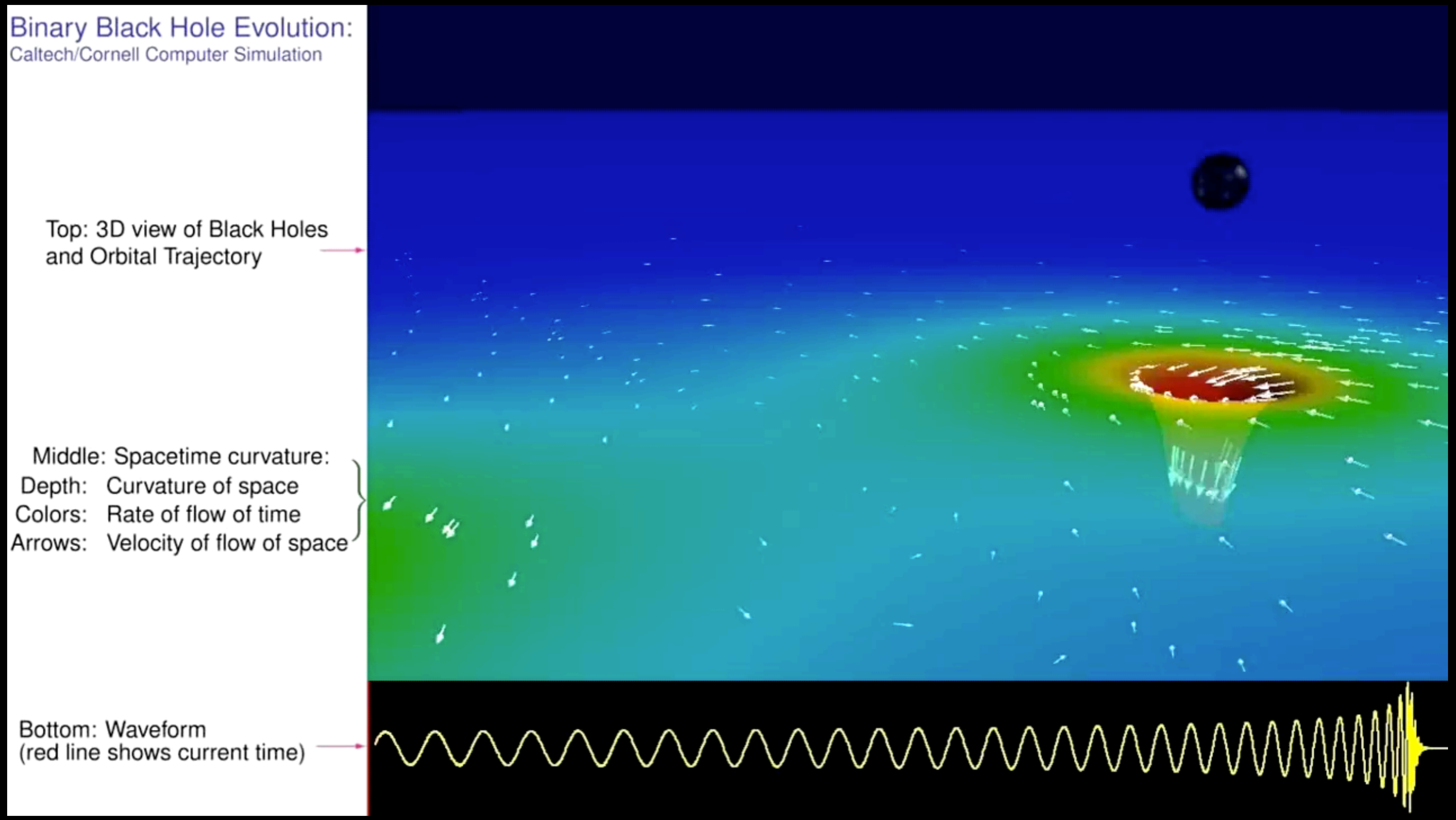
- Solve otherwise unsolvable problems
- Insight into scientific data & results
  - Experimental measurements
  - Results of calculations
  - Complicated pencil & paper results



Movie & calculation by undergraduate
Haroon Khan, Nick Demos,
Simulating eXtreme Spacetimes collaboration

Example: Simulating colliding black holes

- Head-on collision example
  - About 1.4 x 10^{17} floating-point operations
  - 3 days (48 cores on ocean)
  - A month on a typical laptop
- Inspiral, merger, ringdown example
  - About 7.8 x 10^{17} floating-point operations
  - 17 days (48 cores before merger, 36 cores after, on ocean)
  - Months on a typical laptop



