DATA

Richard Harvey

GRESHAM
COLLEGE

DATA

Richard Harvey

IT Livery Company Professor of Information Technology, Gresham College

GRESHAM

COLLEGE

A

www.prof-richard.org

A MOST PLAINE and easie way for the finding of the Sunnes *Amplitude* and *Azimuth*, and thereby the Variation of the *Compasse*, by *Logarithme*.

Written by W. B.

Also another plaine way for the Azimuth, by the Table of Sines in fiue seuerall Cases.

By *I. T.*

by *Logarithme*, which is the most easiest of all *Arethmaticall* workes and first of the *Amplitude*.

To finde the *Amplitude*.

THe *Amplitude* or bredth of the *Sunnes* rising or setting from the true East or West *point*; is found by *Sines* thus, As the sine of the Complement of the *Latitude*, is to the sine of the *Declination*, so is the *Radius* to the sine of the *Amplitude*;

But in *Logarithme*, you are onely to looke the *Logar*: of the Complement of the *Latitude*, and the *Logar*: of the *Declination*; and substract one from the other, the remainer is the *Logarithme* of the *Amplitude*,

Example.

Data { *Complement, Latitude* — 40. deg. 30'. / *Declination* ——— 20. — 40. } *North* } I demand the *Amplitude*?

Comp: Lat: 40. d. 30'. *Loga*: — 4316323
Declinat: 20. 40'. *Loga*: 10414836

6098513 the *Logar*: of the *Amplitude* 32. deg. 55'. and somewhat more.

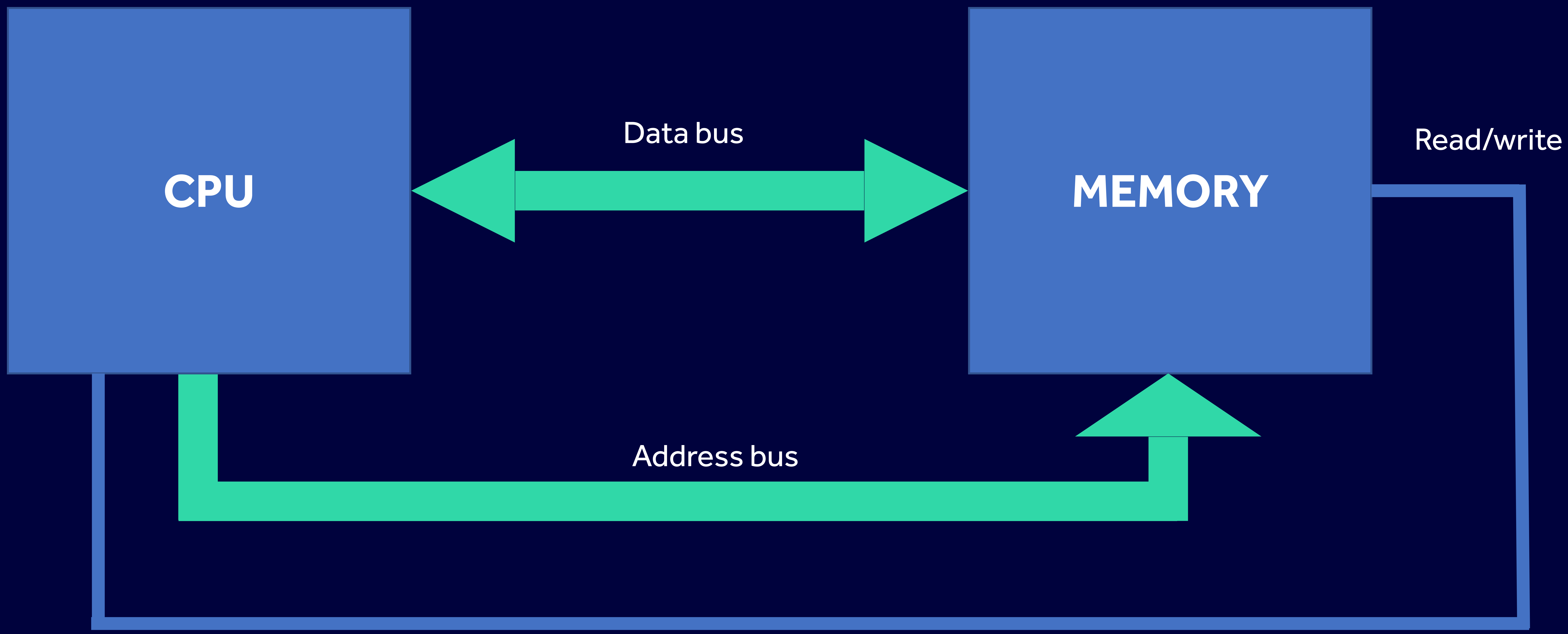
To finde the *Azimuth*.

IT is to be considered that in the Doctrine of *Triangles*, it is required in the solution of any question there are three thinges to be giuen in any *Triangle*, before the question can bee answered, which in this for finding the true *Azimuth* of the Sunne; you are to know or imagine your *Latitude*, the Complement thereof is

 one

Data as digital stuff

How do computers store data?



Data bus width

double precision floating point number	64 bits
floating point number	32 bits
fixed point number	?
long integer	32 bits
short integer	16 bits
character	8 bits

8 bits

Data types

Each computer language gives the programmer access to certain types of data

FORTRAN 77:

```
BYTE (1); CHARACTER (1); INTEGER (4);
REAL (4); DOUBLE (8); COMPLEX (8)
```

Programming standards often specify the size of data which, **usually**, match the sizes available in the machine.

Floating point numbers

Computers, like humans, have to handle numbers with huge scale variations

Idea is to represent numbers in standard form:

sign

mantissa

exponent

base

$$3.14159 = +312159 \times 10^{-5}$$

or in IEEE-754 floating point (base 2):

01000000010010010000111111010000

Floating point numbers

We cannot represent all numbers equally accurately in floating point.

3.14159 -> 01000000010010010000111111010000 ->
3.14159011840820312

Error = 3.141590118408203125 – 3.14159 = $1.18408203125 \times 10^{-7}$

Small!

But small errors can add up

Numerical analysis

We want to add 10.0 to 0.1 using a very simple floating point computer with only one digit for the mantissa