Programming with Python

Programming is like magic

- Say the right cryptic words and something cool happens
- Mess up a word and the spell fizzles


Images courtesy Warner Bros.

pythontutor.com

- Helps you visualize what the program is doing
- Simplified version that we will use: http://cs1110.cs.cornell.edu/tutor/#mode=edit

Google colaboratory

- https://colab.research.google.com
- Google lets us program and run on their computers for free

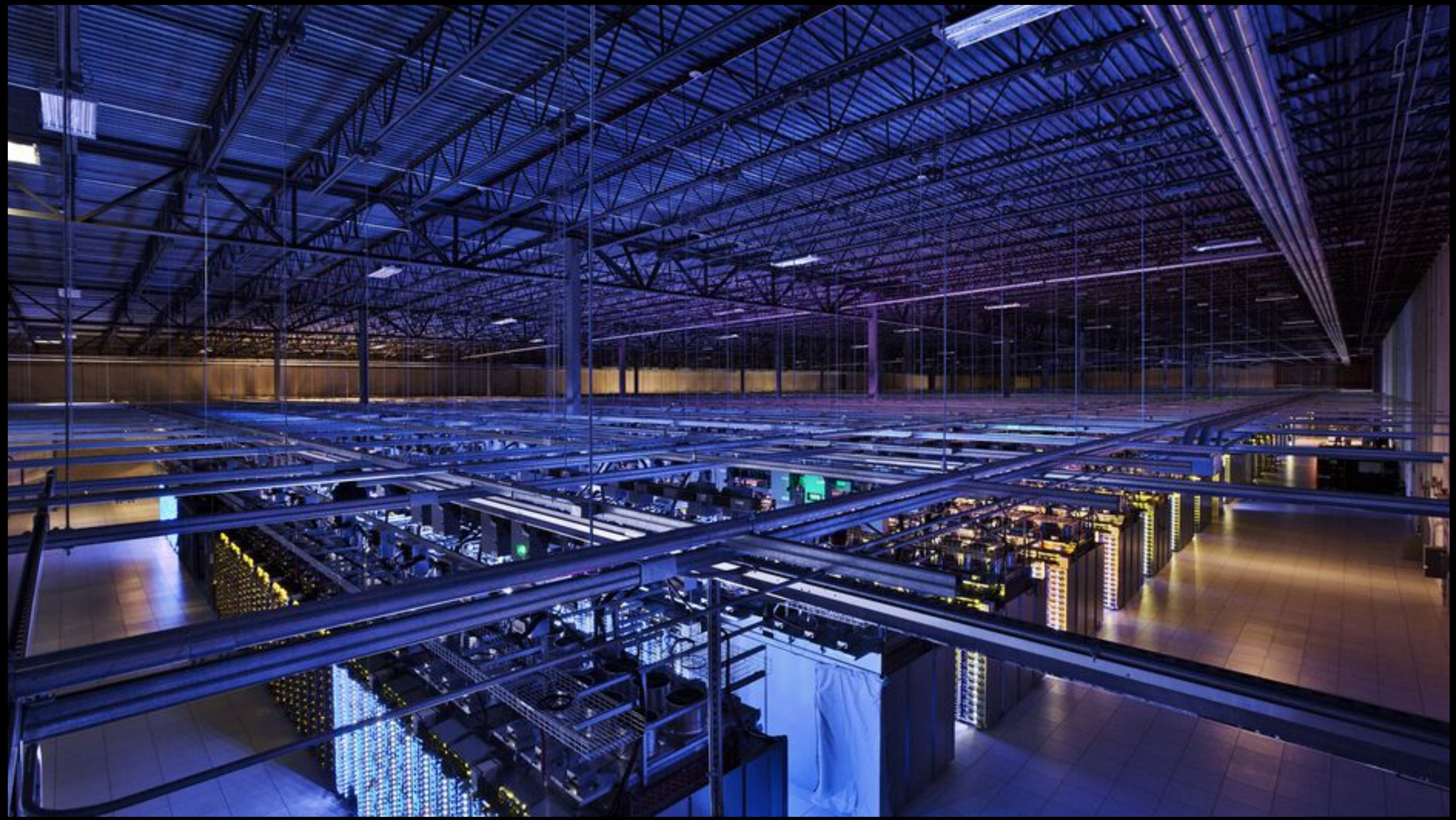


Image courtesy cnet: Google data center, Council Bluffs, Iowa

How to play along

- Open python tutor in a new browser tab in Chrome: http://cs1110.cs.cornell.edu/visualizer/
  - We'll use the tutor to see "inside" what the code is doing, step by step
- Open colab.research.google.com in another tab, and make a new Python3 notebook
  - Save the notebook on your google drive, rename to "YOURNAME_Workshop2018.ipynb"
  - Share the notebook with me (geoffrey4444@gmail.com)
  - We'll use notebook to actually run stuff "for real"
- If you get an error, let me know!

Output

- Your program needs to tell you the result
- Tradition since 1974: first program prints "Hello world"
- Python (language commonly used in scientific computing) makes this easy

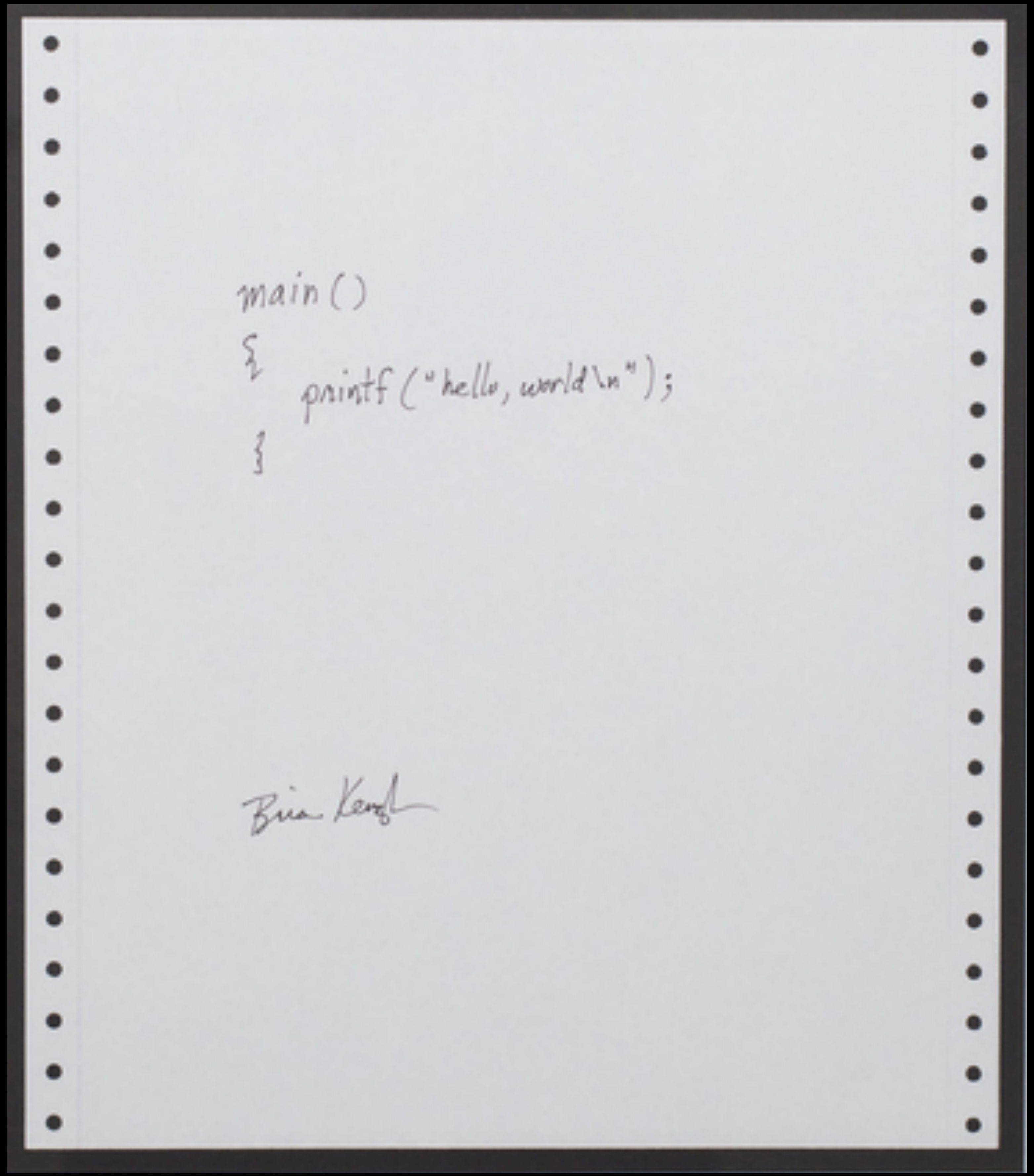
Try in tutor:

```python
print("Hello, world!")
```