$x = 10.0 = 1\times10^{1}$

$y = 0.1 = 1\times10^{-1}$

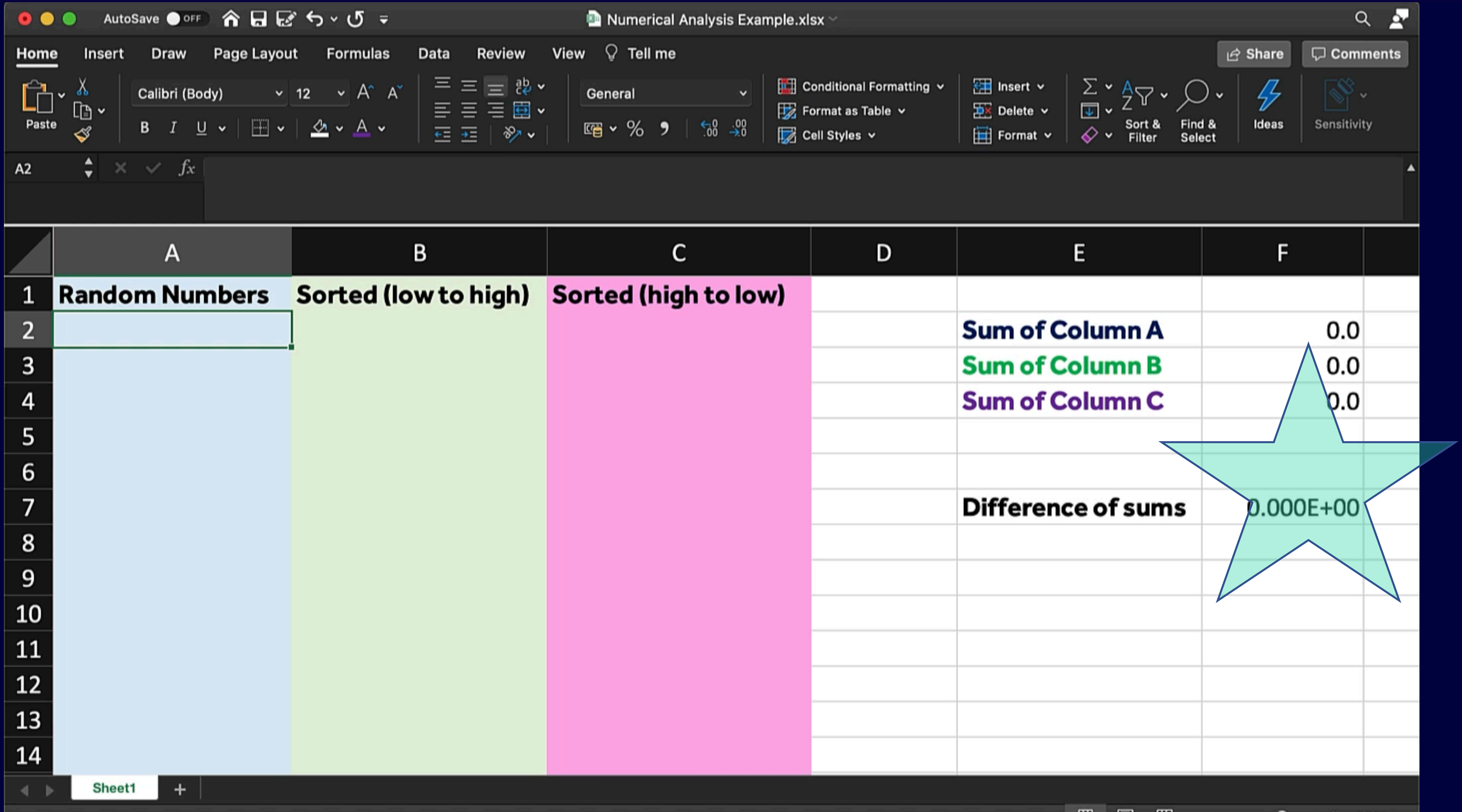
$x + y = 10.1 = 1\times10^{1} = x$

$y + y + y + y + y + y + y + y + y + x= 11.0$

but

$x + y + y + y + y + y + y + y + y + y = 10.0$

Numerical analysis with Excel



Data structures

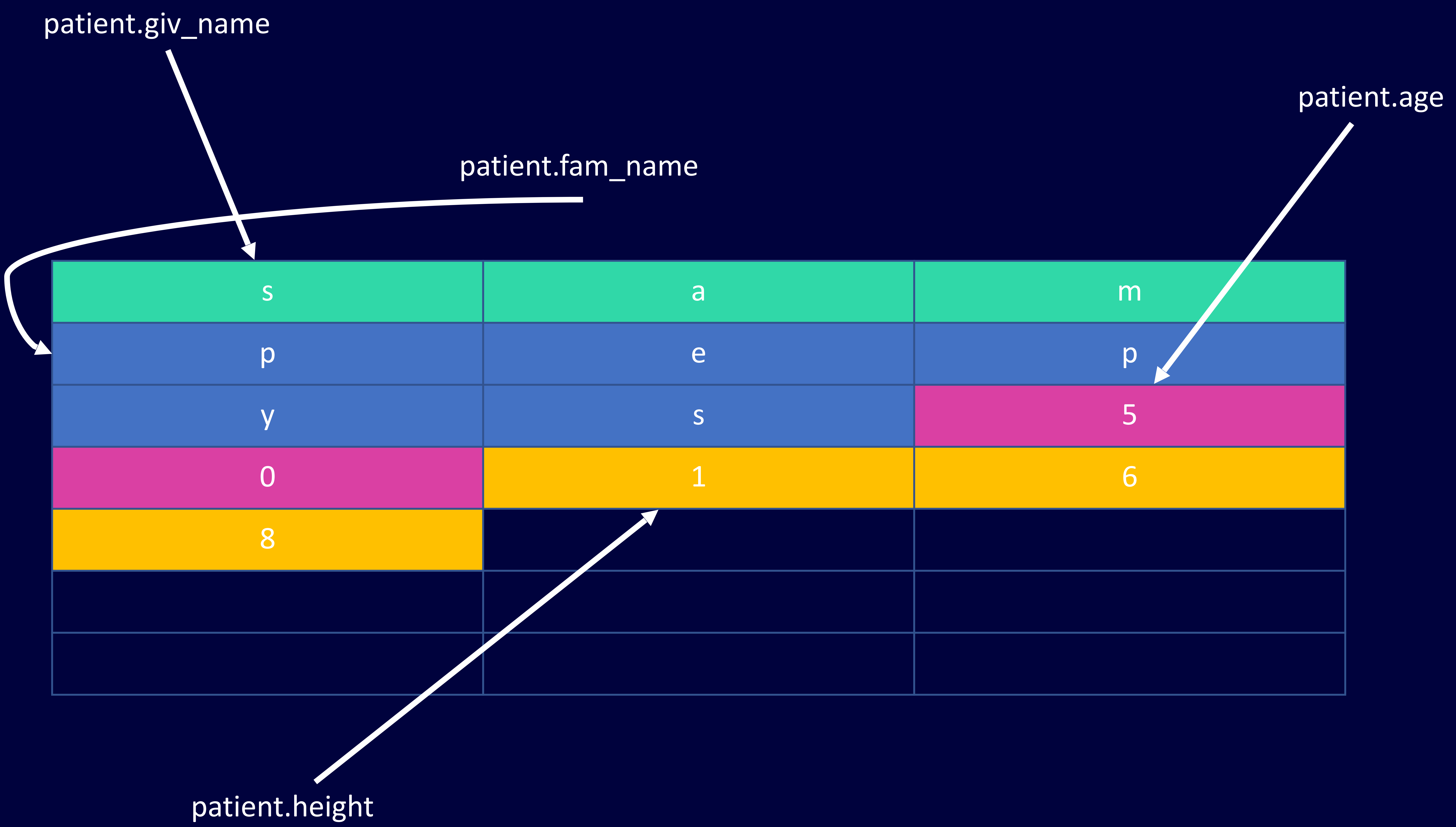
Collections of data that belong together:

arrays

strings

mixed structures

```
     patient - > {string: giv_name
                  string: fam_name
                  integer: age
                  real: height}
```


patient.giv_name
patient.age
patient.fam_name
s
a
m
p
e
p
y
s
5
0
1
6
8
patient.height