- Print basically anything

Try in tutor:

```python
print(4*4+4-4)
```



Brian Kernighan (early UNIX developer), 1978

Libraries

- Don't reinvent the wheel when you want to hit the road
    - (But OK if you want to learn how to make wheels)
- Python has *many* libraries for numerical computing & everything else
- By "Libraries", I mean any pre-written code that you can use in your programs

Try in tutor:

```
import math
print(math.pi)
```

Math

Try in tutor (only type the left hand side of the ==):

- Arithmetic operations built in

```
(4 + 4) * 4 / 4 - 4
```

- Exponents with **

```
4 ** 4 == 256
```

- Scientific notation

```
4e4 == 40000
```

- The rest in the math library

```
math.sin(4)
math.sqrt(4)
```

Expressions

- Value = piece of data of a particular type — `4.444` — `“Hello world”`
- Type = kind of data — `float` — `string`
- Operator = combine values to get a new value — `+ - * /` — `+`
    - Behavior depends on type
- Expression = group of values and operators — `4.0 * 3.0 - 2.0` — `"Hello" + " world"`
- Python evaluates expressions, like a calculator

Clicker question #2.0

- What does Python get when it evaluates this expression? `4.0 * 3.0 - 2.0`

A	4.0
B	10.0
C	Some other number
D	An error

Try out some expressions in the tutor

```
4.0 * 3.0 - 2.0
```

```
"Hello" + " world"
```

Try out some expressions in the tutor

```
print(4.0 * 3.0 - 2.0)
```

```
print("Hello" + " world")
```

```
#make up your own
```

Some types we will need

- Float
- Int
- String
- Boolean

Type: float

- **Values**: real numbers ("numbers with decimal points")
  - Examples `4.1234` `4.0` `4.4e2` `-5.2e-3`
  - If you don't include a decimal point, it is an integer!
- **Operators:** `+ - * / **`

Try in tutor:

```
print(22.0 / 7.0)
```

```
print(-3.0e-3 * 10.0)
```

```
print(8.0**2.0)
```

```
print(1.0/3.0)
```

```
print(type(4))
print(type(4.0))
```

Type: int

- **Values**: integers (whole numbers, positive, negative, zero)
  - Examples `-4` `742352046` `7` `-33`
    - Don't use commas when typing an int or float
- **Operators**: `+ - * **` `/` `//` `%`

Try in tutor:

```
print(2**8)

print(4 * 3 - 2)
```

```
print(7 / 3) #float in Python3,
             #int in Python2 (avoid!)

print(7 // 3) # quotient
print(7 % 3)  # remainder
```

Clicker question #2.1

- In Python 3, what is the value of this expression?
`10 // 3 + 1`

A	4
B	4.3333333333
C	Some other number
D	An error

Type: boolean

- **Values**: true or false
  - Examples `True` `False`

- **Operators**: `and or not`
  - `a and b` is true if both are true, false otherwise
  - `a or b` is true if a is true, b is true, or both are true
    is false if both a and b are false
  - `not a` is true if a is false, false if a is true

= and ==

- = stores results in a named object ("variable")

```
myNumber = 4
print(myNumber * myNumber)
```

```
print(myNumber * myNumber == 16)
True
```

- == tests whether two objects are equal

```
print(2 + 2 == 5)
False
```

Try in the tutor

- = stores results in a named object ("variable")
- == tests whether two objects are equal

```
print(2 + 2 == 4 and 3 + 3 == 6)
print(2 + 2 == 4 and 3 + 3 == 7)
print(2 + 2 == 4 or 3 + 3 == 7)
print(not 3 + 3 == 7)
```

```
a = True
b = True
c = False
d = False
```

```
# Pick a few of these
print(a)
print(not c)
print(not a)
print(a or b)
print(a or c)
print(c or d)
print(a and b)
print(a and c)
print(c and d)
```

Converting types

Try in tutor:

```
q = 4
print("The number is "+q)
```

```
q = 4
print("The number is "+str(q))
```

```
print(type(4))
print(type(str(4)))
print(type(float(4)))
```

Clicker question #2.2

- What does this line print?

```python
import math
print("The value of pi is "+math.pi)
```

A The value of pi is 3.141592653589793

B The value of pi is math.pi

C Something else but not an error

D An error

Clicker question #2.2

- What does this line print?

```
import math
print("The value of pi is "+str(math.pi))
```

A The value of pi is 3.141592653589793

B The value of pi is math.pi

C Something else but not an error

D An error

Comments

- Comments explain what you're doing
- Use comments to explain your code
- Use names that help explain, even without comments

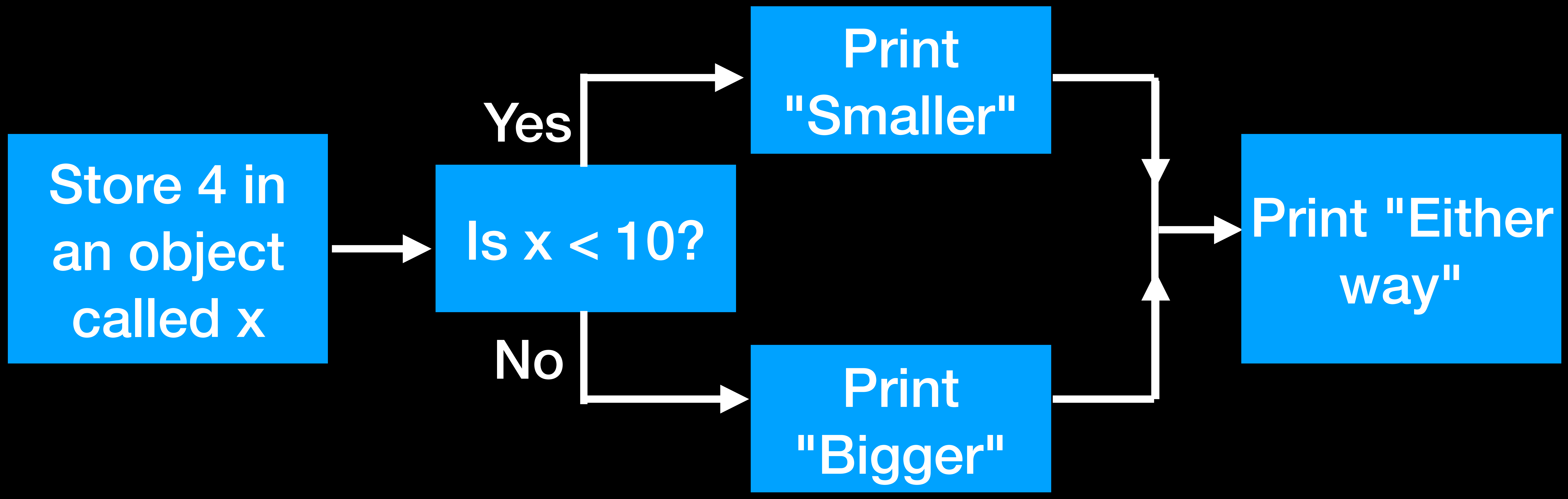
```
# Say hello to someone by name
personName = "Geoffrey"
print("Hello " + personName)
```

If/else

- If does the first indented thing if the stuff in () is True
- Otherwise it does the indented stuff under "else"

```
x = 4
if(x < 10):
    print("Smaller")
else:
    print("Bigger")
print("Either way.")
```

Try in tutor!



Clicker question #2.2b

- What does this program print?

```
x = 4
if x==10 or x==11:
    print('yes')
else:
    print('no')
```

A Yes

B No

C The code gives an error

Clicker question #2.2

- What does this program print?

```
x = 4
if x==10 or 11:
    print('yes')
else:
    print('no')
```