Data structures

Collections of data that make computation easier

linked lists

heaps

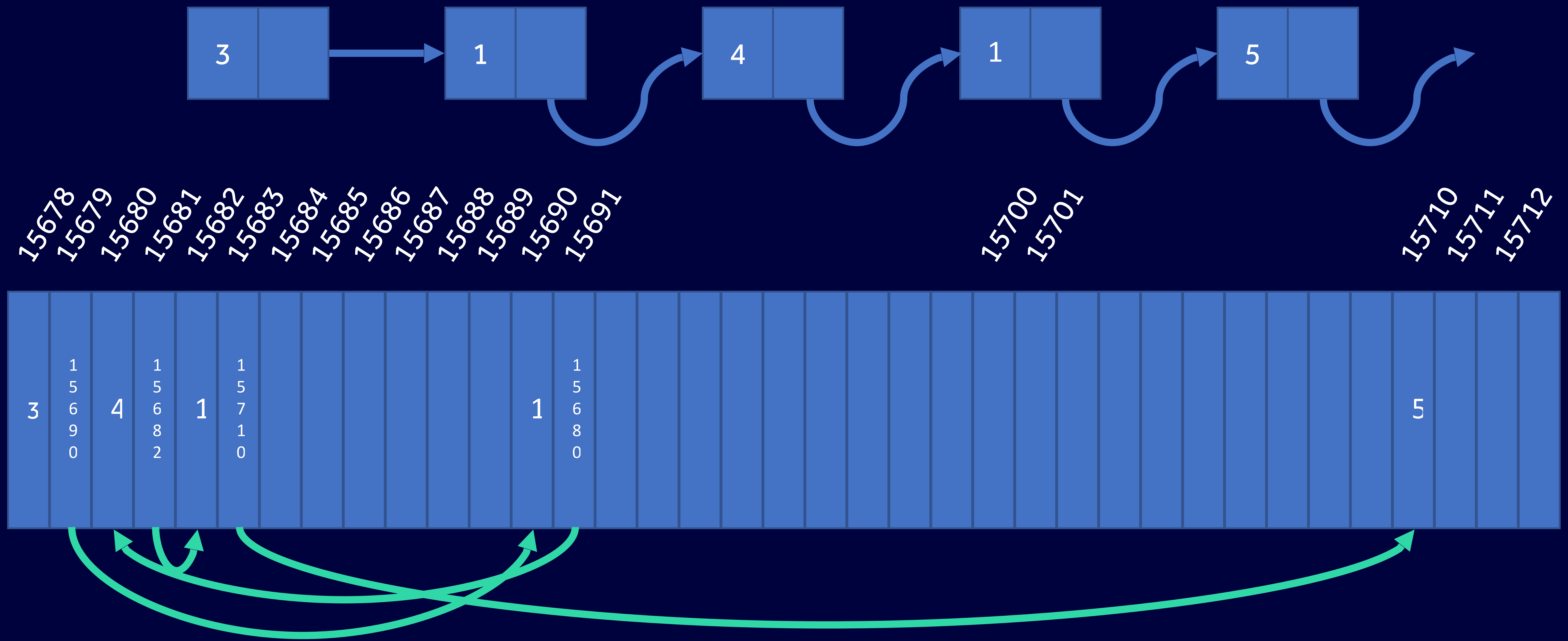
stacks

trees

tries

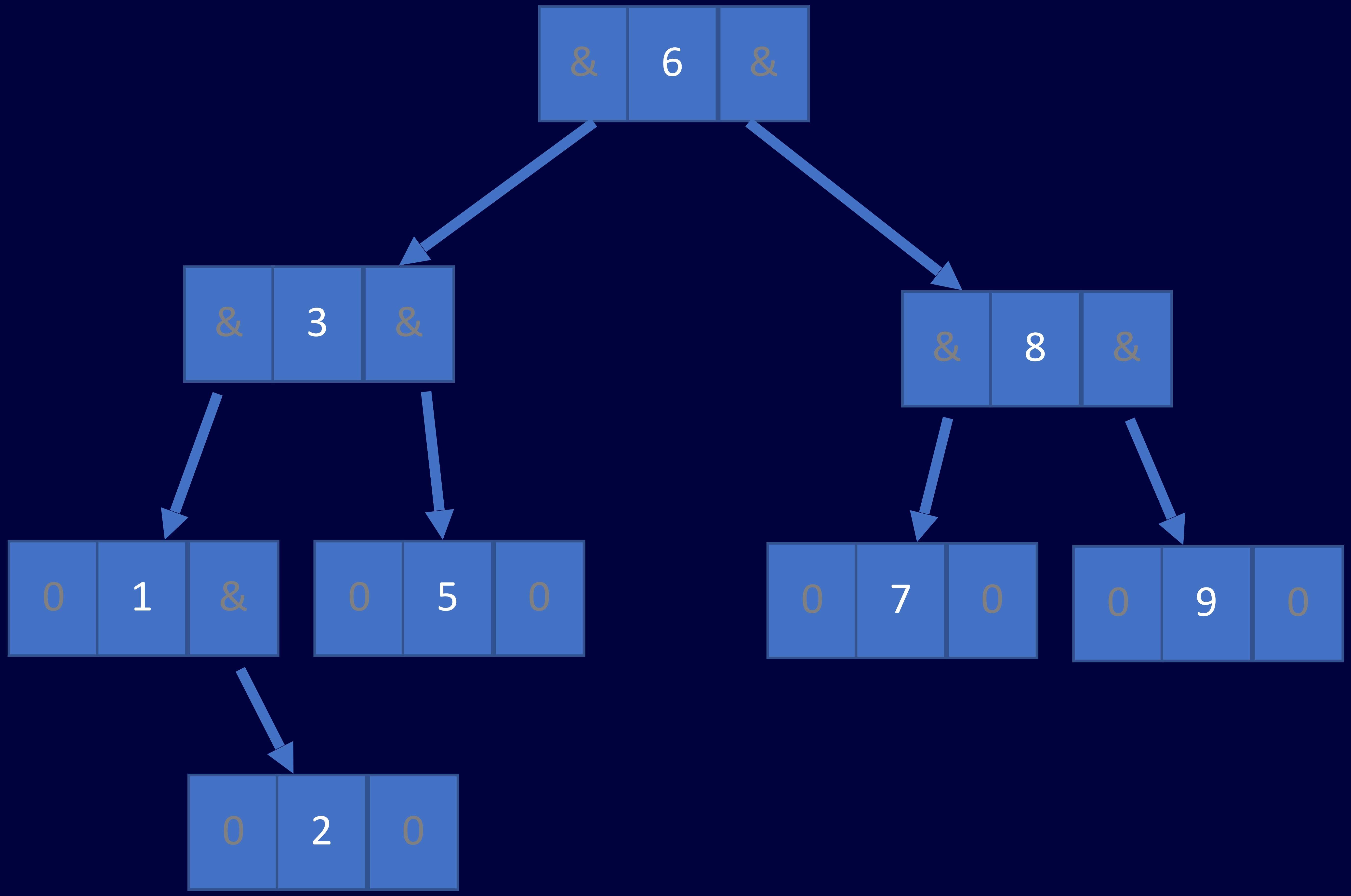
…

Linked list



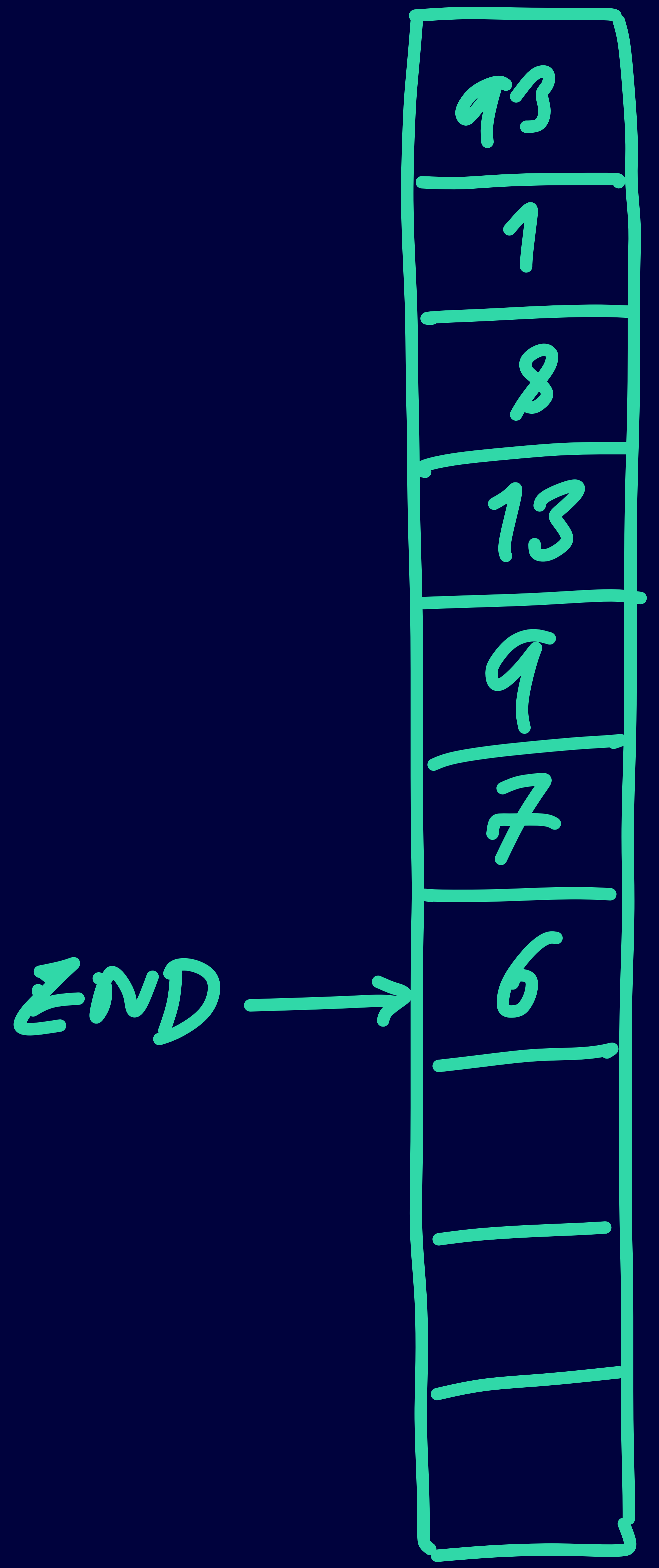
3
1
4
1
5
15678
15679
15680
15681
15682
15683
15684
15685
15686
15687
15688
15689
15690
15691
15700
15701
15710
15711
15712
3
15690
4
15682
1
15710
1
15680
5

Tree



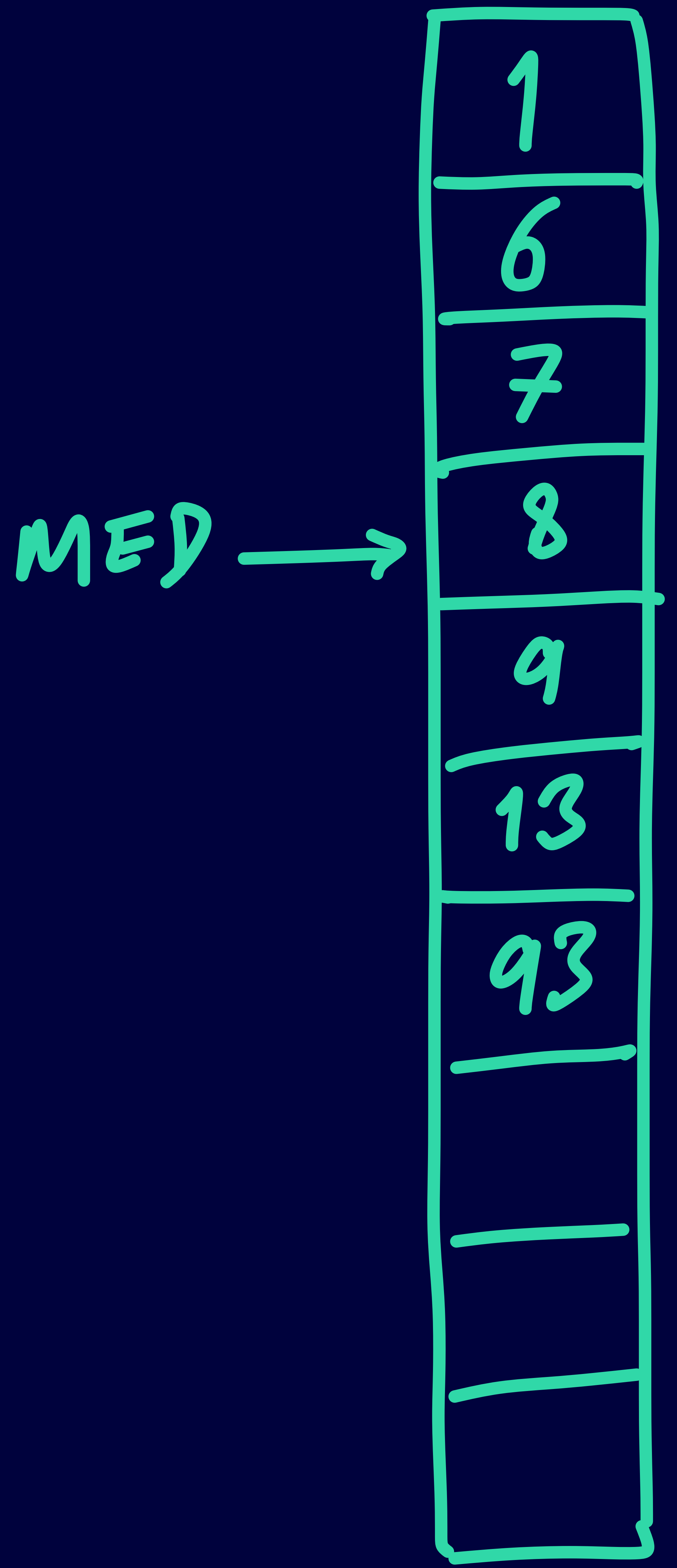
ARRAY

Search
Insertion
Deletion

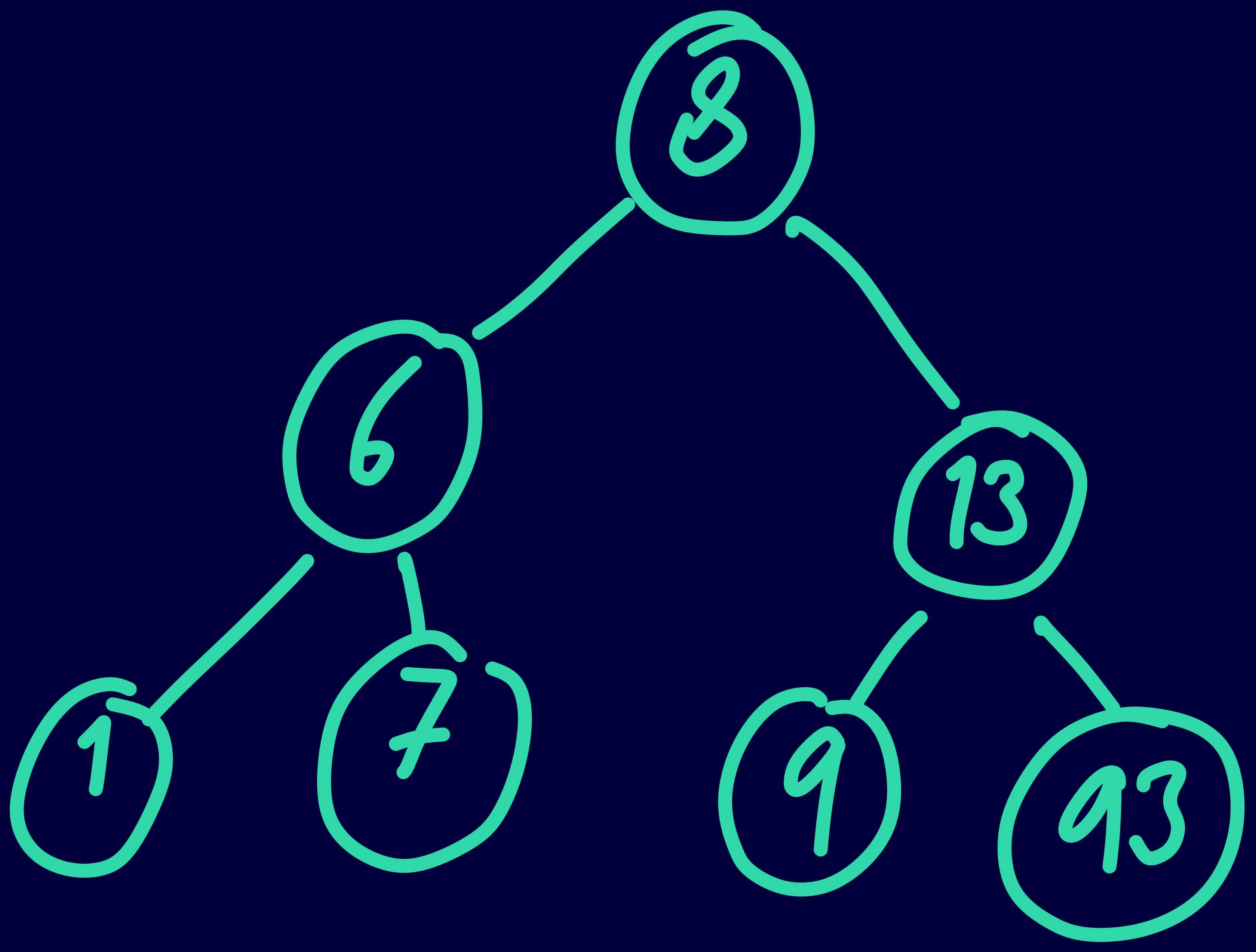


	ARRAY	SORTED ARRAY
Search	$O(n)$	
Insertion	$O(1)$	
Deletion	$O(n)$	

x



	ARRAY	SORTED ARRAY	BST
Search	$O(n)$	$O(\log_2 n)$	$O(\log_2 n)$
Insertion	$O(1)$	$O(n)$	$O(\log_2 n)$
Deletion	$O(n)$	$O(n)$	$O(\log_2 n)$