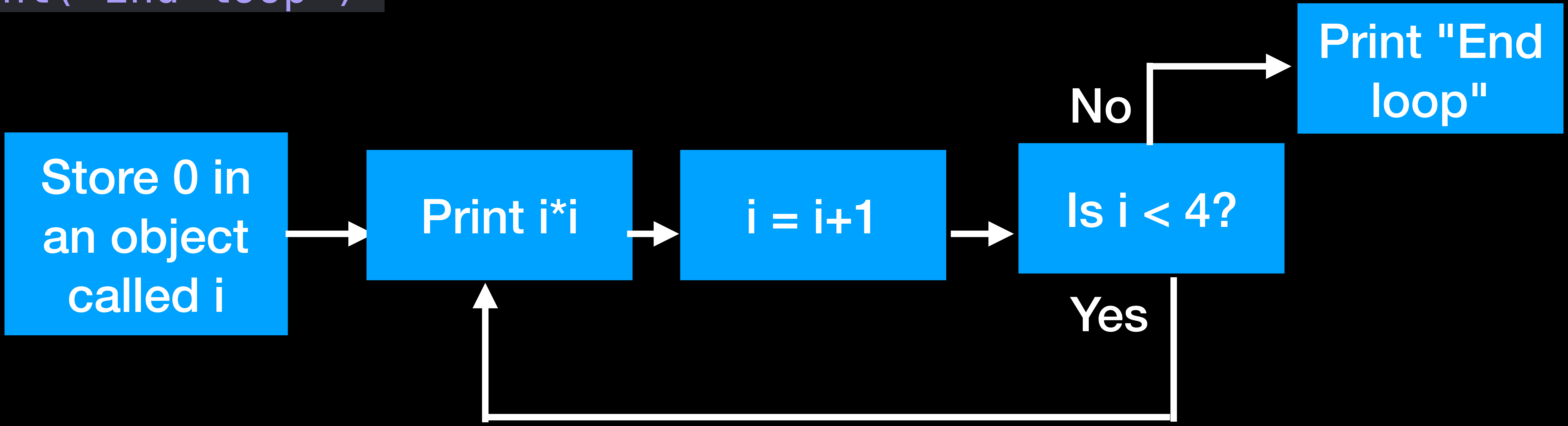
A	Yes
B	No
C	The code gives an error

Try in tutor!

```
i = 0
while i < 4:
  print(i*i)
  i = i + 1
print("End loop")
```

```
0
1
4
9
```

Loops



Loops

```
for i in [1,2,3,4]:
  print(i*i)
```

```
0
1
4
9
```

```
i = 0
while i < 4:
  print(i*i)
  i = i + 1
```

```
0
1
4
9
```

So far, our programs just run & stop…
How do programs with a user interface work?

Clicker question #2.3

- What does this program print?

```
j = 1
while j < 3:
    j = j + 1
print(j)
```

A	1
B	2
C	3
D	4

Clicker question #2.4

- What does this program print?

```
product = 1
j = 1
while j < 3:
    product = product * j
    j = j + 1
print(product)
```

A	1
B	2
C	6
D	24

Clicker question #2.4b

- What does this program print?

```
product = 1
j = 1
while j < 4:
    product = product * j
    j = j + 1
print(product)
```

A	1
B	2
C	6
D	24

Clicker question #2.4c

- What value of x makes the program print 24?

```
product = 1
j = 1
while j < x:
    product = product * j
    j = j + 1
print(product)
```