$$= 1\times10^{-6} \times 1.4 \times 10^{9} = 1400\,s$$

$$= 1\times10^{-6} \times \log_2(1.4\times10^{9}) = 30\,\mu s$$

Time to compare items: 1 μs
Population of India: 1.4×10^{9}

BST (AVL) tree

Can build self-balancing trees such as AVL or Red-Black trees “on the fly”

https://www.cs.usfca.edu/~galles/visualization/AVLtree.html

Evolved data structures

Originally conceived as mapping from the human world to the computer world

Nowadays seen as the adjunct to algorithms

- experienced computer scientists often think about data structures before algorithms

Database: flat files

Somewhere in the bowels of Barnard's Inn is the list of all lectures...

Title	Date	Time	Lecturer	Slides	Tech support	Publicity and liaison	Transcript length	Series
Snow white: evil witches	19/11/2020	18:00	Joanna Bourke	powerpoint	James	Lucia	10	Evil women
Understanding the universe with AI	23/11/2020	13:00	Roberto Trotta	powerpoint	James	Lucia	6	The unexpected universe
Data: the past, the present and the future	18/11/2020	18:00	Richard Harvey	keynote	James	Claire	4	Great Ideas from Computer Science
The changing geography of ill health	25/11/2020	18:00	Chris Witty	keynote	Richard, James	Lucia	8	Major debates in public health

There are at least 2000 lectures online - growing at around 130 per year.

/etc/passwd, awk, sed and grep

Data errors and consistency

Composer Richard Harvey.
Photo from [Wikipedia]

Singer HRVY.
Photo from [BBC News].

Databases: normal forms

Split the flat file into multiple tables to

- reduce the possibility and impact of human error
- at the expense of more tables.

Title	Date	Time	Lecturer	Slides	Tech support	Publicity and liaison	Transcript length	Series
Snow white: evil witches	19/11/2020	18:00	Joanna Bourke	powerpoint	James	Lucia	10	Evil women
Understanding the universe with AI	23/11/2020	13:00	Roberto Trotta	powerpoint	James	Lucia	6	The unexpected universe
Data: the past, the present and the future	18/11/2020	18:00	Richard Harvey	keynote	James	Claire	4	Great Ideas from Computer Science
The changing geography of ill health	25/11/2020	18:00	Chris Witty	keynote	Richard, James	Lucia	8	Major debates in public health

Key	Title	Date	Time	Lecturer	Slides	Publicity and liaison	Transcript length	Series
JB32020	Snow white: evil witches	19/11/2020	18:00	Joanna Bourke	powerpoint	Lucia	10	Evil women
RT22020	Understanding the universe with AI	23/11/2020	13:00	Roberto Trotta	powerpoint	Lucia	6	The unexpected universe
RH22020	Data: the past, the present and the future	18/11/2020	18:00	Richard Harvey	keynote	Claire	4	Great Ideas from Computer Science
CW12020	The changing geography of ill health	25/11/2020	18:00	Chris Witty	keynote	Lucia	8	Major debates in public health

+

Tech support	Key
James	JB32020
James	RT22020
James	RH22020
Richard	CW12020

1NF

Databases

1960s: The start of database systems

1970s: Relational databases

1980s: SQL

1990s: Internet interfaces

2000s: XML

2010s: Big data

Library of Congress, Prints & Photographs Division, FSA/OWI Collection,LC-USF34-045002-D

SQL

Structured Query Language

Developed in the early 1970s

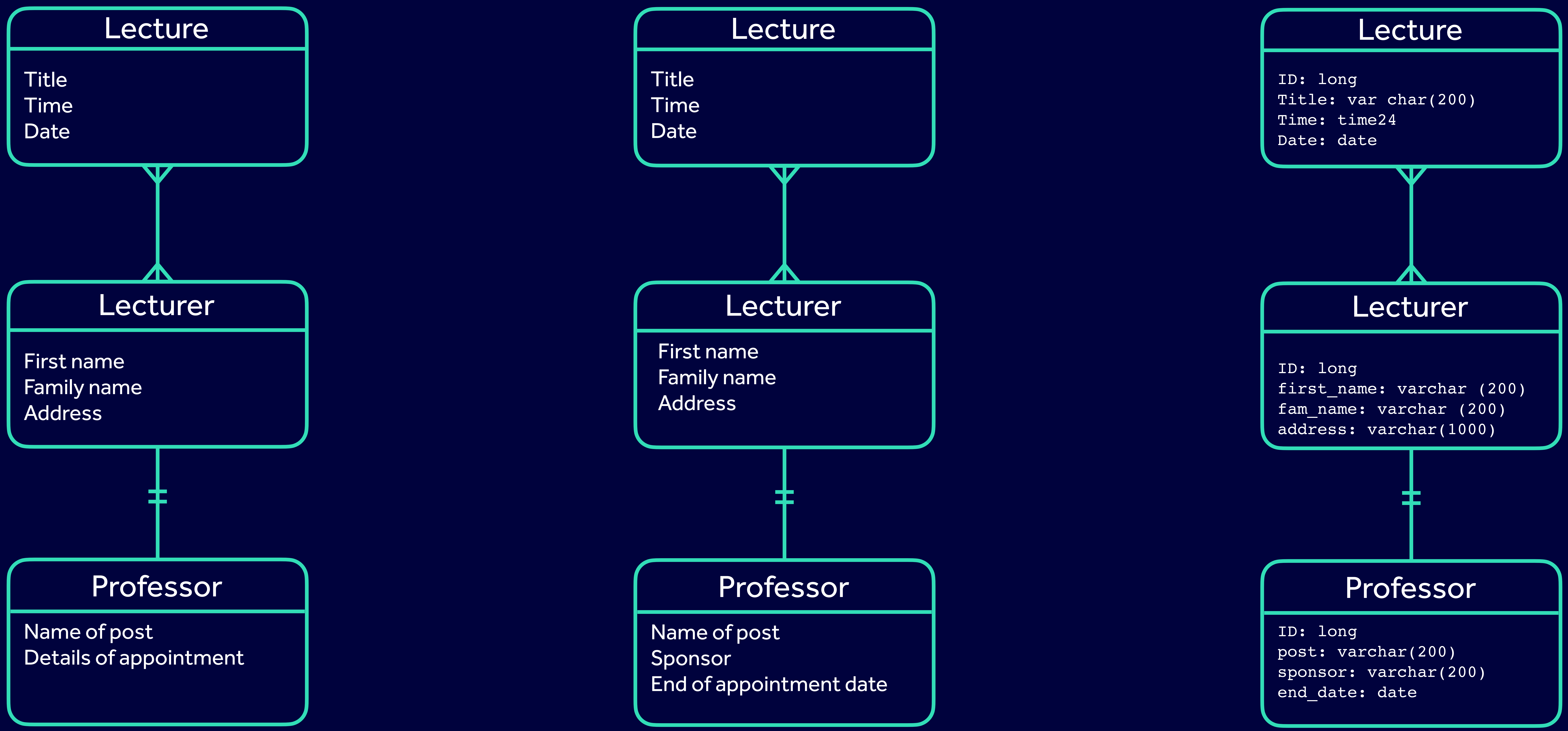
Standardised by ANSI/ISO in 1986 but standards are routinely ignored:

- SQL is a large standard so compliance can be tricky
- Databases are often legacy software
- Manufacturers favour lock-in

```
SELECT TOP 1 Salary
FROM (
   SELECT DISTINCT TOP N Salary
   FROM Employee
   ORDER BY Salary DESC
   )
 ORDER BY Salary ASC
```

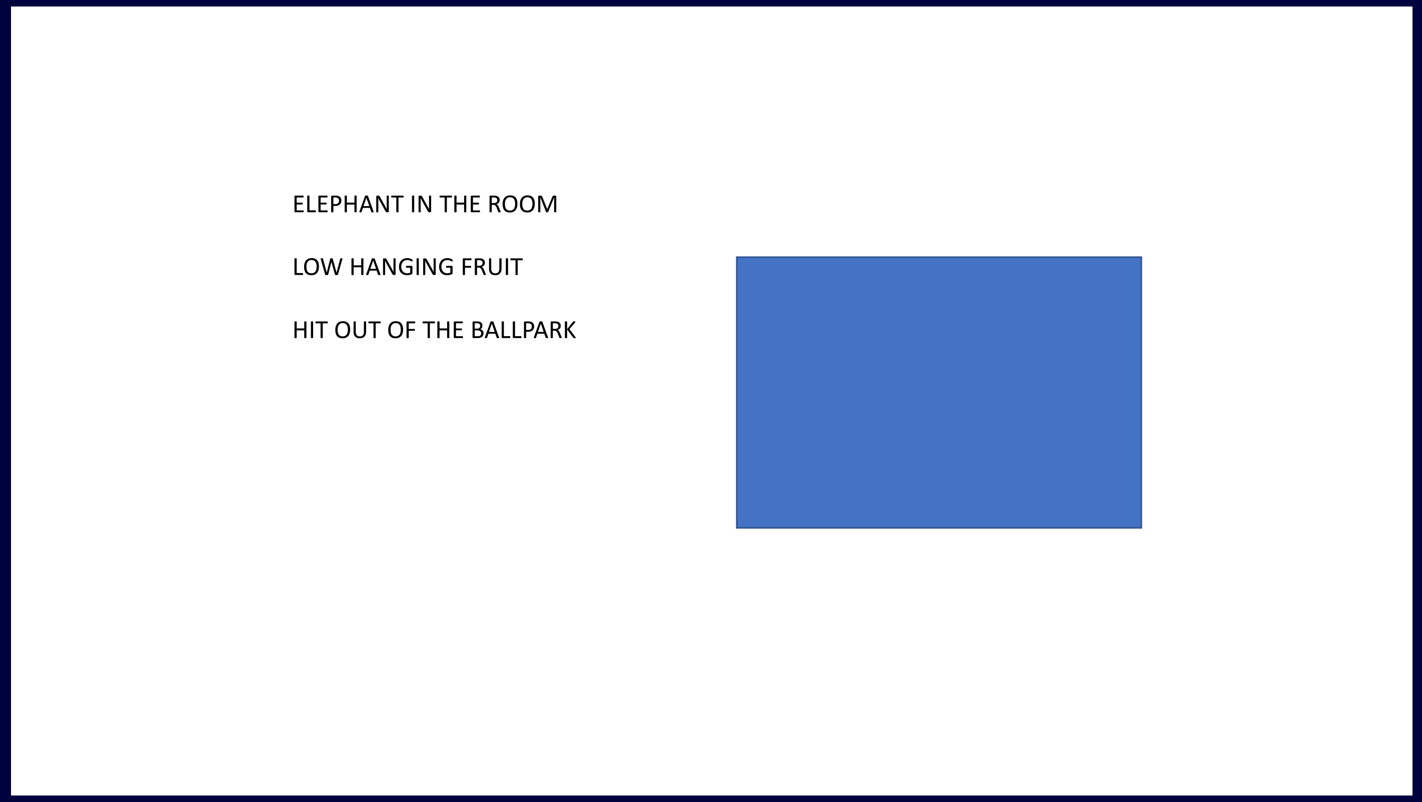
From
https://www.freecodecamp.org/news/sql-example/

ERDs



Lecture
Title
Time
Date
Lecturer
First name
Family name
Address
Professor
Name of post
Details of appointment
Lecture
Title
Time
Date
Lecturer
First name
Family name
Address
Professor
Name of post
Sponsor
End of appointment date
Lecture
ID: long
Title: var char(200)
Time: time24
Date: date
Lecturer
ID: long
first_name: varchar (200)
fam_name: varchar (200)
address: varchar(1000)
Professor
ID: long
post: varchar(200)
sponsor: varchar(200)
end_date: date

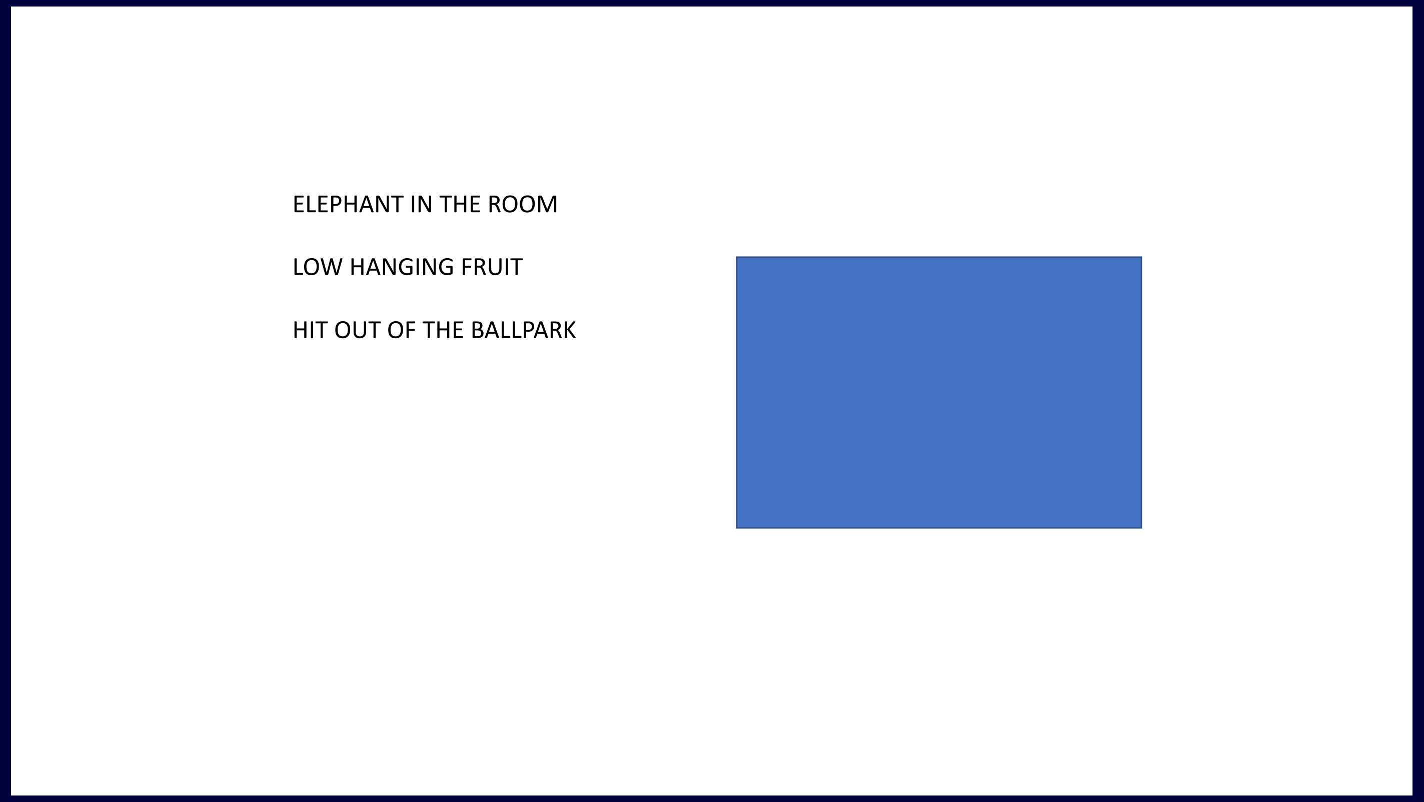
XML



Schema locations

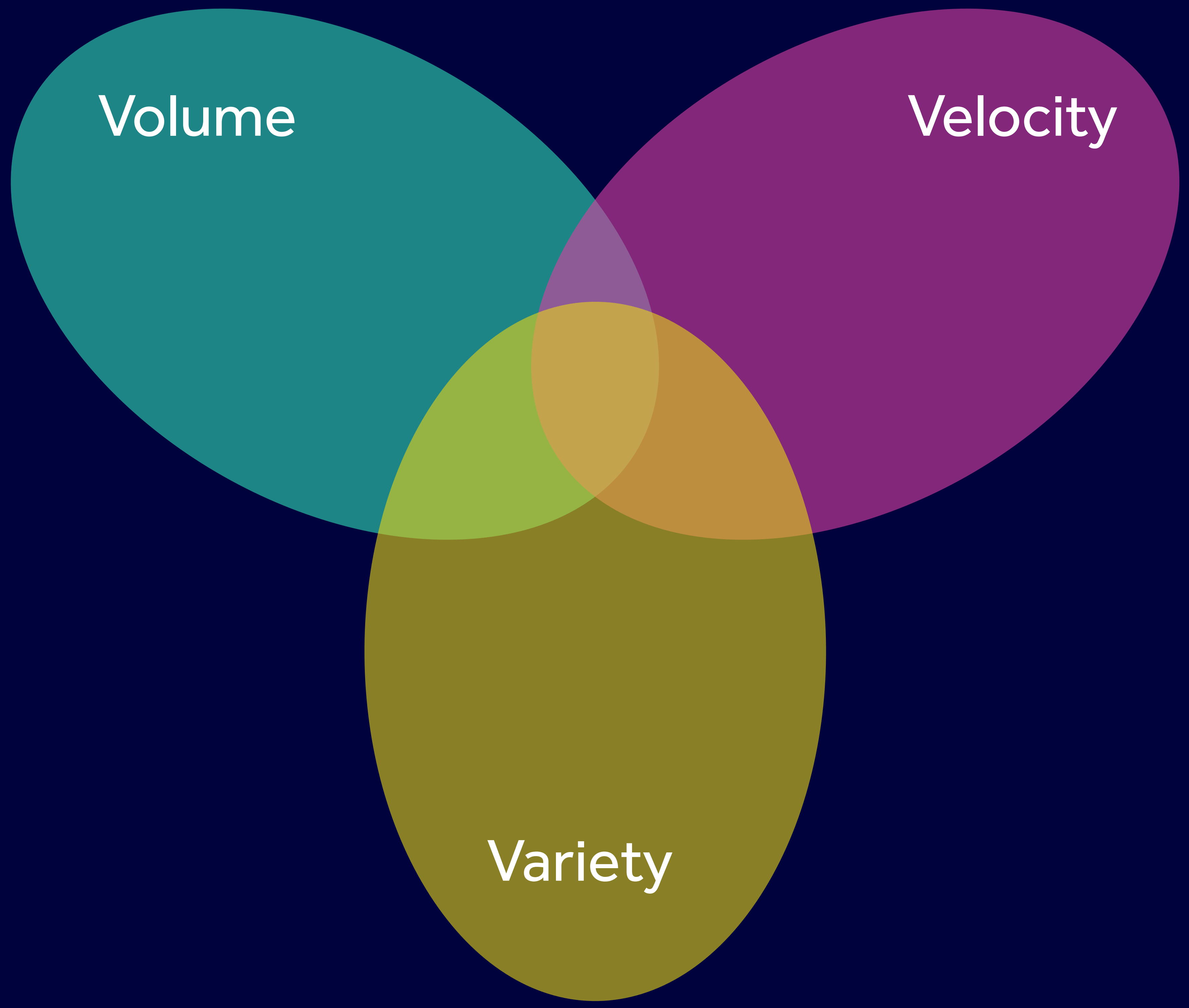
```xml
<?xml version="1.0" encoding="UTF-8"?>
<p:sld xmlns:p="http://schemas.openxmlformats.org/presentationml/2006/main" xmlns:a="http://
schemas.openxmlformats.org/drawingml/2006/main" xmlns:r="http://schemas.openxmlformats.org/
officeDocument/2006/relationships">
   <p:cSld>
      <p:spTree>
         <p:nvGrpSpPr>
            <p:cNvPr id="1" name="" />
            <p:cNvGrpSpPr />
            <p:nvPr />
         </p:nvGrpSpPr>
         <p:grpSpPr>
            <a:xfrm>
               <a:off x="0" y="0" />
               <a:ext cx="0" cy="0" />
               <a:chOff x="0" y="0" />
               <a:chExt cx="0" cy="0" />
            </a:xfrm>
         </p:grpSpPr>
         <p:sp>
            <p:nvSpPr>
               <p:cNvPr id="2" name="TextBox 1">
                  <a:extLst>
                     <a:ext uri="{FF2B5EF4-FFF2-40B4-BE49-F238E27FC236}">
                        <a16:creationId xmlns:a16="http://schemas.microsoft.com/office/drawing/2014/main"
id="{320C71F6-EBC4-4C47-9FAB-776E0702943C}" />
                     </a:ext>
                  </a:extLst>
               </p:cNvPr>
               <p:cNvSpPr txBox="1" />
               <p:nvPr />
            </p:nvSpPr>
            <p:spPr>
               <a:xfrm>
                  <a:off x="2354893" y="1528175" />
```

XML




```xml
<p:sp>
   <p:nvSpPr>
      <p:cNvPr id="2" name="TextBox 1">
         <a:extLst>
            <a:ext uri="{FF2B5EF4-FFF2-40B4-BE49-F238E27FC236}">
               <a16:creationId xmlns:a16="http://schemas.microsoft.com/office/drawing/2014/main" id="{320C71F6-
EBC4-4C47-9FAB-776E0702943C}" />
            </a:ext>
         </a:extLst>
      </p:cNvPr>
      <p:cNvSpPr txBox="1" />
      <p:nvPr />
   </p:nvSpPr>
   <p:spPr>
      <a:xfrm>
         <a:off x="2354893" y="1528175" />
         <a:ext cx="2680542" cy="1477328" />
      </a:xfrm>
      <a:prstGeom prst="rect">
         <a:avLst />
      </a:prstGeom>
      <a:noFill />
   </p:spPr>
   <p:txBody>
      <a:bodyPr wrap="none" rtlCol="0">
         <a:spAutoFit />
      </a:bodyPr>
      <a:lstStyle />
      <a:p>
         <a:r>
            <a:rPr lang="en-GB" dirty="0" />
            <a:t>ELEPHANT IN THE ROOM</a:t>
         </a:r>
      </a:p>
      <a:p>
         <a:endParaRPr lang="en-GB" dirty="0" />
      </a:p>
      <a:p>
         <a:r>
            <a:rPr lang="en-GB" dirty="0" />
            <a:t>LOW HANGING FRUIT</a:t>
         </a:r>
      </a:p>
      <a:p>
         <a:endParaRPr lang="en-GB" dirty="0" />
      </a:p>
      <a:p>
         <a:r>
            <a:rPr lang="en-GB" dirty="0" />
            <a:t>HIT OUT OF THE BALLPARK</a:t>
         </a:r>
      </a:p>
   </p:txBody>
</p:sp>
```