A 3

B 4

C 5

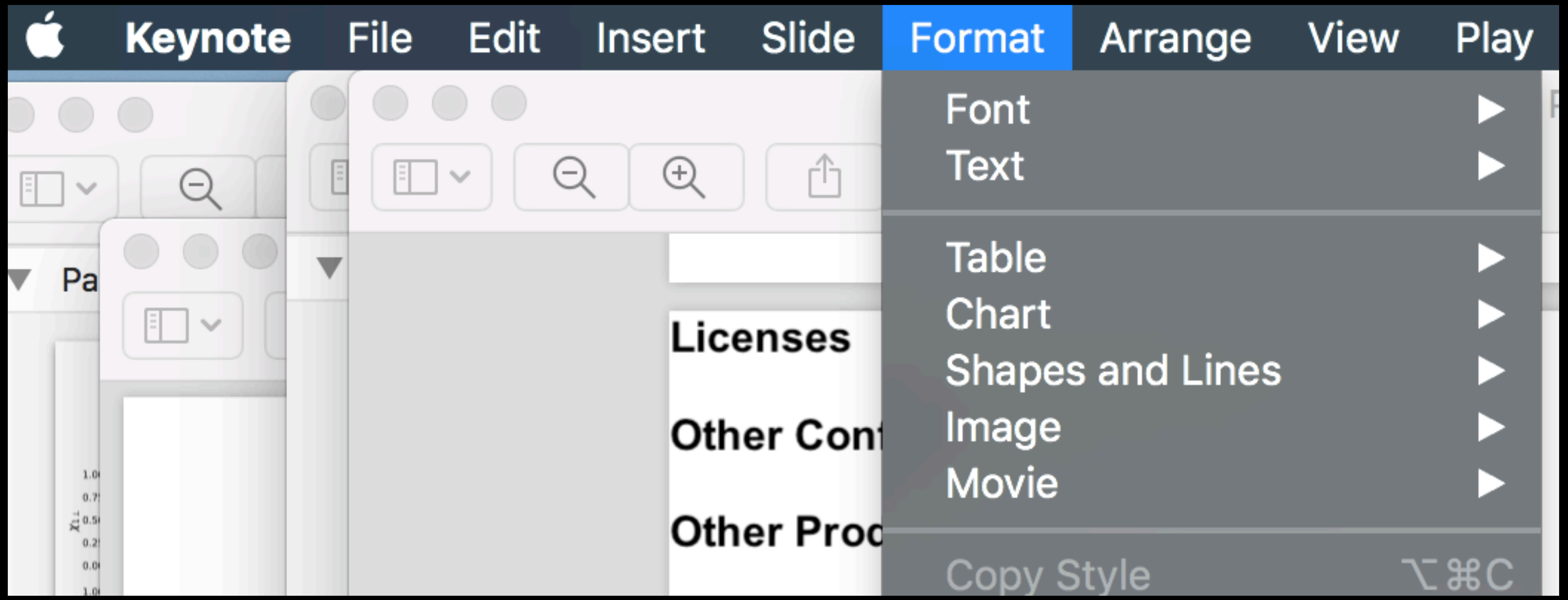
D 6

Loops

- Real life:
  event loop

- Event = key press,
  mouse/trackpad
  click,
  …

```
while message != quit:
  message = get_next_message()
  process_message(message)
```