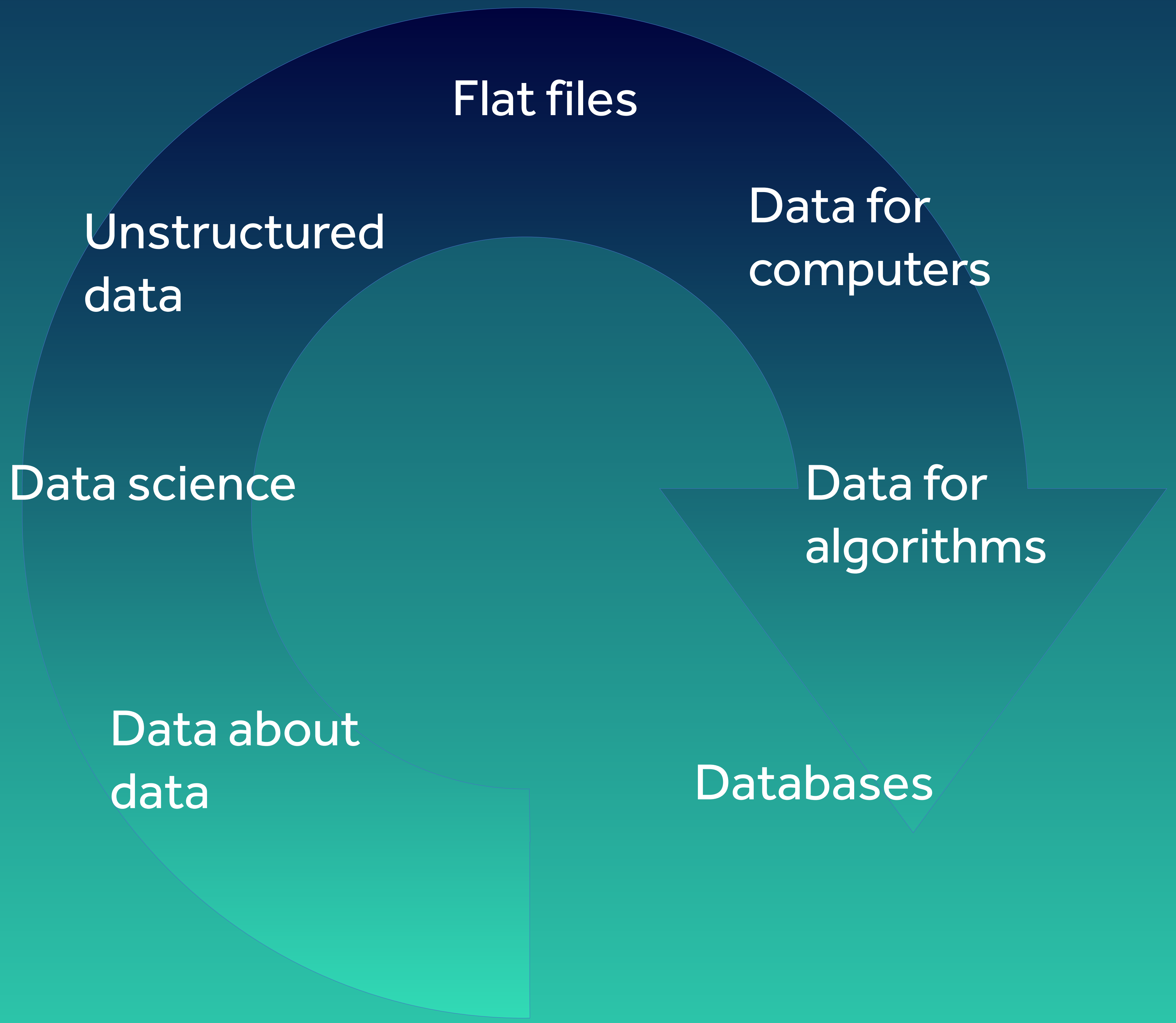
“Big” data

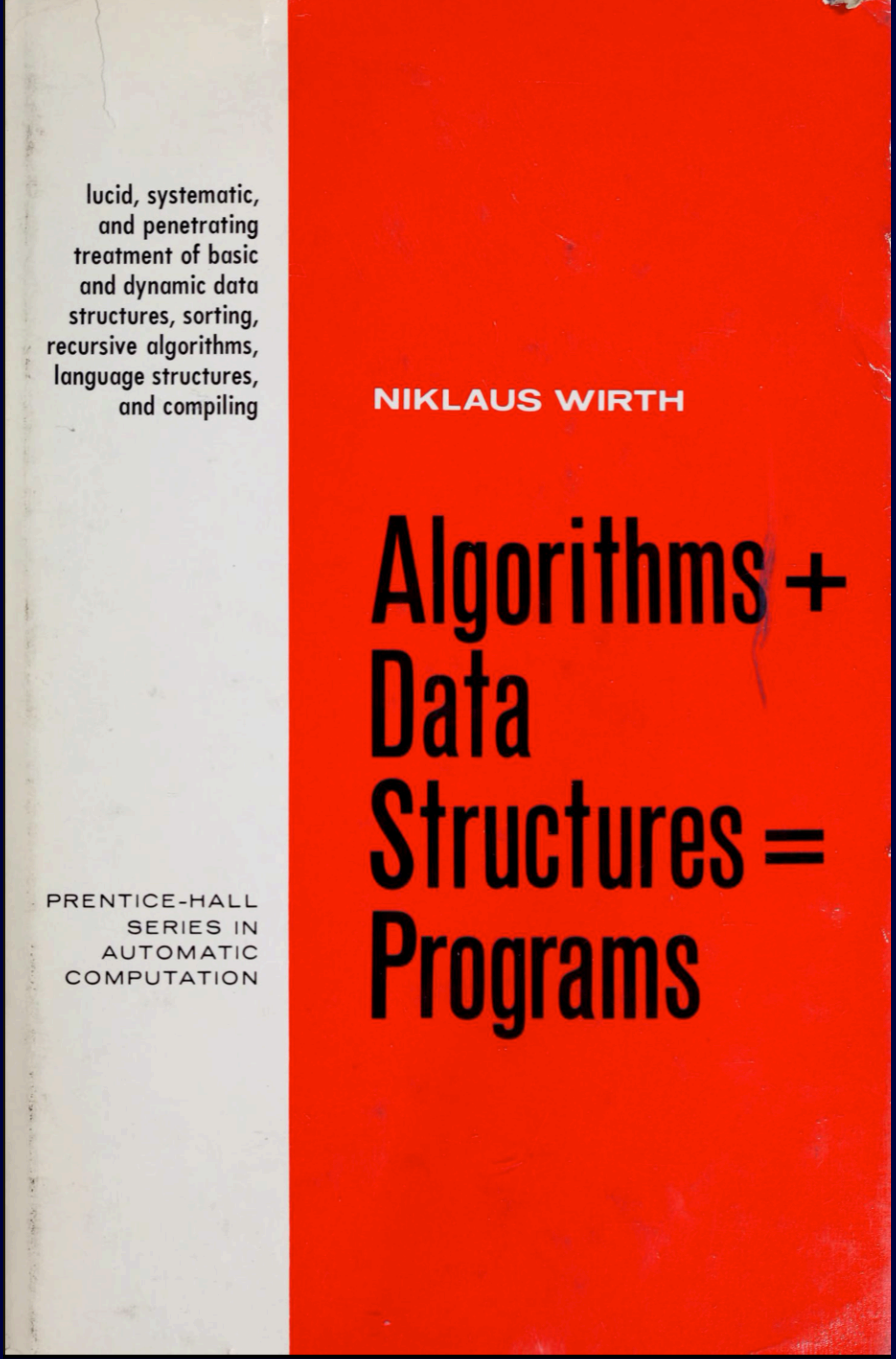


Forward-thinking leaders can begin to aggressively build their organizations’ big data capabilities. This effort will take time, but the impact of developing a superior capacity to take advantage of big data will confer enhanced competitive advantage over the long term and is therefore well worth the investment to create this capability. But the converse is also true. In a big data world, a competitor that fails to sufficiently develop its capabilities will be left behind.

Big data: The next frontier for innovation, competition, and productivity , James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers, McKinsey Global Institute, May 2011

The seven ages of data





Algorithms (previous lecture)

Data structures (this lecture)

Programs (2nd Feb 2021 18:00)

Computers (9th March 2021 18:00)

Networks (20th April 2021 18:00)

Security (25th May 2021 18:00)