My first program

- Basic, 1987

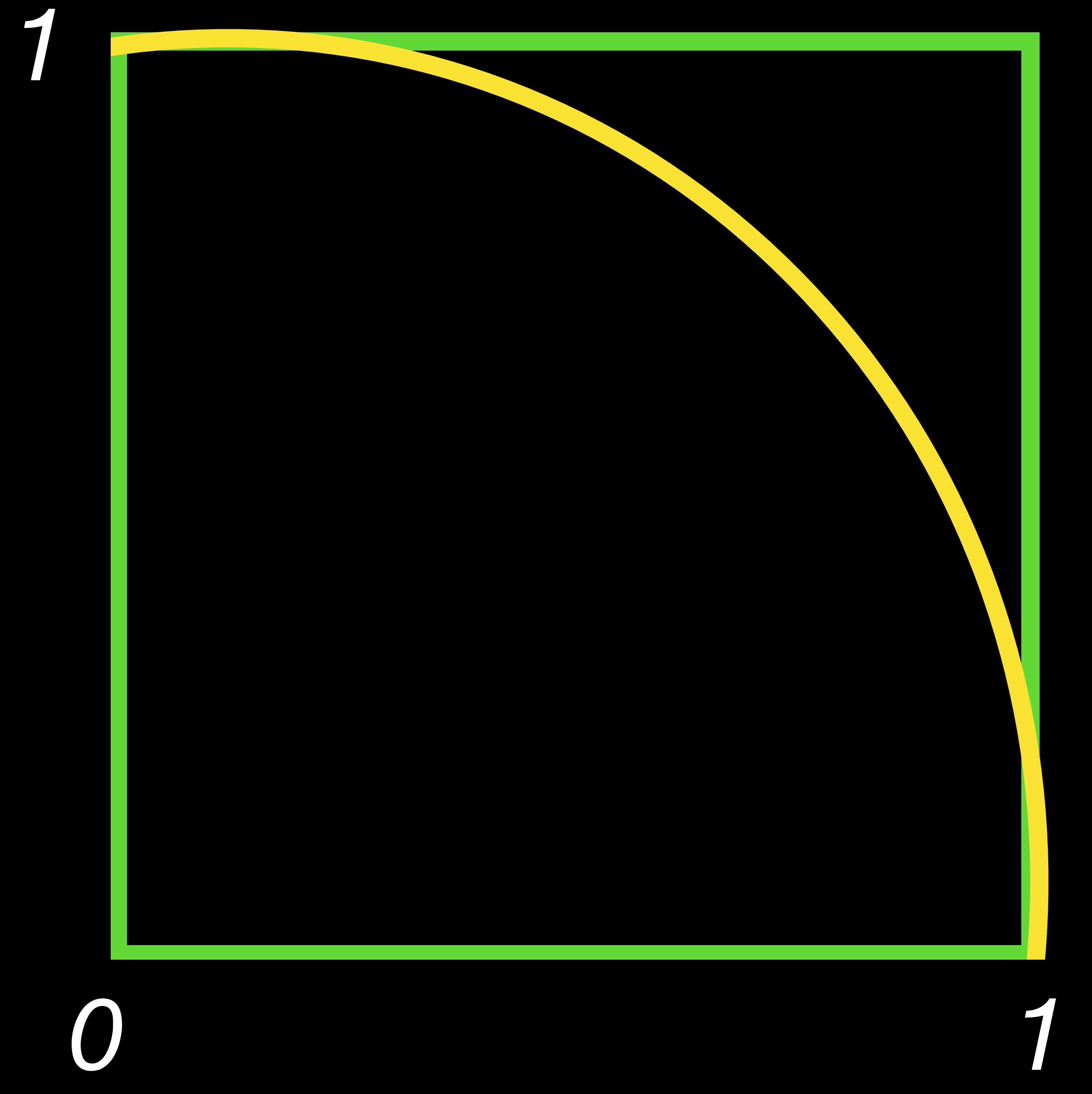
```
10 PRINT "GEOFFREY"
20 GOTO 10
```

- Python equivalent

```
done = False
while not done:
  print("Geoffrey")
```

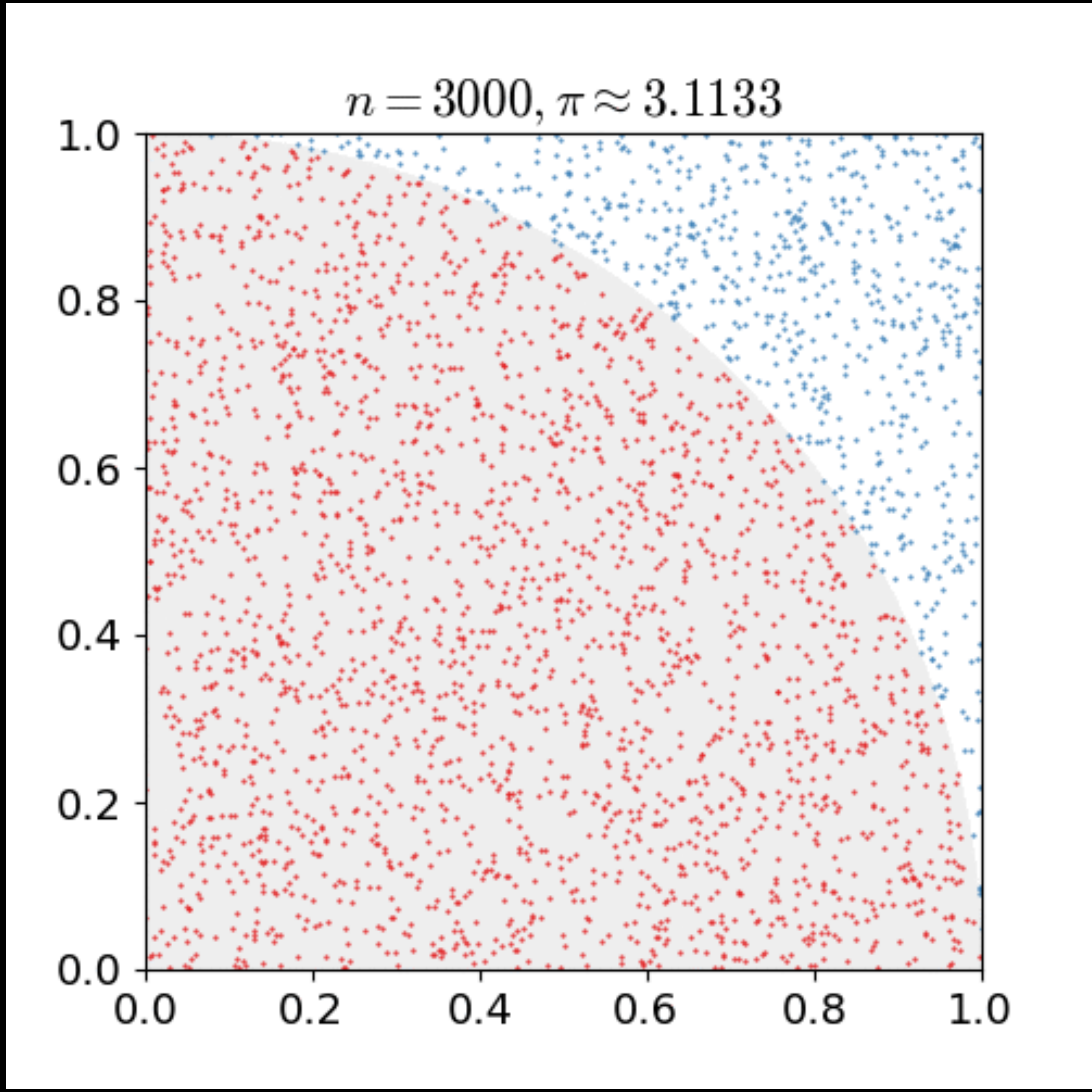
A silly way to compute π

- Area of circle?
- Area of square?
- Idea: throw darts in square
  - (circle area) ÷
    (square area)
    ≈ darts in circle ÷
    darts in square
    = "hits" / ("hits" + "misses")



A silly way to compute π

- Throw darts in square
  - (circle area) ÷ (square area) ≈ darts in circle ÷ darts in square = π/4



Courtesy wikipedia